

# **Fusion of Low-Cost and Light-Weight Sensor System for Mobile Flexible Manipulator**

by  
Chang Tai Kiang, M.Eng.Tech.

**Thesis submitted to the University of Nottingham  
for the degree of Doctor of Philosophy**



The University of  
**Nottingham**

Malaysia Campus

**April 2018**

## **Declaration of Authorship**

I, Chang Tai Kiang, hereby declare that this submission is my own work and that it has not been submitted previously for any other degree. To the best of my knowledge and belief, it contains no material previously published or written by another person, except where information reference is made.

Chang Tai Kiang

30 April 2018

## **Abstract**

There is a need for non-industrial robots such as in homecare and eldercare. Light-weight mobile robots preferred as compared to conventional fixed based robots as the former is safe, portable, convenient and economical to implement. Sensor system for light-weight mobile flexible manipulator is studied in this research.

A mobile flexible link manipulator (MFLM) contributes to high amount of vibrations at the tip, giving rise to inaccurate position estimations. In a control system, there inevitably exists a lag between the sensor feedback and the controller. Consequently, it contributed to instable control of the MFLM. Hence, there it is a need to predict the tip trajectory of the MFLM.

Fusion of low cost sensors is studied to enhance prediction accuracy at the MFLM's tip. A digital camera and an accelerometer are used predict tip of the MFLM. The main disadvantage of camera is the delayed feedback due to the slow data rate and long processing time, while accelerometer composes cumulative errors. Wheel encoder and webcam are used for position estimation of the mobile platform. The strengths and limitations of each sensor were compared.

To solve the above problem, model based predictive sensor systems have been investigated for used on the mobile flexible link manipulator using the selected sensors. Mathematical models were being developed for modeling the reaction of the mobile platform and flexible manipulator when subjected to a series of input voltages and loads.

The model-based Kalman filter fusion prediction algorithm was developed, which gave reasonably good predictions of the vibrations of the tip of flexible manipulator on the mobile platform. To facilitate evaluation of the novel predictive system, a mobile platform was fabricated, where the flexible manipulator and the sensors are mounted onto the platform. Straight path motions were performed for the experimental tests.

The results showed that predictive algorithm with modelled input to the Extended Kalman filter have best prediction to the tip vibration of the MFLM.

## List of Publications

To date, publications originated from this works are listed below:

1. T. K. Chang, K. Y. Chan, A. C. Spowage. Development of a Local Sensor System for Analysis of a Badminton Smash. *Proceedings of the 3rd International Conference on Mechatronics, ICOM'08*, 18 – 20 December 2008, Kuala Lumpur, Malaysia, pg 268-273. (The paper was published in the ICOM'08 issue).
2. Chang Tai Kiang, Chan Kuan Yoong, A. C. Spowage. Local Sensor System for Badminton Smash Analysis. *Technology Conference*, Singapore, 5-7 May 2009, pg 883-888.
3. Chang Tai Kiang, A. C Spowage, Chan Kuan Yoong. “Review of control and sensor system of flexible manipulator”. *Journal of Intelligent and Robotic Systems*, Springer, Accepted 13 June 2014, (DOI: 10.1007/s10846-014-0071-4)
4. Tai Kiang Chang, Jee-Hou Ho. Extended Kalman Filter Based Modelled Predictor for Fusion of Accelerometer and Camera Signal to Estimate the Vibration of a Mobile Flexible Link Manipulator. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*.

## **Acknowledgements**

I would first like to graciously acknowledge Dr. Samer Yahya and Dr. Ho Jee Hou for the support they have provided in the various ways over the course of developing this thesis. They have given me much of the advices and providing valuable resources needed to complete the studies, provided guidance at critical junctures and his review and comments. I would also like to express my appreciation to Dr. Ho Jee Hou for his advice and help in providing resources needed for the research.

I want to express my deep appreciation to my family for their encouragement, support and love. Lastly, I would like to thank Dr. Chan Kuan Yoong and Dr. Andrew Spowage for the hardware supports, and they have been very helpful towards my needs.

## Nomenclature

$a$	Acceleration
$e$	Back E.M.F voltage
$d$	Displacement of mobile platform
$D$	Damping coefficient
$E$	Young's modulus
$E_k$	Kinetic energy
$E_p$	Potential energy
$F$	Force
$f_n$	Natural frequency
$f_s$	Sampling frequency
$g$	Gravitational force
$F_a$	Frictional force between wheel and wheel's axle
$F_{disturb}$	Disturbance force
$Fps$	Frame per second
$F_s$	Static friction
$F_w$	Friction of rear wheel
$I$	Moment of inertia of beam
$I_a$	Current
$J_m$	Motor's moment of inertia
$J_w$	Wheel's moment of inertia
$KE$	Kinetic energy
$k_{stiff}$	Stiffness of beam
$k_e$	Back E.M.F constant
$k_i$	Torque constant

$k_{sp}$	Spring constant
$k_v$	Offset voltage constant
$l$	Length of mobile platform
$L$	Armature inductance of motor
$L_b$	Length of flexible beam
$m$	mass
$N_F$	Normal force acting vertically on the on the wheel
$P$	Force acting at the free end of beam
$PE$	Potential energy
$R$	Armature resistance of motor
$R_a$	Axle radius of wheel
$R_v$	Virtual radius of wheel
$R_w$	Wheel radius
rad	Angular measurement, radian
rpm	Revolution per minute
s	Second (time)
$t$	Time
$T_{coul}$	Coulomb friction
$T_{coup}$	Coupling torque
$T_{fr}$	Frictional torque
$T_{static}$	Static friction
$T_{stribeck}$	Stribeck friction
V	Voltage
$u$	Input vector
$v$	Linear velocity
$v_{el}$	Linear velocity of mobile platform

$v_c$	Viscous friction coefficient
$v_s$	Stribeck velocity
$w$	Deflection of flexible beam
$\omega_n$	Natural frequency
$\omega_m$	Angular velocity
$x$	Displacement or position measurement
$y$	Output vector
$\tau_{acc}$	Acceleration torque for the mobile platform
$\tau_c$	Torque at the caster wheels
$\tau_f$	Torque acting on front wheel
$\tau_m$	Torque generated by the motor
$\tau_t$	Total torque at the wheels to move the vehicle
$\Gamma$	Shaft torque at the wheel
$\theta_m$	Angular displacement of motor shaft
$\theta_w$	Angular displacement of wheel
$\mu$	adhesion coefficient acting between wheel and ground
$\xi$	Damping ratio
$\eta$	Gear ratio
$\mathcal{L}$	Lagrange's function

## Abbreviations

MEMS	Micro Electro Mechanical Systems
KF	Kalman Filter
EKF	Extended Kalman Filter
UKF	Unscented Kalman Filter
GPS	Global Positioning System
IMU	Inertial Measurement Unit
FIS	Fuzzy Inference System
ANN	Artificial Neural Network
ANFIS	Adaptive Neuro-Fuzzy Inference System
PF	Particle Filter
DOF	Degree of Freedom
AE	Acoustic Emission
LVDT	Linear Variable Differential Transformers
PDF	Probability density function
PZT	Piezoelectric transducer
FM	Flexible manipulator
FLM	Flexible link manipulator
MFLM	Mobile flexible link manipulator
TPBVP	Two point boundary value problem
RMS	Root mean square
IEEE	Institute of Electrical and Electronics Engineering
LED	Light emitting diode
USB	Universal serial bus
IR	Infra Red

# Table of contents

Declaration of Authorship.....	ii
Abstract.....	iii
List of Publications .....	iv
Acknowledgements.....	v
Nomenclature.....	vi
Abbreviations.....	ix
Table of contents.....	x
1. Introduction.....	1
1.1 Project motivation.....	2
1.2 Aim and Objectives.....	4
1.3 Research phases .....	5
1.4 Research contributions.....	6
1.5 Outline of the thesis .....	7
2. Review on past related works.....	8
2.1 Sensors for Robotic Manipulators.....	9
2.1.1 Absolute techniques for robot localization .....	10
2.1.2 Absolute measurement methods for flexible manipulator displacement ..	11
2.1.3 Comparison of absolute methods.....	12
2.1.4 Relative methods.....	15
2.1.5 Comparison of relative methods .....	16
2.2 Sensor selections .....	18
2.2.1 Selecting sensors in absolute methods.....	19
2.2.2 Selecting sensor in relative methods.....	20
2.3 Sensors introductions .....	21
2.3.1 Camera and webcam .....	21
2.3.2 Accelerometer .....	25
2.4 Sensor fusion techniques.....	31
2.4.1 Sensor fusion techniques – Weighted averaging .....	31
2.4.2 Sensor fusion techniques – Kalman filter .....	32
2.4.3 Sensor fusion techniques – Complementary filters.....	33
2.4.4 Sensor fusion techniques – Particle filter.....	34
2.4.5 Sensor fusion techniques – Fuzzy inference system.....	35
2.4.6 Sensor fusion techniques – Neutral network.....	35
2.4.7 Comparison of sensor fusion techniques.....	36
2.5 Literature review of past works.....	38
2.6 Benchmarking Conclusion.....	44

3	Sensor Characterizations and Modelling of Mobile Flexible Link Manipulator.....	47
3.1	Characterizations of Accelerometer .....	47
3.1.1	Mathematical equations for computing accelerometer signal.....	47
3.1.2	Methodology for testing and conditioning the accelerometer.....	49
3.1.3	Preparation of the accelerometer.....	51
3.1.4	Filter selection.....	53
3.1.5	Calibrating the accelerometer .....	54
3.1.6	Effect of walking bias errors .....	54
3.1.7	Dynamic verifying the accuracy of the accelerometer.....	55
3.1.8	Cross-axis affects.....	57
3.2	Characterization of camera .....	58
3.2.1	Algorithm for computing camera data .....	58
3.2.2	Methodology for testing camera .....	60
3.2.3	Preparation and setup for testing web cam and CMOS camera.....	63
3.2.4	Calibration of camera for measuring vibration of flexible beam.....	67
3.2.5	Tests for mobile robot localization with webcam.....	67
3.2.6	Tests for vibration tracking of the flexible beam with camera .....	68
3.3	Characterization of encoder .....	69
3.3.1	Equation for computing encoder signal .....	70
3.3.2	Methodology for testing encoder .....	70
3.3.3	Preparation for testing the encoder .....	72
3.3.4	Calibrate the encoder .....	73
3.3.5	Testing the encoder .....	73
3.4	Modelling for Mobile Flexible Link Manipulator .....	74
3.4.1	Dynamic Model of Mobile platform.....	74
3.5	Model of flexible link manipulator .....	88
3.6	Model of mobile flexible link manipulator .....	96
3.7	Methodology for identifying the constants for the motor and simulations to test the model .....	100
3.7.1	Methodology for identifying back EMF constant and armature resistance.....	101
3.7.2	Methodology for identifying torque constant .....	102
3.7.3	Methodology for identifying armature inductance and moment of inertia .....	102
3.7.4	Methodology for identifying viscous friction coefficient and Coulomb friction .....	103
3.7.5	Methodology for identifying static friction and stribeak effect .....	104
3.7.6	Testing the motor model .....	104
3.8	Chapter summary .....	105
4.	Tests results for the sensors and simulations for the mobile flexible link manipulator .....	107
4.1	Test results of testing and conditioning the accelerometer .....	107
4.1.1	Static tests and noises attenuation .....	107
4.1.2	Vibration tests and selection of curve smoothing filter .....	109
4.1.3	Results for calibrating accelerometer.....	111

4.1.4	Results for dealing with walking bias errors.....	112
4.1.6	Evaluating the results .....	123
4.1.7	Cross-axis affects .....	124
4.2	Tests results for webcam and camera.....	127
4.2.1	Tests results for initial displacement measurement using webcam .....	127
4.2.2	Test results for displacement measurement that used interpolation method for the webcam .....	129
4.2.3	Results for webcam dynamic tests for localization of mobile platform..	131
4.2.4	Results for capturing vibration of flexible beam using camera .....	135
4.2.5	Results for vibration tracking of flexible beam using camera .....	136
4.3	Results for the encoder tests.....	137
4.3.1	Results for static tests of the encoder .....	138
4.3.2	Results for dynamic tests of the encoder.....	138
4.4	Simulations for the models of motor.....	140
4.4.1	Identifying the constants for the motor .....	140
4.4.2	Simulation results for testing the motor model .....	145
4.5	Chapter Conclusion .....	147
5	Mobile flexible link manipulator and its predictive sensor system.....	149
5.1	Kalman filter and cross correlation algorithms .....	149
5.1.1	Kalman filter .....	150
5.1.2	Extended Kalman filter .....	151
5.1.3	Cross-correlation .....	153
5.2	Development of sensor fusion algorithm .....	154
5.2.1	Sensor fusion algorithm .....	154
5.2.2	Applying the extended Kalman filter.....	157
5.3	Development of predictive algorithm .....	158
5.5	Methodology for comparing simulated vibration to actual vibration of the flexible manipulator .....	165
5.6	Methodology for comparing simulated vibration with actual vibration for the mobile flexible link manipulator.....	167
5.6.1	Test rig setup for the mobile flexible link manipulator.....	169
5.6.2	Tests and simulations on the mobile flexible link manipulator .....	170
5.7	Methodology for tests on the mobile flexible link manipulator with sensors.	172
5.7.1	Setup for mobile flexible link manipulator with sensors .....	173
5.7.2	Tests on the mobile flexible link manipulator.....	176
5.8	Validation tests with sensor fusion algorithm.....	177
5.9	Methodology for validating predictive algorithm on the MFLM .....	179
5.10	Summary .....	181
6	Performance of the predictive sensor system - results and discussions.....	183
6.2	Results for comparing simulated vibration to actual vibration of the mobile flexible link manipulator.....	187
6.2.1	Tests with 0.67 V step input voltage.....	187
6.2.2	Tests with 1.34 V step input voltage.....	189

6.2.3	Tess with 2.01V step input voltage .....	190
6.2.4	Tests with 2.68 V step input voltage (this test included unexpected halt disturbance).....	192
6.2.5	Test with 3.35 V step input voltage (this included unexpected disturbances .....	193
6.3	Experimental test results of the mobile flexible link manipulator with sensors	195
6.3.1	No payload, tested with various input voltages .....	195
6.3.2	With 10 g payload, tested with various input voltages .....	198
6.3.3	With 30 g payload, tested with various input voltages .....	202
6.4	Test results with sensor fusion algorithm .....	205
6.4.1	Tests without camera data correlation .....	205
6.4.2	Tests with camera data correlation.....	206
6.4.3	Tests with windowed frame for vibration tracking.....	208
6.5	Validation results with predictive algorithm.....	211
6.5.1	Validation test results comparing the predictive algorithms.....	212
6.5.2	Validation tests with different payloads and input voltages .....	213
6.6	Conclusion .....	222
7	Conclusions and future works .....	224
7.1	Conclusions of current works.....	224
7.2	Proposed future works.....	226
	Bibliography.....	227
Appendix A	Reviews of encoders .....	238
A.1	Types of encoders .....	238
A.2	Comparison of encoders.....	239
Appendix B	Datasheet for sensors.....	240
B.1	Accelerometer datasheet .....	240
B.2	Camera datasheet .....	244
B.3	Range sensor datasheet .....	246
B.3.1	Long range measurement .....	246
B.3.2	Short range measurement.....	249
B.4	Encoder datasheet.....	252
B.5	Current sensor datasheet .....	254
Appendix C	Smoothing filters.....	255
Appendix D	Matlab Codes .....	257
D.1	Parallel computing .....	257
D.3	Acquire image from camera.....	259
D.4	Processing the image.....	260

# 1. Introduction

The tremendous success of robotics has had automating many industrial activities, mainly based on articulated<sup>1</sup> robots. The first industrial robot was “Unimates” (UNiversal AutoMATION) [1] developed by George Devol and Joseph Engleberger during early 60’s. Today’s robotics is emerged into achieving similar advancements in non-conventional environments to change our way of life. As the technology advances, there will be needs for domestic robots in the near future [2]. There are existing applications that pose potential for lightweight alternatives:

1) The aerospace and outer space industries, such as driving lighter robots in satellites using lesser energy, for example the space station remote manipulator system [3]. Weights of the robots have to be minimized while transporting into outer space.

2) These flexible manipulators can be seen in hospitals as well, especially in the operation rooms. Ljungblad et al. [4] reported the need for the hospital robots. Current robotic systems are bulky and high-cost, thus, these applications are open for future development of light-weight robots.

3) As an example of using robots in offices and restaurants is the autonomous restaurant service robot by Yu et al. [5]. These robots are mainly large and high-cost, which can be replaced with light-weight robotic manipulators.

4) In homecare; there is a future trend having homecare robot for elderly. Yamazaki et al. [6] the need for the homecare robots in aging society. This is still a very new technology in domestic area, homecare robots for assisting handicapped or elderly hold tremendous potentials for assisting aging populations. The size and cost of these robotic systems needs to reduce considerably before it can gain acceptances.

In many developed countries, fertility rates are low, homecare robots is predicted to play important role in providing assistance to the aging populations in the near future [6]. Domestic and service robots may one day be used to lower this burden of care and

---

<sup>1</sup> An articulated robot is a robot with rotary joints, which can range from simple two-joint arms to multiple-joints arms.

enhance the quality of life for the worlds' senior citizens. Naturally, this will lead to a future need for domestic and homecare robots.

The problems are these robotic systems are mostly bulky and high power consumption, and expensive. For these reasons, much works are still needed to develop lighter, faster, lower energy and more economical robots. However, the very challenging problems of the light-weight flexible link robotic manipulators are the lack of accuracy in trajectory and positioning control.

There are many factors to be considered when designing non-industrial robots, whereas safety being a fundamental. Other factors to consider are it has to be economical and lower power consumption. Lightweight mobile flexible link manipulators can bring success for these factors. One of the key enabling technologies (in the view of the author) that can make such systems one step closer to reality is by having a low cost and light-weight sensor system that can predict the end-effector's position and vibration. This is the main focus of this research.

## **1.1 Project motivation**

Extensive use of flexible link manipulators (FLMs) in various robotic applications has brought research interests for many scholars over the world over the last few decades. The study of FLMs began since the 1970's [7] across many engineering disciplines, while the 1980's rise major research efforts. Earlier studies were mainly focused on the dynamic modeling of these flexible manipulators. Those studies were documented very well [8] and in some text books. The newer applications and latest technological advancements brought attentions on the importance of end-effector trajectory tracking and vibration reductions at the tip of FLMs. Recently, researchers have focused on the controlling of the flexible manipulator as well as applying them onto different robotic platforms. Both link and joint flexibility were considered, which can be either revolute or prismatic manipulators.

Many research works have been focusing on single or two links flexible manipulators, or manipulators with only joint flexibility, but most of the papers are still based on single link types. Also there has not been many works that are based on flexible manipulators

with large elastic deflections. There is also lack in the study on mobile flexible manipulators. Mobile robots has very much wider workspace with small footprint, hence it become reasonable to employ mobile robot into home-based environments. It is thus the motivation of this research to focus the study on tracking the tip vibration of the mobile flexible link manipulator.

For position tracking of the manipulator, conventional robots are designed to have maximum stiffness so as to achieve minimum vibration and good positioning accuracy of the end-effector. Consequently, these types of robots suffered from bulky design that requires high power drives. Besides, minimal vibrations at the end-effector still exist when operating at high speed and at high payloads [9].

FLMs on the other hand enjoyed the advantages of smaller footprint, lighter, faster and lower energy requirement. In addition, flexible manipulators offer wider and newer robotics applications which are safer, economical and portable. The major problem of the flexible manipulators is the constrained to the high amount of vibrations [10]. This vibrations and flexing of the flexible manipulator contributed considerable uncertainties for position estimation at the tip of the FLM due to the nonminimum phase characteristic caused by the control problem of non-holonomic drive and unbounded response at the tip. The cause is due to under-actuation feature happening at the joint due to the lesser actuators as compared to the degrees-of-freedom [11].

For any applications, such as fetching a glass of water, the end-effector of the robot needs to move along a desired trajectory at a specified speed [12]. The links of flexible manipulators will deflect and vibrate during and after a maneuver. For the end effector trajectory tracking, the actuators apply the control torque (or force) according to the end-effector feedback, so as to have low the tracking errors. The velocity profile of the robot has to be controlled, and to be more effectively plan the control of the end-effector, we need to predict the trajectory at the tip of the FLM.

Many methods have been proposed to solve the trajectory tracking problems for flexible manipulators by roboticists for decades. The important result from surveying the vast body of literature on accurate position tracking of FLM is that to date there is no low cost solution for the problem. Most highly accurate position measurement systems are expensive, and often bulky. An example of one such system characterized with high

accuracy is an external measurement system [13] which uses laser tracker system. These measurement systems consequently increase the cost of the robot, and at the same time add on to the overall weight of the robot arm. Therefore, there is a shortage of the research on the feedback system that can predict the vibration at the tip of the flexible manipulator [14].

In the knowledge of this research, there has not been any works that based on the fusion of data to predict future behaviour for the tip of mobile flexible link manipulator. Therefore, accurate position tracking and prediction for light-weight mobile flexible manipulator having large elastic deflection posed research motivation to explore. The aim is to investigate the fusion technique for low-cost and light-weight sensors for achieving accurate localization and vibration estimation.

## **1.2 Aim and Objectives**

The aim of this research is to achieve one centimetre position prediction accuracy for the mobile flexible link manipulator's (MFLM's) end-effector for the duration and end of manoeuvres. The research objectives are as follows:

1. Identify the sensors that are light-weight and small-size, suitable for implementing on the MFLM that can be used for motion and vibration estimation, follow by reviewing the sensor fusion techniques suitable for fusing the selected sensors. Then, review past related works on tracking the vibration of FLMs and MFLMs.
2. Test all adopted sensors for their accuracy and their repeatability in displacement measurement.
3. Develop the model for the mobile flexible manipulator.
4. Develop methods to determine the constants of the DC motor.
5. Fabricate a flexible link manipulator and mobile test rigs, and then test the behaviors at tip of MFLM.
6. Develop of sensor fusion algorithms for fusing the sensors to predict the trajectory at the tip of the FLM, and formulate algorithms that can predict vibration of FLM based on the sensors' outputs.

7. Perform validation tests on the mobile flexible link manipulator using the novel algorithm to verify the prediction accuracies.

### **1.3 Research phases**

The project is divided into 3 phases, in each phase some of the research objectives were carried out:

Phase 1 involves identifying the sensors and the sensor fusion techniques that can be used for motion and vibration tracking of the mobile platform and the flexible manipulator. Then, a review of the literature on the various techniques that were being worked on for tracking the vibrations of FLMs and MFLMs.

Phase 2 involves the characterization of the selected motion sensors and developing the models for the mobile flexible link manipulator. The following tasks were set:

1. Define methodologies for testing the sensors, followed by carrying out the tests.
2. Developing the models for the motor, the flexible manipulator and the mobile platform.
3. Define methodologies for identifying the various constants and friction coefficients for the motors, followed by validating the motor model.

Phase 3 is development of the novel motion and vibration prediction algorithm for the mobile flexible link manipulator using the chosen sensors. The following tasks were set:

1. Develop the sensor fusion and predictive algorithms for tracking the vibration at the tip of the mobile flexible manipulator.
2. Build the flexible manipulator and mobile test rigs.
3. Define methodologies for validating the model of the flexible manipulator and the mobile flexible manipulator, followed carrying out the simulations.
4. Define methodologies for testing the vibration tracking of the flexible manipulator and mobile flexible manipulator using the chosen sensors, followed by carrying out the tests.
5. Define methodologies for testing the novel sensor fusion and prediction algorithms, followed by carrying the tests.

## 1.4 Research contributions

The thesis contributes to the following areas:

1. Reviewed a list of sensors and sensor fusion techniques suitable for tracking and predicting the vibration of the MFLM. Benchmarked a list of sensor tracking methods and sensor fusion techniques. (Refer to chapter 2)
2. Methodologies were being defined for testing the selected sensors. The comprehensive test results ascertained that the selected sensors are suitable for use in the current research work on motion and vibration tracking of MFLM. On the tests on accelerometer, a method was developed to remove walking bias errors present in accelerometer. Other tests proved that each sensor have their drawbacks which draw a conclusion that we could not depend on a single sensor type for realtime motion measurement in the current research work. Sensor fusion was thus recommended. (Refer to chapter 3 and 4)
3. Formulated the dynamic model for the motor, flexible link manipulator and mobile flexible link manipulator. The model can be used to accurately predict the movement of the MFLM in the absent of disturbances and at sufficient input voltage. (Refer to section 3.4 of chapter 3)
4. Developed simple methods of determining the constants for the DC motor and its friction coefficients. The accuracy of the motor model and acquired motor constants parameters has been verified. (Refer to section 3.7 and 4.4.1, as well as the validation results in section 4.4.2).
5. Fabricated separately a flexible beam for the flexible link manipulator and a mobile platform. The flexible link manipulator and the mobile platform were then assembled to form the whole mobile flexible link manipulator. The behaviour of the flexible, the mobile platform and the MFLM were tested and compared with the simulations of the model being developed. (Refer to sections 5.4 to 5.6, as well as the results in sections 6.1 and 6.2).

6. A novel local motion and vibration prediction algorithm has been developed for predicting ahead the vibration at the tip of the manipulator on the mobile flexible robotic manipulator. (Refer to sections 5.2 and 5.3).
7. Experimental verification and comparison of the developed algorithms were carried out on the MFLM test rig. The results proved that the novel algorithm yielded better tracking and prediction of the vibration compared to the benchmarked methods. (Refer to sections 5.8 and 5.9, and the validation results in sections 6.4 and 6.5)

## **1.5 Outline of the thesis**

Chapter 2 presents a review of the sensors and the sensor fusion techniques, followed by the selection of sensors. It is then followed by a literature review of past works. Benchmarking of the fusion approaches were concluded.

Chapter 3 started with deriving the methodologies for testing the chosen sensors (accelerometer, camera and encoder). Next, it is the development of the models for the motor, flexible manipulator, and MFLM are developed. It then followed by deriving the methodologies for identifying the motor's constant and friction coefficients, as well as deriving the methodologies to test the motor model. Chapter 4 presented the results for the tests derived chapter 3.

Chapter 5 started with the development of the sensor fusion and prediction algorithms. Next, it is the fabrications of the flexible manipulator and the mobile platform. Following that is the deriving of the methodologies for comparing the simulated vibration with actual vibration for the flexible manipulator and the MFLM. Finally, it presents the deriving of the methodologies for testing the mobile MFLM with the sensors individually; with fusion system; and with prediction algorithm. Chapter 6 presented the results for the tests derived in chapter 5.

Chapter 7 is final conclusion and outlines the proposals for future works.

## 2. Review on past related works

This chapter covers the first research objective set in section 1.2, through a rigorous review of the available literature on motion and position estimation techniques.

The feedback signals for FLMs are mostly taken from the displacement of the tip, via the strain along the beam or the tip's acceleration. Methods for acquiring displacement of the tip includes: optical (e.g. vision cameras), range sensors, position sensitive devices and electromagnetic sensors. Strain gauges are commonly used for the strain measurement for estimating link deflections. Inertia navigation systems (accelerometers and/or gyroscopes) and odometers (encoder at the motor) are used to obtain acceleration and position measurements.

The approach to motion and position estimation can be broadly classified as relative or absolute methods, and their combinations. Absolute method estimates position using active beacon fixed at known locations and a receiver fixed on the robot [15]. Although this approach can provide accurate position estimation at fixed points, they suffer from signal out of range problems and usually have low sampling rate.

Relative methods determine the position by incrementally integrating the motion information over a period of time. These methods continuously provide motion estimation at high frequency, but they suffer from systematic and random errors integrated within the position measurements, and accumulated at an unbound rate.

In order to improve motion estimation of the FLM, two or more sensors are usually fused together using sensor fusion algorithms. Common fusion algorithms are weighted averaging, Kalman filter, Fuzzy inference system and neural network.

The control strategies for the trajectory of the FLM tip can be open-loop-schemes (feed-forward control) or closed-loop-schemes (feedback control) [14]. The open-loop-schemes require accurate model estimation where the controller controls the manipulator trajectory. However, without feedback, accurate control of flexible-link robotic systems having high level of vibrations is difficult. The closed-loop-systems utilize some forms of feedback signals for the controller, and by applying appropriate control strategies. There are several approaches utilising closed-loop control strategies

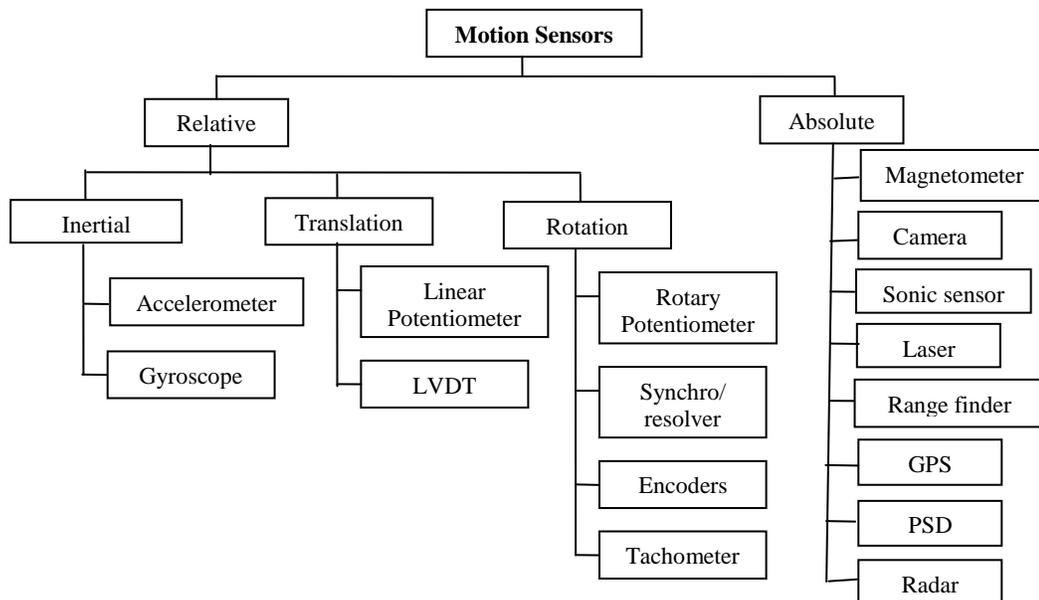
for flexible manipulators. The feedback controllers can be divided into collocated and non-collocated controls. Recent years, motion prediction control strategies for dynamical systems with time delay have been proposed in the attempt to find successful control. The model predictor control offers a good tool for dealing with time delay.

## 2.1 Sensors for Robotic Manipulators

The fundamental function of any sensor is the conversion of a physical phenomenon into quantitative electrical signals, as depicted in Fig. 2.1. For localization of robotic manipulator, the physical phenomenon is the movements of the manipulator and its vibrations. The localization approaches for robots can be broadly classified as either relative or absolute. Many partial solutions can be categorized into these two groups, or their combinations. Fig. 2.2 listed the sensors [15, 16] that represent a simple classification of the relevant sensors. The following sections provide a brief overview of these two groups of sensors.



**Fig. 2.1** Fundamental function of a sensor [16]



**Fig. 2.2** Types of motion estimation sensors (absolute method or relative method) [15, 16]

### **2.1.1 Absolute techniques for robot localization**

Absolute localization techniques determine the position of the robot with respect to a known reference. Localization via absolute methods is independent of time and the initial position, and is capable of maintaining accurate position estimation and free from cumulative errors or drifts.

Absolute approach for robot localizations provide constant accurate localization from transmitters (such as fixed beacons) which are placed at designated locations, with a sensor (known as receiver) fixed on board the vehicle to detect the position of the vehicle. This arrangement receives signals at low frequency rate and is prone to signal drop-out.

The main problems associated with absolute localization methods are:

- I. initial cost for installing markers (or beacons) in the environment where the robot operates;
- II. higher computational costs for landmark mapping;
- III. signal outage; and
- IV. low signal resolution.

Common absolute methods for robot localizations include active or passive beacon systems, GPS (Global Positioning System), map matching and landmark recognition. This research only considered landmark recognition method.

Landmark recognition involves recognizing distinctive features within the environment and tracking the motion of the robot relative to the landmark via a machine vision or camera [17, 18]. Landmarks have to be carefully chosen for easily identifiable objects in fixed locations. Commonly, artificial landmarks are designed for easy and optimal detectability in indoor environment, where natural landmarks are particularly difficult to identify. This approach is computationally intensive because the robot has to store numerous images for recognition, and not very accurate.

One method to improve landmark recognition is by placing predefined landmarks on the ceiling. Zhang et al. [19], Shih and Ku [20] and Lan et al. [21] uses landmarks on the ceiling. Artificial landmarks are placed on the ceiling to be used for mobile robot indoor

guidance system, where camera is placed on board of the robot to capture the landmarks. In this method there will not be obstruction or changes in the landmarks. Therefore, it will not have image lost issue in this method. QR codes were introduced, where information of the location are stored in QR codes which are strategically placed in the environment where the robot operate. The mobile robot equipped with high speed cameras focusing at the ceiling to read QR codes. The position of the robot is estimated based on the positional relationship between the robot and QR codes. This method is recognized as a benchmarking technique in this research..

### **2.1.2 Absolute measurement methods for flexible manipulator displacement**

This method, the displacement tracking is measured from a known location. There are three types of displacement measurement methods. One is where the sensor measures distance from target to the sensor, for example, range sensor. Another is where displacement measurement is based on the change in internal properties of the sensor. An example is the measurement of strain in the strain gauges [22]. The third type is based on displacement measurement from a point, such as using PSD or camera.

In this research, vision method is used for the displacement estimation of the flexible manipulator, while range sensor is used for benchmark measurement. Thus, only camera vision methods and range sensors are discussed as follows:

#### **A. Camera vision methods**

Vision systems are more reliable compared to sensors such as strain gauges [23], thus can be a feasible means for measuring the vibration of the flexible manipulator. They are not subjected expose to high level of noises. Therefore, vision devices have gained more research attentions in the recent years. References [23 – 25] are some of the research works that utilized cameras for control feedback to the FLM that brought promising results in vibration estimations. For mobile robot localizations, vision methods have been very extensively used [17, 18] for both in indoor and outdoor environments, where landmark mapping are most frequently used.

However, the disadvantages of the camera vision approaches are that they require stationary cameras and still background to assure good image quality, high resolution

and high speed camera to track rapid motion which very expensive [15]. This method requires large amount of traceable image features and sometimes rely on feature correspondences in multiple cameras. Other problems include discriminative image recognition due to silhouettes and other artifacts / objects, and inconsistency problems due to different lightings. Problems of limited view range, obstructions and interference. It also requires complicated image processing algorithms to analyze the data, detect and track objects.

The next major problem of position sensing based on vision is the delay between image capturing and image processing [15]. The resolution of CCD cameras could be inadequate, or do not have fast frame rate, which caused long processing time for the images resulting in significant low update rates and data lag. Additionally, optical devices require a free line of sight between camera and object, and they suffer from vision image drop-outs problem resulting in loss of information problem.

## **B. Range sensors**

Range sensors, can give accurate and good sensing at high rate, but it is not convenient to fix on moving robot arm as it is required to be fixed to a nearby reference object [14]. This approach can only be used for validation or calibration purposes. Lu et al. [26] used a renishaw laser interferometer for the tip position measurement of flexible beam travelling on linear motor. A linear encoder measures displacement of the base.

### **2.1.3 Comparison of absolute methods**

The important requirements for the sensors in this research are price, weight and power requirement. Table 2.1 lists some of the common sensors used for absolute estimation methods, and outlining their operating principles, common applications and estimation method. Table 2.2 lists their advantages and disadvantages.

**Table 2.1** Principle of operation and applications of absolute sensors

Sensor	Principle of operation, common applications and method	Ref.
Laser or infrared range finder (optical energy source)	<p>Based on Doppler effect: Phase shift in frequency of reflected beam is proportional to velocity of moving object.</p> <p>Commonly used for accuracy assessment of machine tools, vibration measurement and other position critical motion systems.</p> <p>Displacement method.</p>	27, 28
Position sensitive device	<p>Photodiodes detect the centre position of a light spot projected on a surface.</p> <p>Position, displacement and vibration sensing.</p> <p>Displacement method.</p>	29
Magnetometer / magnetic sensor	<p>Measure the direction on magnetic field in the vicinity of object.</p> <p>Measures the local earth's magnetic field.</p> <p>Commonly used in conjunction with IMU sensors for position and orientation estimation.</p> <p>Displacement method.</p>	27
Camera	<p>Position estimation by image recognition.</p> <p>Commonly used in motion estimations in people and mobile robots.</p> <p>Map matching and landmark recognition methods.</p>	17, 23, 24, 25, 30, 31
Sonic sensor	<p>Position estimation are based on time of arrival of the acoustic transient emitted from a source of interest or from sonic sensor and listen for echoes.</p> <p>Commonly used to detect distant objects or array of sensors position detection of surface damages. Also used in position estimation in an array of sonar sensors. Other uses include object detection and collision avoidance.</p> <p>Position measurement using beacons.</p>	27, 28

**Table 2.2** Advantages and disadvantages of absolute sensors

Sensor	Advantages / Disadvantages	Ref.
Laser or infrared range finder (optical energy source)	<p><b>Advantages</b></p> <ol style="list-style-type: none"> <li>1. High precision and accuracy.</li> <li>2. High and stable measurement frequency.</li> </ol> <p><b>Disadvantages</b></p> <ol style="list-style-type: none"> <li>1. Very expensive.</li> <li>2. Complex equipment.</li> <li>3. Air pressure, temperature and humidity can alter air's refractive index and change speed of light and wavelength.</li> </ol>	15, 27, 28
Position sensitive device	<p><b>Advantages</b></p> <ol style="list-style-type: none"> <li>1. High speed measurement.</li> <li>2. Highly precise measurements.</li> </ol> <p><b>Disadvantages</b></p> <ol style="list-style-type: none"> <li>1. Limited to small displacement.</li> <li>2. Expensive.</li> </ol>	29
Magnetometer / magnetic sensor	<p><b>Advantages</b></p> <ol style="list-style-type: none"> <li>1. Insensitive to vibrations.</li> </ol> <p><b>Disadvantages</b></p> <ol style="list-style-type: none"> <li>1. Low resolution up to 0.1mm.</li> <li>2. Susceptible to magnetic interferences.</li> <li>3. Distortions due to metal objects.</li> </ol>	27
Camera	<p><b>Advantages</b></p> <ol style="list-style-type: none"> <li>1. Captures detailed information about overall motion.</li> <li>2. Possible to add feature for obstacle detection and avoidance.</li> <li>3. Accurate</li> </ol> <p><b>Disadvantages</b></p> <ol style="list-style-type: none"> <li>1. Complex algorithms.</li> <li>2. Low data rate.</li> <li>3. Data delay.</li> <li>4. Line-of-sight problems.</li> <li>5. High dependent on background and lighting.</li> <li>6. Limited field of view.</li> </ol>	15, 17, 23, 24, 25, 30, 31
Sonic sensor	<p><b>Advantages</b></p> <ol style="list-style-type: none"> <li>1. Superior signal-to-noise ratio.</li> <li>2. Sensitivity at the ultraprecision scale.</li> <li>3. Tends to propagate at frequencies (typically kHz and MHz range) above the anticipated frequencies attributed to natural structural modes or machining.</li> <li>4. Propagation of the sound energy is not limited by obstacles which obstruct the line of sight.</li> <li>5. Passive, affordable, robust, and compact.</li> <li>6. Low cost and light weight</li> </ol> <p><b>Disadvantages</b></p> <ol style="list-style-type: none"> <li>1. Easily effected by unreliable acoustic source.</li> <li>2. Reflection of acoustic source from walls in environment.</li> </ol>	27, 28, 32

### **2.1.4 Relative methods**

In relative position estimation, the position and orientation uses information provided by sensors onboard the robot. The motion estimation is obtained by comparing current data with previous data [15]. There are two methods of relative localizations [14]: odometry and inertial navigation.

The errors of the sensors are contributed by systematic and non-systematic errors. Systematic errors are errors internal to the sensor itself, such as bias drift and poor sensitivities in low grade sensors or low sampling rates. Non-systematic errors are external errors, such as wheel slippage, incorrect measurement of wheel diameters, and flexural deformation along links.

#### **2.1.4.1 Odometry**

Odometry is usually the estimation of the position relative to a starting point of estimation [15]. It is widely used method in robotic positioning estimations. Encoders and potentiometers are common sensors used. The sensors are placed at the wheel shaft of vehicle or joint of robot arm.

#### **2.1.4.2 Inertial navigation**

Inertial navigation is localizations where acceleration signals are integrated to obtain velocity and position information [14]. Common examples are gyroscopes and accelerometers, which measure the rate of rotation or acceleration. Inertial sensors offer other advantages which includes non-radiating and non-jamming, so they can be used in harsh environments. Main problem is that data from inertial sensors are that they are filled with noises, resulting in unbounded accumulation of errors due to integrating of the acceleration signal, thus poor accuracy for long period estimations.

The key benefit is that the inertial rate sensor continues to maintain a useful tracking of relative position and do not have signal drop-outs issue [15]. Accelerometer provides high frequency signals about the vibration of FLM with good short term accuracy. Many researches utilize accelerometer signals as a benchmark to verify the sensor system being developed. For instance, the tip position was sensed by an accelerometer

and fed to the motor controller for a combined feedback and feed-forward control of a single flexible link manipulator system, as reported by Li et al. [33].

### 2.1.5 Comparison of relative methods

Table 2.3 lists some of the common sensors used for relative estimations and outlines their operating principles, common applications and measurement method. Table 2.4 lists their advantages and disadvantages.

**Table 2.3** Principle of operation, common applications and method for relative sensors

Sensor	Principle of operation, common applications, price range, weight and power consumption	Ref.
Linear/ rotary potentiometer	Consist of resistance element provided with movable contact for displacement measurement. Commonly used as position transducer, such as joystick. Odometry measurement method	27, 34
Tachometer	Measures the speed at which a mechanical device is rotating. Commonly used to measure rotation speed of shaft and disk, as in motor. Odometry measurement method	27
Absolute/ Incremental Encoder	Measure rotation speed of shaft and disk with directions. Commonly used to measure rotation speed and/or position of shaft and robotic arm. Odometry measurement method	27
MEMS Accelerometer	Sensing transducer provides output proportional to their acceleration. Commonly used in vehicle for speed and displacement measurement, especially crash measurement. Inertial navigation method	16, 34 35, 36, 37
MEMS Gyroscope	Angular rate measurement based on Coriolis effect. Commonly used for angular velocity. Inertial navigation method	16, 34, 35, 36, 37

**Table 2.4** Advantages and disadvantages for relative sensors

Sensor	Advantages / Disadvantages	Ref.
Linear/ rotary potentiometer	<p><b>Advantages</b></p> <ol style="list-style-type: none"> <li>1. High accuracy at short displacements.</li> <li>2. High resolution.</li> <li>3. Low cost</li> </ol> <p><b>Disadvantages</b></p> <ol style="list-style-type: none"> <li>1. Movable contacts are prone to wear and tear.</li> <li>2. Can only measure short displacement.</li> </ol>	27, 34
Tachometer	<p><b>Advantages</b></p> <ol style="list-style-type: none"> <li>1. Can measure high speed.</li> <li>2. Low cost.</li> <li>3. Simple and easy to install.</li> </ol> <p><b>Disadvantages</b></p> <ol style="list-style-type: none"> <li>1. Bulkier.</li> <li>2. Only measure one direction.</li> <li>3. Possible of missed count.</li> </ol>	27
Absolute/ Incremental Encoder	<p><b>Advantages</b></p> <ol style="list-style-type: none"> <li>1. Can measure high speed.</li> <li>2. Low cost.</li> <li>3. Simple and easy to install.</li> <li>4. Incremental encoder able to rotate through many revolutions.</li> </ol> <p><b>Disadvantages</b></p> <ol style="list-style-type: none"> <li>1. Possible to missed count.</li> <li>2. Bulkier.</li> </ol>	27
MEMS Accelerometer	<p><b>Advantages</b></p> <ol style="list-style-type: none"> <li>1. Low cost, compact, light, robust, small, low power.</li> <li>2. Can measure vibration, acceleration and tilt.</li> <li>3. Rapid data acquisition at high frequencies.</li> <li>4. Relatively accurate at short distance.</li> <li>5. Self-contained.</li> <li>6. Signals not affected by external noises.</li> </ol> <p><b>Disadvantages</b></p> <ol style="list-style-type: none"> <li>1. High bias noise density.</li> <li>2. Lower resolution.</li> <li>3. Require high enough frequencies (&gt;0.2Hz) for accurate acceleration measurement.</li> <li>4. Output influenced by gravity.</li> </ol>	16, 34, 35, 36, 37
MEMS Gyroscope	<p><b>Advantages</b></p> <ol style="list-style-type: none"> <li>1. Low cost, compact, light, robust, small, low power.</li> <li>2. Rapid data acquisition at high frequencies.</li> <li>3. Relatively accurate.</li> <li>4. Not influence by gravity.</li> <li>5. High signal to noise ratio.</li> </ol> <p><b>Disadvantages</b></p> <ol style="list-style-type: none"> <li>1. Lower resolution.</li> <li>2. Cross-coupling between drive and sense mode oscillations.</li> <li>3. Noisy output.</li> </ol>	16, 34, 35, 36, 37

## 2.2 Sensor selections

This section explains how the sensors were being selected for this research. The mobile manipulator investigated in this research consists of two physical components: a flexible beam and a mobile base. Table 2.5 proposes the possible absolute and relative methods for the motion estimations.

**Table 2.5** Appropriate absolute and relative estimation methods for motion sensing

Component	Motion path	Absolute method	Relative method
Beam displacement	1. Linear or angular displacement at tip.	1. Beacons 2. Landmark recognition 3. Map matching	Inertial navigation
Vibrating beam	1. Short reciprocating displacement.	1. Displacement measurement	Inertial navigation
Vehicle displacement	1. Linear or angular displacement.	1. Beacons 2. Landmark recognition 3. Map matching	Inertial navigation
Rotating wheel	1. Continuous rotation	Nil	1. Odometry 2. Inertial navigation

To select the right sensors for the manipulator system for use in the domestic environments, the important requirements for the sensors are outlined as follows:

- I. The sensor has to be light-weight.
- II. The sensor needs to be low cost.
- III. Output from the sensor should be able to be converted in displacement.
- IV. The sensor should be safe for use in domestic environments.
- V. The power requirement of the sensor should be low.
- VI. Measurement range: The sensor needs to be able to measure the upper and lower limit required for the application.
- VII. The sampling rate of the signal should meet the maximum sampling period of 50 msec or 20 Hz minimum sampling rate.

### **2.2.1 Selecting sensors in absolute methods**

Beacon, landmark recognition and map matching can only be used for position estimation of the mobile platform. Displacement method can be used for vibration measurement of the flexible beam. As only landmark recognition is adopted, this section outlines only the landmark recognition method and the displacement method for vibration measurement.

#### **A. Landmark recognition**

For the landmark recognition, the obvious choice of sensor system is vision-based via camera. Depending on the required resolution of the image captures, camera ranges from 15 frames per sec (fps) to 60 fps. A low cost choice is webcam, which provides 30 fps frame rate. The main drawback of webcams is that they offer lower quality images and relatively low frame rate. There is a large amount of memory usage and is computing intensive for this method. The strongest problem is its high sensitivity to the changes in pattern aspect (illumination, size and perspective). When capturing moving images, it is notorious that the images be blurred, making object detection difficult. Camera and webcam are chosen for localization of the robotic manipulator. Refer to section 2.3.1 for information about cameras and webcams.

#### **B. Displacement measurement sensors for sensing vibrating displacement**

For small vibrational displacements at the tip of flexible beam, range finder, position sensitive devices (PSD), strain gauges and cameras can be used.

Range-finders (or range sensors) are devices that use a laser beam to measure the distance to targets [28]. Laser pulse is sent towards the target and measuring how long it takes for the pulse to return to the device. Range sensors can give very accurate and high output rate for position measurements. The sample period is about 50 ms or 20 pulses per second sampling frequency (refer to Appendix B.3 for datasheet of IR distance sensor used in this research). However, range sensors can only give short distance measurements (less than 10 meters) and it is inappropriate to mount a range sensor at the tip of a manipulator. Due to the accurate measurement, it is usually only used for bench-

marking the vibration measurements, and is used in this research for benchmarking vibration of flexible manipulator and movement of mobile platform.

PSDs are capable of high speed measurements and high accuracy [29]. It has measuring rates of up to over 100 kHz. However, PSD may be too bulky to be mounted at the tip of a flexible beam.

Strain gauges are nearly weightless and have a small form factor, and it can provide measurement frequency up to a few hundred kHz [16]. However, they require amplifier circuitry and they are not sensitive to small deflections along the long and slender beam. To measure the deflection at the tip, often several strain gauges are needed to be placed at several locations along the beam to take measurements at each segment, but still may tend to be inaccurate.

### **2.2.2 Selecting sensor in relative methods**

Odometry and inertia sensors selections are outlined in this section.

#### **A. Odometry**

Odometry sensors are commonly used on joints and wheel shafts in conventional robots, giving accurate localization in the absence of slippage or vibration [38]. To meet the criteria of low power requirement, low cost, light-weight and high measurement rate; potentiometer, tachometer and encoder are good choices for measurement at the joints of the manipulator, as well as at the wheels.

However, odometry is notoriously unreliable, and it was not expected to be perfect. Potentiometer has limited number turns, thus not suitable to long distance measurement for wheels angular displacement. Tachometer can only measure wheel rotation and speed; it cannot provide direction of motion. Encoder can provide position, speed and direction of the angular displacement of the wheel, thus is best amount the three. Therefore, encoder is choice. There are two types of encoders; absolute and incremental. Incremental encoders are cheaper and are good for speed and acceleration measurements, but are not able to retain position information during start-up. See Appendix A for reviews on encoders.

## **B. Inertial navigation**

For the measurement of displacement at the manipulator tip, as well as motion of mobile base, accelerometers and gyroscopes offer low power consumption, low cost and lightweight solutions with relatively good speed and position estimations [39] over short distances. Accelerometers and gyroscopes offer very high sampling rate, thus are very useful for vibration measurement for the flexible link manipulator. The main problems of IMU sensors are that they are susceptible to noises which contribute significantly to uncertainty. Many research papers apply the combination of both gyroscope and accelerometer for robotic localizations or robot arm displacements. Gyroscopes can only offer angular estimations, while accelerometer is usually sufficient to offer both angular and translational estimations.

There are several types of MEMS accelerometers, while the most popular classes are piezoelectric and capacitive types [36]. The former has better accuracy at small sensing range, while the latter has larger sensing ranges with lower accuracies. A 3-axis piezoelectric accelerometer can provide up to 1.6 kHz output rate, thus is chosen. See section 2.3.2 for information on accelerometers.

## **2.3 Sensors introductions**

Camera and accelerometer are used in this research work for flexible beam vibration measurement, thus a brief introduction of these sensors are presented this section.

### **2.3.1 Camera and webcam**

Robot localization using cameras can either be fixing the camera at a stationary position to capture the moving robot or by fixing the camera on the robot to capture the features (or marker) in its environment [18 – 21]. By using high speed camera, camera can also capture vibration of a beam. By computing the pixels from the reference point in image, the robot position can be estimated.

#### **A. Review of cameras**

In recent years, digital cameras have become very cheap and they have opened up new possibilities as a sensor for robot perception. Computer vision concerns with artificial

systems that extract information from images taken using the camera [40]. a 3 dimensional scene is projected onto a 2 dimensional plane when an image is recorded through a camera. Cameras capture a video stream as a sequence to images or frames. In a computer vision system, a software program extracts individual frames and performs position or motion estimations. The process for estimating motion or position is the determination of motion vectors that describe the transformation from a 2D image to another image extracted from adjacent frames of a video sequence.

The motion vectors may relate to the whole image (global motion estimation) or specific parts, such as a specific shape of an object. Translational models or many other models that can approximate the motions and represent the motion vectors. Pixel based methods ("direct") and feature based methods ("indirect") are methods for finding motion vectors [40].

## **B. Type of cameras**

Cameras can be categorized into CCD (charge-coupled device) cameras and CMOS (complementary metal oxide semiconductor) cameras [40]. Both types contain image sensors that convert light into electric charges and process this into electrical signals. After the light is converts into electrons, the sensor reads the value (accumulated charge) of each cell in the image. The differences between the two sensors are:

1. A CCD is an analog device [41, 42]. In a CCD sensor, the charge on every pixel is transferred through a very limited number of output nodes (often just one) and converted to a voltage. Small electrical charges in each photo sensor are generated when the light strikes the chip. The charges are read from the chip and converted to voltage value one pixel at a time. Each of the pixel's value are then converted into digital value via analog-to-digital converter by measuring the amount of charge at each photosite and converting this measurement to binary form.
- A CMOS chip is a type of active pixel sensor made using the CMOS semiconductor [41, 42]. In a CMOS sensor, each pixel has its own circuitry next to each photo sensor which performs a charge-to-voltage conversion. The sensor may include additional circuitry such as amplifiers, noise-correction and

digitization circuits, to convert the voltage to digital data. These functions caused higher design complexity and minimise the area available for light capture.

### **C. Comparison of the sensors**

The advantages and disadvantages of the CCD and CMOS devices are outlined below [41, 42]:

1. In a CCD device, all of the pixels can be allocated to light capture, and has high output uniformity, thus CCD sensors create low-noise, high-quality images. In a CMOS device, the design complexity reduces the area available for light capture. Each pixel is doing its own conversion, resulting in lower uniformity. Hence, CMOS sensors are more susceptible to noise.
2. CMOS sensors consume lower power, while CCD sensors consume high power, which can be as high as 100 times more power than the CMOS sensor. CMOS sensors are therefore ideal for portable devices.
3. CCD sensors are more mature and tend to have higher quality and more pixels due to mass production for a longer period of time.
4. CMOS sensors offer smaller system size than CCD sensors due to more integration which offer more functions on the chip. CCD cameras are usually relatively heavy and large.
5. CMOS chips tend to be lower cost than CCDs because they are fabricated on standard silicon production lines.
6. CMOS contain less than 70% active sensors, while the CCD contains 100% active sensors. Therefore, CMOS is less sensitive than CCD due to CMOS contain very much lesser fill factor the CCD chips.
7. CCD is better for low contrast images, but the pixel resolution is high ranging from 1.4 megapixels to 16 megapixels. The frame rate can be from 0.1 fps (frames per second) to 2300 fps [40]. CMOS is much more flexible than CCD. CMOS sensors can be windowed to read out less data at a higher frame rate. For instance, a readout rate of 15 frames per second can be achieved with sensor having a resolution of 1280 x 1024. Nearly 70 frames per second could be achieved by windowing the sensor and only reading out a 640 x 480 portion of the image.

8. CMOS can provide faster readout than CCDs by having fewer components.
9. When the sensor is overloaded, CCD sensors are more susceptible to vertical smear from bright light sources. CCDs which do not suffer from this problem due to high-end frame transfer.

As such, CCDs tend to provide high-quality images with lots of pixels and excellent light sensitivity. CMOS sensors have lower resolution, lower sensitivity and lower quality. CMOS designers are making higher image quality CMOS sensors [40], while CCD designers are lowering the power requirements and pixel sizes for CCD sensors. Therefore, neither technology has a clear advantage over each other and selection must be determined for each specific application.

#### **D. The lens**

Another important component for the camera is the lens. There are three types of lens [41]: Manual-focus lens, auto-focus lens and fixed-focus lens. Manual-focus lens require human operator for adjusting the focus point. Auto-focus lens require expensive mechanical parts, more power and are heavier.

A photographic lens for which the focus is not adjustable is called a fixed-focus lens [41]. Fixed-focus lens do not require expensive electronics, moving parts or consume power. The focus is set at the time of manufacture and remains fixed. A fixed-focus lens relies on sufficient depth of field (portion of the image that appears sharp) to produce acceptably sharp images, thus does not required to determine the correct focusing distance and setting the lens to that focal point. It is acceptable for cameras used for capturing images of objects further away than a meter by setting the hyperfocal distance, so that the depth of field ranges all the way down from half the hyperfocal distance to infinity. This system is automatic and it can also be more predictable than the auto-focus lens.

The disadvantages of fixed-focus lens include the unable to produce sharp images compared to lens that has been set to the match the focal point for a scene [41]. For close-up objects (within 2.4 - 3.7 meters) it is not possible to achieve a completely sharp image.

To increase the depth of field, most cameras without fixed-focus lenses also have a small aperture. However, small apertures reduce the amount of light incident on the chips, this presents severe limitations for imaging of fast-moving objects which require short exposure times.

### **E. Webcams**

Webcams, surveillance cameras and camera phones usually use fixed-focus lenses with low resolution CCD cameras [43]. The low resolution of the detector allows a loose focusing on the CCD without noticeable loss of image quality. As such, the circle of confusion (an optical spot caused by a cone of light rays from a lens not coming to a perfect focus when imaging a point source) gets bigger and hyperfocal distance (a distance beyond which all objects can be brought into an "acceptable" focus) smaller.

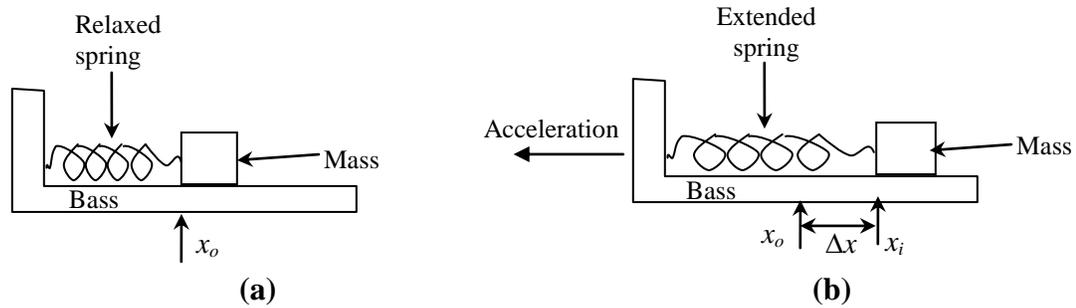
There are various lenses available for webcams, while most webcams use plastic lens that can be screwed in and out to set the camera's focus. The image sensors are either CMOS or CCD, with CMOS being cheaper [42]. Most commercially available webcams can provide VGA (Video Graphics Array)-resolution video at frame rates up to 30 frames per second [43]. Current newer devices can have video in multi-megapixel resolutions, running at frame rates as high as 120 frames per second, producing a 320×240 video.

### **2.3.2 Accelerometer**

MEMS accelerometers are low cost, low power consumption, small and light [36]. They are non-radiating, self-contained, dead-reckoning and non-jamming. They can provide direct measurements of dynamic information [44]. However, MEMS accelerometers are typically limited in accuracy and measurement ranges, as well as being susceptible to noise and uncertainties. One example is Johnson noise [45] which is associated with the device's mechanical resistance due to their small size. These uncertainties lead to substantially large amounts of measurement drift during double integration to yield position information. Based on these factors, accelerometers alone are not suitable for long range measurements.

## A. Principle of operation of accelerometers

Accelerometers are sensors that provide an output proportional to gravitational acceleration [27]. The measurement of acceleration relies upon the Newton's second law of motion for mass acceleration and Hooke's law for spring action [36]. Fig. 2.3 outlines the principle of operation for the sensing element of an accelerometer [36].



**Fig. 2.3** Principle of basic mass spring system of an accelerometers sensing element, a) No acceleration applied, b) Acceleration applied [36]

Newton's law states that with a constant mass, the force is equal to the product of the mass of the body and its acceleration,  $F = ma$ . Hooke's law states the amount by which a spring with constant  $k$  is deformed from its equilibrium position  $x$  is linearly related to the force acting on the spring, and is given by [36]:

$$F = -k_{sp}x \quad (2.1)$$

Combining Newton's law and Hooke's law, the acceleration can be obtained as [36]:

$$a = k_{sp}x/m \quad (2.2)$$

where

$m$  = mass of object, (kg)

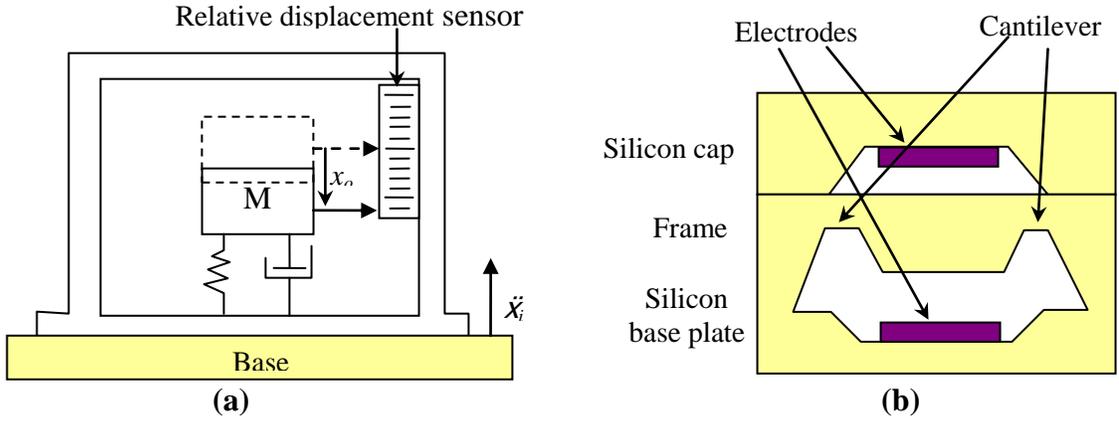
$x$  = distance that the spring has been stretched or compressed, (m)

$F$  = applied force or acting force, ( $\text{kgm/s}^2$ )

$k_{sp}$  = force constant (or spring constant)

$a$  = acceleration of the object

Micromachined accelerometers are light and very stiff [29]. Therefore they have small sensitivity and damping ratio, but have a high natural frequency. Fig. 2.4b [36] depicts a micro-machined capacitive silicon accelerometer.



**Fig. 2.4** A mass-spring system of second-order, underdamped sensors. a) The proof mass displaced by the acceleration applied to the housing. b) Micro-machined capacitive silicon accelerometer [36]

As illustrated in Fig. 2.4a, the accelerometer is a second-order system which contains one energy dissipating element and two energy-storing elements. The input  $x_k$  and output  $y_k$  are related by a second-order linear differential equation of the form [36]:

$$a_2 \frac{d^2 y_k}{dt^2} + a_1 \frac{dy_k}{dt} + a_0 y_k = x_k \quad (2.3)$$

The corresponding transfer function is [36]:

$$\frac{Y(s)}{X(s)} = \frac{k w_n^2}{s^2 + 2 \zeta w_n s + w_n^2} \quad (2.4)$$

where  $\zeta$  is damping ratio,  $k$  is static sensitivity, and  $w_n$  ( $2\pi f_n$ ) is the natural undamped angular frequency of the sensor. The frequency response of the accelerometer is determined by the resonant frequency ( $w$ ), which can be estimated by:  $w_o = \sqrt{k/m}$  [45]. Combining equation (2.1) and  $w_o$ , the sensitivity and displacement per g of acceleration can then be determined as [36]:

$$x_g = \frac{mg}{k} = \frac{g}{w_o^2} \quad (2.5)$$

where:

$$x_g = \text{displacement of seismic mass}$$

$m$  = seismic mass (or prove mass, see Fig. 2.4a)

$k_{sp}$  = spring constant of the device

$g$  = acceleration,  $9.81 \text{ m/s}^2$

$w_o$  = angular resonant frequency

In the capacitive silicon accelerometer, the force exerted by the proof mass due to the applied acceleration cause flexes of the cantilevers. This changes the capacitance between that mass surface and the fixed electrodes [36].

A photodetector (with an ancillary light source and a shutter), an inductive, a capacitive sensor or a potentiometer can measure the displacement  $x_o$  (see Fig. 2.4a) of the proof mass ( $M$ ) with respect to the armature fixed to the element undergoing an acceleration  $\ddot{X}_i$  [36]. Alternatively, the stress of a flexing element holding the mass can be sensed, such as strain gages or a piezoelectric element. By Newton's second law, the force on the mass is communicated through the spring deflection (by Hooke's law) and the internal viscous friction. The force equation of the system is [36]:

$$M(\ddot{X}_i - \ddot{X}_o) = KX_o + B\dot{X}_o \quad (2.6)$$

where  $K$  is the spring constant or stiffness and  $B$  is the viscous frictional coefficient. Performing Laplace transform of  $\ddot{X}_i = s^2X_i(s)$ , yields:

$$M s^2X_i(s) = X_o(s)[K + Bs + Ms^2] \quad (2.7)$$

The transfer function is [36]

$$\frac{X_o(s)}{\ddot{X}_i} = \frac{X_o(s)}{s^2X_i} = \frac{M}{Ks^2 + s\left(\frac{B}{M}\right) + K/M} \quad (2.8)$$

We need to consider also the gravitational acceleration, where the accelerometer's axis is affected by an angle with respect to the horizontal plane, with the term  $Mg \sin \theta$ . The output  $y_k$  would then be defined as  $x_o + (Mg \sin \theta)/K$ .

For measurement of acceleration, the response is low-pass and  $w_n$  need to be higher than the maximal frequency variation of the acceleration to be measured. For measurement of vibrational displacement—high-pass response— $w_n$  must be lower than the frequency of the displacement and there is no dc response [36].

## **B. Types of accelerometers**

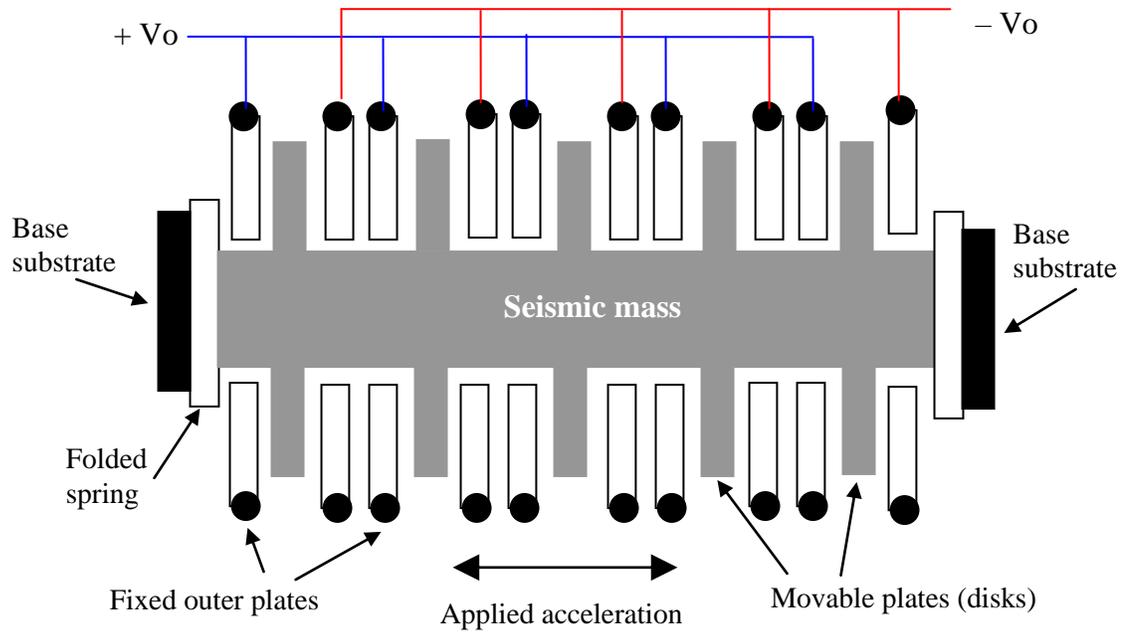
The types of MEMS accelerometers include capacitive, piezoresistive, electromagnetic, piezoelectric [35] and strain gage [45]. Three of the common types of MEMS accelerometers are piezoresistive, piezoelectric and capacitive accelerometers. The most popular class of MEMS accelerometers is the piezoelectric accelerometer, due to the simplicity of the sensor element more successful types are based on capacitive transduction, i.e. no requirement for exotic materials, have low power consumption, and good thermal stability. The output can be analogue, digital, or ratiometric; supplying voltage or any of various types of pulse modulation.

## **C. Comparison of piezoresistive, piezoelectric and capacitive accelerometers**

Capacitive-based MEMS accelerometers have achieved more commercial success than piezoresistive and piezoelectric designs [36], and are suitable for measuring low-frequency motion and steady-state acceleration. The piezoelectric types of accelerometers are widely used for vibration and shock measurements. Piezoresistive accelerometers are desirable for shock measurement but less useful for vibration due to their low sensitivity.

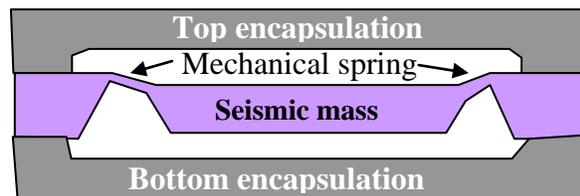
Compared to the capacitive accelerometer, the piezoresistive accelerometer has low sensitivity, and thus requires an amplifier circuit, interfacing the piezoresistive accelerometer to a DAQ system [36]. The piezoresistive accelerometers have noise levels 25 times lower than the capacitive accelerometers, due to the higher resolution of capacitive changes. Accelerometers' noise sources are attributable to Hooge noise (inversely proportional to the frequency) and Johnson noise (constant over all frequencies).

Capacitive accelerometers [37] rely on the changing separation of capacitor plates. The seismic element in the form of a disk with spiral elements attached, as shown in Fig. 2.5. The disk is sandwiched between the capacitor plates. The movement of air through holes in the disk provides the damping, the spiral elements provides the spring force. The movement of the proof mass disrupt the balance of the differential capacitor for the capacitive accelerometers.



**Fig. 2.5** Schematic of capacitive accelerometer [37]

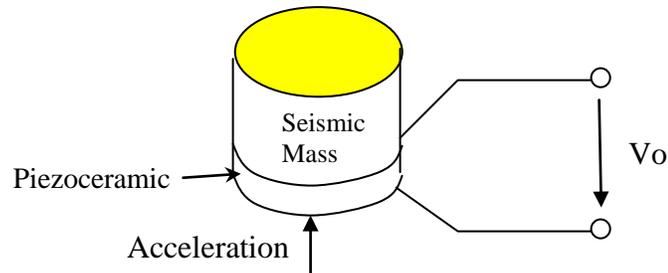
For Piezoresistive accelerometers, the mass movement is identified by relying on strain induced within a flexural element that attaches the proof mass to the sensor housing, as shown in Fig. 2.6 [36]. A micro-machined silicon mass suspended by multiple beams from a silicon frame is the sensing element. The motion of the suspended mass changes the strain in the beams cause the change in resistance for the piezoresistors located in the beams.



**Fig. 2.6** Schematic of piezoresistive accelerometer [36]

Piezoelectric accelerometers rely on disk compression of the piezoceramic material sandwiched between two electrodes, as shown in Fig. 2.7 [36]. When the accelerometer is subject to acceleration, a force is generated which acts on the piezoelectric element, producing a charge proportional to the applied force. There are two types of

piezoelectric sensor: low impedance and high impedance types. Low-impedance types use the piezoelectric sensing element and also incorporates a miniaturized built-in charge-to-voltage converter. High-impedance also have piezoelectric sensing element that need a charge amplifier or external impedance converter for charge-to-voltage conversion. External power supply is required to energize the electronics.



**Fig. 2.7** Principle of piezoelectric accelerometer [36]

## 2.4 Sensor fusion techniques

To the knowledge of the author, at present there is no single sensor that is sufficiently accurate to determine displacement information over a long period. Various sensor fusion techniques that were applied by researchers in the past are reviewed here. The following fusion techniques are briefly discussed:

1. Weighted averaging
2. Kalman filter
3. Complimentary filter
4. Particle filter
6. Fuzzy inference system
7. Neural network

### 2.4.1 Sensor fusion techniques – Weighted averaging

Weighted averaging [46] method is an approach for homogeneous sensors where various sensor data values or their interpretations are put together and taking the weighted average to arrive at a composite fused signal. Here, given  $N$  sensor readings  $x_1, \dots, x_N$ , parameters  $w_1, \dots, w_N$  where  $\sum w_i = 1$  are used to find a fused sensor reading  $\sum w_i x_i / \sum w_i$ .

Judicious choice of  $w_i$  can compensate for sensors with different accuracy and reliability. Letting  $x_k$  to be the latest position measurement, the fusion algorithm for weighted averaging is [47]:

$$x_k = \frac{\sum w_i x_i}{\sum w_i} \quad (1)$$

Weighted averaging is simple and efficient for sensor fusion. This method provides a quick solution to fuse multiple sensors. It is not computationally demanding and thus is suitable for real-time position estimation. Unfortunately, it ignores the noise factors involved and does not compensate for sensor failures. Furthermore, the input signals need to be of similar type. The technique also fails to provide any measure of the accuracy of the value obtained, and it does not give good accuracy [48].

#### **2.4.2 Sensor fusion techniques – Kalman filter**

Kalman filter (KF) is a well known method used in the theory of stochastic dynamic systems [49, 50 51], widely used for multi-sensor fusion in motion tracking. KF is an optimal linear estimator based on an iterative and recursive process, which can be used to improve the quality of estimates of unknown quantities. It recursively evaluates an optimal estimate of the state of a linear system. This estimation algorithm resembles that of a predictor-corrector algorithm.

KF is an improvement compared to the weighted average approach, but it is more sophisticated. Like Weiner filter theory [52], KF requires the characteristics of each signal to be completely specified. The difference being, Weiner filters use constant gains, while KFs contain time varying gains which are derived using the Kalman gain matrix [49]. A physical process model is required for the KF for the estimation process. The input is related to the output through a differential or difference equation. KF can be applied to a vast class of problems having multiple inputs and outputs including complex process and measurement relationships due to its matrix formulation. Due to their recursive nature, KFs are applicable to implementation on a computer. Only the most recent data and measurements are needed for a new estimation of the state of the system.

However, as Kalman filter is based on a linear process model and measurement equation, a nonlinear process will be no longer optimal. So, KF is only suitable if sensor

data is linear. For nonlinear process, a linearized or extended Kalman filter (EKF) can be used [49]. EKF can estimate the states of nonlinear systems using a linearized approximation, which is linearised around the most recent estimate. Jacobian matrix is requires to compute the state vector.

However, EKF gives poor performance if it has a significant amount of nonlinearities, or if the noise is non-Gaussian [50]. The bad estimates could potentially become worse and lead to an eventual divergence of the filter. This situation may be more significant when the measurement errors and initial uncertainty are large. Therefore, the measurement for sensors, such as those from inertial sensors, contains noises which are usually non-Gaussian [49], making this type of linearized models less promising.

KF and EKF are therefore optimal only for fusion of multiple sensors if the dynamic motions are linear and the sensor noises are Gaussian, otherwise it will produce unacceptable results due to its linearization process. It is computationally more demanding due to the complexity of these types of filters [49]. The complexities of the nonlinear models involved may delay updated state estimates. Furthermore, bad data due to sensor failure or outliers in the data will not be detected [53]. The only difference between KF and EKF is that the latter require linearizing the non-linear function.

### **2.4.3 Sensor fusion techniques – Complementary filters**

Weiner and KF are based upon the assumption of knowing the spectral characteristics, which is often difficult to satisfy in most practical applications. KFs rely heavily on measurement statistics and accurate process model, while complementary filters does not require complete statistical data regarding the signals and therefore more robust [54].

Compared to KF, the formulation of a complementary filter is more straightforward. With a lower computational overhead, complementary filter can produce estimates with accuracy comparable to that of KF, at shorter development time [55]. Complementary filters are commonly designed to combine multiple measurements of the same signal in a complementary fashion. The goal is to minimize the square of the expected error.

A complementary filter for attitude estimation performs low-pass filtering on low-frequency attitude estimates and high-pass filtering on a biased high-frequency attitude

estimate, and fuses these estimates together to obtain an all-pass estimates of attitude [56].

Vasconcelos et al. [54] used discrete-time complementary filter for attitude and position estimation of autonomous surface craft based on the sensor data from rate gyros, accelerometers and GPS. It was able to achieve the good performance. However, complementary is usually used for pitch and roll angle estimation.

#### **2.4.4 Sensor fusion techniques – Particle filter**

Particle filters (PF), are sophisticated model estimation techniques based on simulation [57]. It is also known as sequential Monte Carlo. PF is an alternative to the EKF or Unscented Kalman filter (UKF). The advantage of PF is when there are sufficient samples, PF can be used to estimate the nonlinear problems, and PF can be made more accurate at the price of additional computational effort.

Linearized models are usually not good solution for problems with nonlinearities and non-Gaussian characteristics, while PF [58] can provide a general solution to these problems. PFs do not require a fixed computation time. With the available of computational resources, their accuracy increases. PF can be quite easy to implement. Non-linear models can be implemented without linearizing the model. Also, problems with state variables having hard constraints can be incorporated in the estimation without any difficulties [58]. So this method can handle improvements of non-linear data using models developed by system identification methods. But, the computational load for the PF increases with the complexity of the problem, e.g. the number of states [57]. Exponentially many particles in  $d$  are required to populate a  $d$ -dimensional space.

There are two categories of PFs: those that require resampling on re-used particles to prevent divergence, and those that do not require no resampling because no re-use of particles [58]. Resampling it may lead to a loss of diversity and contain many repeated particles, causing large estimation errors [57]. A small process noise can cause all particles collapse to a single particle within a short period of time  $t_k$ . Due to the Monte Carlo method, PF contributes to large computational cost [58]. Therefore, PFs are not suitable for complicated applications.

#### **2.4.5 Sensor fusion techniques – Fuzzy inference system**

Fuzzy logic introduces the notion of a partial set membership [59]. An associated value is assigned to every member of a fuzzy set. It uses set membership values that range between 0 and 1, and in its linguistic form.

A complex nonlinear input-output relation can be realized in the fuzzy inference system (FIS) as a synthesis of multiple simple input-output relations [60]. It uses a series of if-then rules to approximate closely any nonlinear input-output mapping, where each rule describes the simple input-output relation.

FIS can be used in conjunction with KF/EKF for solving non-random uncertainty problems, where fuzzy subsets detect faulty sensor data, improving the reliability of KF/EKF for sensor fusion [61]. FIS is often used with neural network as Adaptive neuro-fuzzy inference systems.

However, FIS derived for one application may not be applicable for use on similar applications but having different condition, because under different operating conditions the set of rules will be different [60]. So, the rules may need to be changed from one condition to another. Therefore, the designer has to understand the behaviour of system before he can design the rules for the FIS.

#### **2.4.6 Sensor fusion techniques – Neural network**

In neural networks, artificial neurons are used connect and store information in weights [62]. Each individual neuron takes weighted inputs, and then performs a simple function which produces an output. The advantage of neural networks is that the knowledge of the uncertain source is not required. The network indirectly obtains the output using training data [63]. The input data can processed directly from sensor signals quickly.

Neural Networks are often used together with fuzzy logic, forming adaptive neuro-fuzzy inference systems (ANFIS), using the mathematical properties of ANN (Artificial Neural Network) to tune the rules based fuzzy system that estimate the complex unknown information [64]. During the training period when the sensors signals are received, the consequent linear parameters, determined by the fuzzy systems, are determined using means of least squares methods. ANFIS models will tune the

parameters describing the membership functions for each input parameters if the predicted mean square error does not meet the required error, until the training target is achieved.

When comparing the ANN and ANFIS algorithm with KF (kalman filter) [65, 66], ANN and ANFIS showed to have higher precision and robustness. KF produces error readings when noise and change of sensor state were introduced into the simulations. ANN and ANFIS were able to maintain reliable accuracies, and can adjust the system to adapt to changes contextual information. The fusion of sensors can be made practical and effective enough to be autonomous and adaptive to the uncertainties.

However, if the incoming data is not within the boundary of the trained data set or not within the inference rule, and if there is too much non-linearities and uncertainties, the training sets could become overloaded resulting in over-fitting of the network [67]. Consequently, the network does not correctly generalize the input-output relationship, but only recognizes those instances that were being trained the network. The trained data sets do not represent the overall population of data accurately, which can results in faulty networks producing false output.

#### **2.4.7 Comparison of sensor fusion techniques**

For the purpose of this research, it is mandatory that the fusion algorithms be suitable for integrating motion sensors. Table 2.6 depicts the advantages and disadvantages of fusion algorithms that could be applicable for the sensors used in the current research.

**Table 2.6** Advantages and disadvantages of fusion algorithms

Fusion algorithm	Advantages / Disadvantages	Ref.
Fuzzy inference system	<p><b>Advantages</b></p> <ul style="list-style-type: none"> <li>▪ Interpretation capability, which mimic human decision making.</li> <li>▪ Rapid computation due to intrinsic parallel processing nature.</li> <li>▪ Can handle imprecise and imperfect information.</li> <li>▪ Ease of encoding a priori knowledge</li> <li>▪ Robust as uncertainties are used in formulating the system.</li> <li>▪ Simplicity and flexibility.</li> <li>▪ Can model nonlinear functions of arbitrary complexity.</li> </ul> <p><b>Disadvantages</b></p> <ul style="list-style-type: none"> <li>▪ Lack of learning capabilities.</li> <li>▪ Imprecision and incomplete data approximation.</li> <li>▪ Cannot solve problems with no known answer.</li> <li>▪ Extensive verification and validation are required.</li> <li>▪ Highly abstract and heuristic.</li> <li>▪ Lack of self-tuning and self-organizing mechanisms.</li> </ul>	59, 60
Neural network	<p><b>Advantages</b></p> <ul style="list-style-type: none"> <li>▪ Not sequential or necessarily deterministic.</li> <li>▪ Data relationship not required to know.</li> <li>▪ Able to self-tune.</li> <li>▪ Can handle relationships with dynamic or non-linear problems.</li> <li>▪ Applicable to model various systems which is difficult or impossible to explain.</li> </ul> <p><b>Disadvantages</b></p> <ul style="list-style-type: none"> <li>▪ "Black box" nature, rely on trail-and error to find hidden layers and nodes.</li> <li>▪ Greater computational burden, the back-propagational networks also tend to be slower to train.</li> <li>▪ Cannot to handle linguistic information.</li> <li>▪ Imprecise or vague information cannot be managed.</li> <li>▪ Cannot resolve conflicts.</li> <li>▪ Cannot combine numeric data with logical or linguistic data.</li> </ul>	62, 63, 65, 68
Neuro-Fuzzy	<p><b>Advantages</b></p> <ul style="list-style-type: none"> <li>▪ Contradictory requirements in fuzzy modeling: interpretability versus accuracy.</li> <li>▪ All kind of information is possible.</li> <li>▪ Can manage imperfect, imprecise, partial or vague information.</li> <li>▪ Capabilities self-learn and self-tune.</li> <li>▪ No prior knowledge of relationships of data needed.</li> <li>▪ Fuzzy number operations yield fast computation.</li> <li>▪ Possible to promoting implicit and explicit knowledge integration.</li> <li>▪ Fuzzy rules can be used to extract knowledge.</li> </ul> <p><b>Disadvantages</b></p> <ul style="list-style-type: none"> <li>▪ Number of inputs allowed and/or to the limited form to create their own structure and rules.</li> </ul>	64, 66

**Table 2.6** Advantages and disadvantages of fusion algorithms (continued...)

<b>Fusion algorithm</b>	<b>Advantages / Disadvantages</b>	<b>Ref.</b>
Complementary filter	<p><b>Advantages</b></p> <ul style="list-style-type: none"> <li>▪ Not computational complex.</li> <li>▪ Has no singularity problem.</li> </ul> <p><b>Disadvantages</b></p> <ul style="list-style-type: none"> <li>▪ Nonlinear characteristic result in large steady state errors when moved rapidly through large orientation angles.</li> </ul>	54, 55
Weighted average	<p><b>Advantages</b></p> <ul style="list-style-type: none"> <li>▪ Simple and efficient</li> <li>▪ Proportional to accuracy of sensors or to credibility of sensor information</li> </ul> <p><b>Disadvantages</b></p> <ul style="list-style-type: none"> <li>▪ All the input data type need to be similar</li> <li>▪ Ignore noise factors involved and does not compensate for module failures.</li> </ul>	46, 47
Kalman filter	<p><b>Advantages</b></p> <ul style="list-style-type: none"> <li>▪ Useful when state vector can be identified and related to its previous value through state transition matrix.</li> <li>▪ Simple if small system matrix.</li> <li>▪ Good result if no sensor fault.</li> </ul> <p><b>Disadvantages</b></p> <ul style="list-style-type: none"> <li>▪ Can be problematic if noise is not Gaussian.</li> <li>▪ Sensor failure is not tolerated.</li> <li>▪ Sensitive to outliers.</li> <li>▪ Overload computation if more than 3 states</li> <li>▪ Gives optimal solution for Gaussian noises</li> </ul>	49, 50, 51
Particle filter	<p><b>Advantages</b></p> <ul style="list-style-type: none"> <li>▪ Can handle non-Gaussian and non-linearities noises.</li> <li>▪ Handles heavily skewed probability density functions (pdfs).</li> <li>▪ Handles bimodal/ multimodal pdfs.</li> </ul> <p><b>Disadvantages</b></p> <ul style="list-style-type: none"> <li>▪ The computational load increases with the complexity of the problem, e.g. the number of states.</li> </ul>	57

## 2.5 Literature review of past works

This section reviews past works related in measurement and prediction of mobile robot localizations and flexible manipulator displacements. Due to their extended workspace and better kinetic flexibility as compared to fixed based manipulators, mobile robotic manipulators have been extensively discussed in the recent literature with wide range of applications. The main challenge is the instability and inaccuracy of the robotic system to control the tip of the mobile flexible manipulator owing to complex dynamics and

singularity of the mobile platform and the robotic arm. A number of research publications have made effort to guarantee the stability and robustness in controlling the end-effector of the flexible manipulators. Following are some literature literature of past works:

#### **i. Meta-heuristic optimization method**

Esfandiar et al. [69] presented a mobile manipulator with flexible links and joints that moves from point to point via planned optimal path. Meta-heuristic optimization methods were applied to plan the optimal path. The mobile manipulators having large deformation in links were considered. They considered maximum load carried by manipulator and minimum transmission time for the optimization. However, the time cost of the calculation will rapidly increase when using the aforementioned methods if the number of the path points increases. Furthermore, some heuristic algorithms rely on finding configuration settings through trial and error, resulting in complexity and the result may be only locally optimal.

#### **ii. Finite-time tracking control**

Wei and Wu [70] presented a finite-time tracking control for a mobile manipulator with affine and holonomic constraints. The finite-time controller was designed to ensure output tracking errors of closed-loop system converge to zero in finite time. However, this algorithm assumes that the platform velocity remains constant or known. If the platform velocity and the vibration at the end-effector are unknown, the error will not converge to zero at finite time. The prediction error is sensitive to the prediction horizon and the accuracy of modeling the system.

#### **iii. MPC method**

Predictive controllers were developed in many research studies. Examples are MPC (model-predictive control) and FE (finite element) MPC. Abdolvand and Fatehi [71] presented a model-based prediction for vibration suppression of a flexible manipulator. Dubay et al. [22] utilized finite element based prediction to evaluate the behaviour of the flexible beam. Boscarriol and Zanotto [72] proposed a model predictive control strategy applied to five flexible-link mechanisms trajectory tracking for compliant mechanisms with effective vibration suppression. Avanzini et al. [73] exploited MPC approach to

compute an online suitable trajectory for both the base and the arm of a mobile manipulator. Model predictive control is used to find the most appropriate path towards desired pose of the manipulator. Generic MPC used to find the sequence of the future control variables.

The issue is that in order to guarantee convergence the weight matrices of the MPC problem have to be thoroughly analyzed. A good knowledge of the system is required to design the model, otherwise it will result in unstable control. Bakhti [74] developed an Extended Kalman filter observer to synthesize using the linear model of the flexible beam to predict the response of a beam. Wei and Liu [75] made use of the previous feedback trajectory and the feedback lag to calculate the corrected reference trajectory for flexible link manipulators. Combination of [74] and [75] can be the benchmark for used in this research.

#### **iv. New displacement approach**

Heidari et al. [76] presented a new displacement approach to determine the optimal trajectory of a large deflection flexible manipulator on the point-to-point motion. Nonlinear finite element model for the dynamic analysis was employed to describe the nonlinear model of the flexible manipulator. The Pontryagin's minimum principle was used to obtain the optimality conditions. However, the strategy is a feed-forward control for trajectory planning. Open loop control was used for the vibration modes of the flexible manipulator. Therefore, there would still be large residual vibrations that need to be controlled.

#### **v. Kalman filter approach**

To suppress the effect of noise that corrupt either the process dynamic model or output measurements, Bakhti [74] proposed Kalman filter on a multivariable model-based predictive controller to damp out the mechanical vibration of a flexible one-link manipulator using state variables feedback. Lagrange equations were used to model the flexible manipulator. Feedback measurements were taken from the joint angular position and vibration velocity of the tip. The simulation results demonstrate the efficiency of the Kalman filter to suppress the effect of noise that corrupt either the process dynamic model or the outputs measurements.

Reddy and Jacob [77] employed Kalman filter as state estimator to estimate the flexural state of a flexible link manipulator when applying state-dependent Riccati equation technique in the vibration control. Using Kalman filter minimizes the effect of noises that may corrupt the state measurements. The results from the simulation showed the SDRE control based on Kalman filter was quite accurate in positioning and vibration suppression of the FLM.

Ding and Xiao [78] employed Kalman filter for system states estimation for speed control and resonance suppression of a flexible joint manipulator based on singular perturbation method. The system was divided into a slow system and a fast system for control, where slow system controls the joint motor speed and the fast system suppress the resonance due to the flexibility of the joint. The effectiveness was verified by simulation and experimental results.

Kalman filter shown to be an optimal state observer for systems which cannot be modelled accurately using deterministic model. This method is good when the prediction horizon is short. For long prediction horizon, there will be over prediction during an unknown disturbance input resulting in increased errors at trajectory changes. As stated in section 2.4.2, Kalman filter is useful only when the model and sensor data is linear. For nonlinear process or sensor data, an extended Kalman filter can be used.

#### **vi. Extended Kalman filter approach**

Based on extended Kalman filter, Ahmad and Namerikawa [79] studied localization of a mobile robot bounded by measurement data intermittently unavailable and existence of uncertainties. Their results focussed on minimizing the measurement innovation instead of Kalman gain. The experiment conducted demonstrated that when measurement data was missing at intermittently for a short period of time, the robot was still able to estimate its location, and its errors were statistically bounded. It shows that measurement innovation is very helpful in deciding and pursuing the whole system uncertainties when measurement data are not entirely lost. But it requires that the initial state covariance, process and measurement noises are sufficiently small. Moreover, the linearization error must be reduced to maintain good estimation.

Suarez et al. [25] proposed the application of a stereo vision system for estimating the controlling a Cartesian and joint deflection in an anthropomorphic, compliant and ultra-lightweight dual arm. A colour marker was attached at the end effector of the arm for the stereo vision system to visually track its position. Extended Kalman filter was used to estimate the Cartesian position and velocity of the markers.

Like Kalman filter, extended Kalman filter shown to be an optimal state observer for systems which cannot be modelled accurately using deterministic model and with the present of nonlinear process and sensor data. But, this method is good when the prediction horizon is short. For long prediction horizon, there will be over prediction during an unknown disturbance input resulting in increased errors at trajectory changes. Full state measurements might be difficult to be implemented in the practical engineering environments.

#### **vii. Neural Network approach**

Tang and Wan [80] developed a robust adaptive dynamic surface control method using neural networks for mobile manipulators. They reported the reduced amount of calculation of the algorithm, and that the simulation results showed that the algorithm can be applied in the mobile manipulator system with nonholonomic constraints, uncertainties and disturbances. Naijian et al [81] designed a robust neural network control system is designed for the mobile manipulator, and reported to has achieved position tracking control successfully. They have reported that this robust neural network controller can ensure high accurate position tracking error and incomplete constraint force in different conditions

However, these works were based on rigid manipulator, and were only based on simulation. The prediction capability of the manipulator trajectory has not been studied. It is difficult to train sufficient amount of data when applied for the prediction of flexible manipulator.

Sun et al. [68] employed adaptive neural networks to suppress vibration of a flexible robotic manipulator. The system was modelled with lumped spring-mass approach to improve the accuracy in describing the elastic deflection of the flexible manipulator. The feasibility of the proposed neural network controllers was tested on the Quanser

platform. Potentiometer sensor was used to measure the servo angle and strain gage measure the tip deflection of the flexible beam. Compared to the PD (proportional-derivative) control, it was observed that neural network is quicker in converging tip tracking errors to around zero, thus achieved fast setting time for the trajectory. The control performance was not influenced by the high number of node in the lumped spring-mass flexible manipulator system.

Neural network approach requires relatively less information about the dynamic of the system. This approach has also been effectively addressed control problems in nonlinear systems with unknown dynamics. The problem is that if insufficient learning is provided, neural network could run into errors in the position estimation. Thus, large amount of learning data is required in order to ensure accuracy of this approach. They did not consider input nonlinearities in the control design.

#### **viii. Fuzzy logic approach**

Li et al. [33] proposed an adaptive fuzzy output feedback approach to control a single-link robotic manipulator coupled to a brushed direct current motor with a flexible joint. The fuzzy logic system was designed as an adaptive fuzzy filter observer to estimate the immeasurable states of the unknown nonlinear dynamic associated with the mechanical and electrical subsystems with the use of only the measurement of the link position. The adaptive fuzzy output feedback control approach was developed by combining the adaptive backstepping and dynamic surface control techniques. Only the measurement of the link position is required for feedback and the fuzzy can estimate the velocity signal, thus does not require velocity measurement.

The advantage of such approach is that it does not require all the states of the system be measured directly. It can solve the control problem of robotic manipulators with unknown nonlinear uncertainties. However, this fuzzy logic approaches still not robust enough. It is difficult to develop the rules that can cover all unknown dynamics, the rules have to be carefully tuned to achieve maximum performance.

## 2.6 Benchmarking Conclusion

This chapter reviewed the sensors, sensor fusion techniques, and followed by literature review of past works for prediction of robotic manipulators.

For indoor robot localization, the commonly employed sensors are odometer (usually encoders) and IMU such as accelerometer. Webcams (and cameras) can be implemented using low cost landmark recognition method, where artificial landmark can be placed on the ceiling, such as QR codes. This way, not only that it can reduce the number of landmark need to be memorized, but also easier for the camera to capture the image.

Range sensor, position sensitive device and strain gauges can be used for sensing tip deformation and vibration of flexible beam. IMU and high speed camera are other commonly used sensors for vibration estimation of flexible link manipulators. Range sensors were often used only for validation purposes for vibration estimation of FLMs. Strain gauges provide cheap and easy implementation for flexible link deflection sensing, but prone to noises and interferences, and do not provide accurate measurement for long and slender beams. Accelerometers are often fused with strain gauges and/or cameras to overcome the deficiency to enhance signal accuracy and reliability. The problem of vision based data is the delay in signal output.

Because of the inaccuracies of low cost sensors, it is difficult to rely on a single sensor type to provide accurate displacement measurement. Often, fusions of different sensor types are implemented to improve the robustness and accuracy. The fusion algorithm should work in such that the sensors complement each other to achieve desirable results. Weighted averaging, Kalman filter, complementary filter, particle filter, fuzzy inference system and neural network are some fusion algorithms commonly used for sensor fusion. Among these, Kalman filter is the most appropriate sensor fusion algorithm for use in this research due to its readily available prediction stage and its recursive nature which made it suitable for vibration prediction. However, Kalman filter relies on the trustworthy data from the sensor, a situation which is not always possible and especially for low cost robotics.

Prediction methods are often used to overcome the delayed signal from camera, where past and present outputs are extrapolated to predict future outputs. The reviewed

algorithms used in the prediction and control of robotic manipulators and mobile robots includes meta-heuristic optimization, finite-time tracking, MPC, new displacement approach, Kalman filter, extended Kalman filter, Neural network and fuzzy logic.

Meta-heuristic optimization provides a good method for planning the path of the mobile manipulator. But, the methods require good understanding of the system motions and offline pre-planning to generate desired paths.

The finite-time controller converges tracking errors to zero at finite time, which requires knowledge of the system and the feedback of the current state of the system. Accurate knowledge of the platform velocity and motion trajectory of the system is required. The prediction errors will increase when applying input, and any unknown disturbance input would cause high tracking errors and would not converge to zero at finite time.

Model predictive control (MPC) provides a good prediction for future control variables. In this method, the controller can constantly plan for the motion trajectory based on the model created for the system. The accuracy of the model is important for stability and accurate control of the robotic manipulator. However, the accurate models of real robotic system are not available or extremely hard to formulate for complex systems. Any singularity not being mathematically model will result in high error.

The new displacement approach that make used of nonlinear finite element model and provides an open loop control for vibration modes of the flexible manipulator. Any unknown disturbance is still cannot be modelled, and without a feedback the control becomes unstable when encountering disturbance in the motion input.

The Kalman filter approach is good for prediction approach for the flexible manipulator, when delay compensator is used to extrapolate past and present output to predict the future output. However, it is restricted to trajectory that linear or short horizon prediction. Consequently, it resulted in high error due to overestimation to the change in direction of the trajectory or disturbance. Extended Kalman filter which linearize the nonlinearity of the flexible manipulator can be used to improve the error. However, for a highly nonlinear system, linearization may introduce errors leading to divergence. The non-geometric error sources for the flexible manipulator are either difficult or impossible to model correctly and completely.

When Extended Kalman filter is applied with sinusoidal regression algorithm, the prediction can be extended to longer horizon. Furthermore, this method can cope with trajectory with changing direction and unknown disturbance. The cross-correlation that computes the time-delay between the accelerometer and camera made the signal more robust with minimal errors.

Neural networks shows ability to predict and suppress vibration of a flexible robotic manipulator, and quick in converging tip tracking errors to around zero. However, as neural networks are data driven, their performance depends on the quantity and quality of the trained data. Provided that these data can sufficiently represent the flexible manipulator's trajectory, otherwise it will run into extreme error in the prediction process. Huge amount of training data will be required to accurately predict the vibration of the flexible manipulator, which will lead to over-fitting problem.

Fuzzy logic approach is based on a set of rules to mimic the trajectory of the flexible manipulator. Due to the dynamic singularity of the flexible manipulator, it is impossible for to develop the set of reasoning that can accurately predict the behaviour of the tip vibration of the manipulator. This approach is thus not robust enough to cover the unknown dynamic and predict the trajectory of the flexible manipulator.

In conclusion, MPC and KF/EKF methods are most appropriate benchmark for the vibration prediction of the flexible manipulator. The prediction horizon can be extended with the implementation of the sinusoidal regression.

### 3 Sensor Characterizations and Modelling of Mobile Flexible Link Manipulator

There are two parts in this chapter. First part is the characterization of the accelerometer, camera and encoder, which covers sections 3.1 to 3.3. Second part is the modelling of the mobile flexible link manipulator, which covers section 3.4 to 3.7.

#### 3.1 Characterizations of Accelerometer

Displacement information of a moving vehicle or a vibrating beam can be obtained through double integration on the acceleration signal from an accelerometer. To double integrate an acceleration signal to get the displacement, the initial position and initial velocity must be known. These initial conditions are usually zero or from rest position.

##### 3.1.1 Mathematical equations for computing accelerometer signal

The output from accelerometer is actually gravitational output, in terms of voltage. To convert to the acceleration output, the following equation can be used [34]:

$$a_i = (V_i - k_{vi})/g_{si} \times 9.81 \quad (3.1)$$

where

$a$  = acceleration,

$V$  = output voltage from the sensor,

$k_v$  = offset voltage, and

$g_s$  = sensor gain

$i$  = represented as  $x$ ,  $y$  or  $z$  axis of the accelerometer.

The first integration of the acceleration  $a$  obtains the velocity  $v$  as in the following equation [34]:

$$v(t) = v(t_0) + \int_{t_0}^t a(t) dt \quad (3.2)$$

where

$t_0$  = initial time,

$v(t_0)$  = initial velocity.

To get the displacement output from velocity, a similar formula as above is used [34]:

$$x(t) = x(t_0) + \int_{t_0}^t v(t) dt \quad (3.3)$$

where

$x(t)$  = the initial position.

The next stage is the integration of the acceleration signal into displacement data. There are a number of numerical integration algorithms. The rectangular integration method is the simplest way for performing a numerical integration. An accumulator is used to sum all past sampled inputs and the current input sample, divided by the sampling rate. Rectangular integration is represented by the following difference equation [82]:

$$y_n = \frac{1}{f_s} \sum_{k=0}^n x_{n-k} = y_{n-1} + \frac{1}{f_s} x_n \quad (3.4)$$

where

$x$  = past inputs,

$f_s$  = the sampling frequency,

$y$  = the integrator's output,

$n$  = the number of data points to integrate.

Another numerical integration method is based on trapezoidal rule. This method is more commonly used as it is more accurate. The trapezoidal integration can be represented as [82]:

$$y_n = y_{n-1} + \frac{1}{2f_s} (x_{n-1} + x_n), n > 0 \quad (3.5)$$

Simpson's rule is another method of integration, which can be computed by the following difference equation [82]:

$$y_n = y_{n-1} + \frac{1}{f_s} \cdot \frac{x_{n-1} + 4x_n + x_{n+1}}{6} \quad (3.6)$$

Unlike the other two methods, this method requires a future input data,  $x_{n+1}$  in order to get the output,  $y$ . This requires the system to be able to predict the future input. We decided to use the trapezoidal method of integration to perform the numerical analysis.

Simpson's rule resulted in high integration errors; possibly because of a large difference in successive future sample inputs.

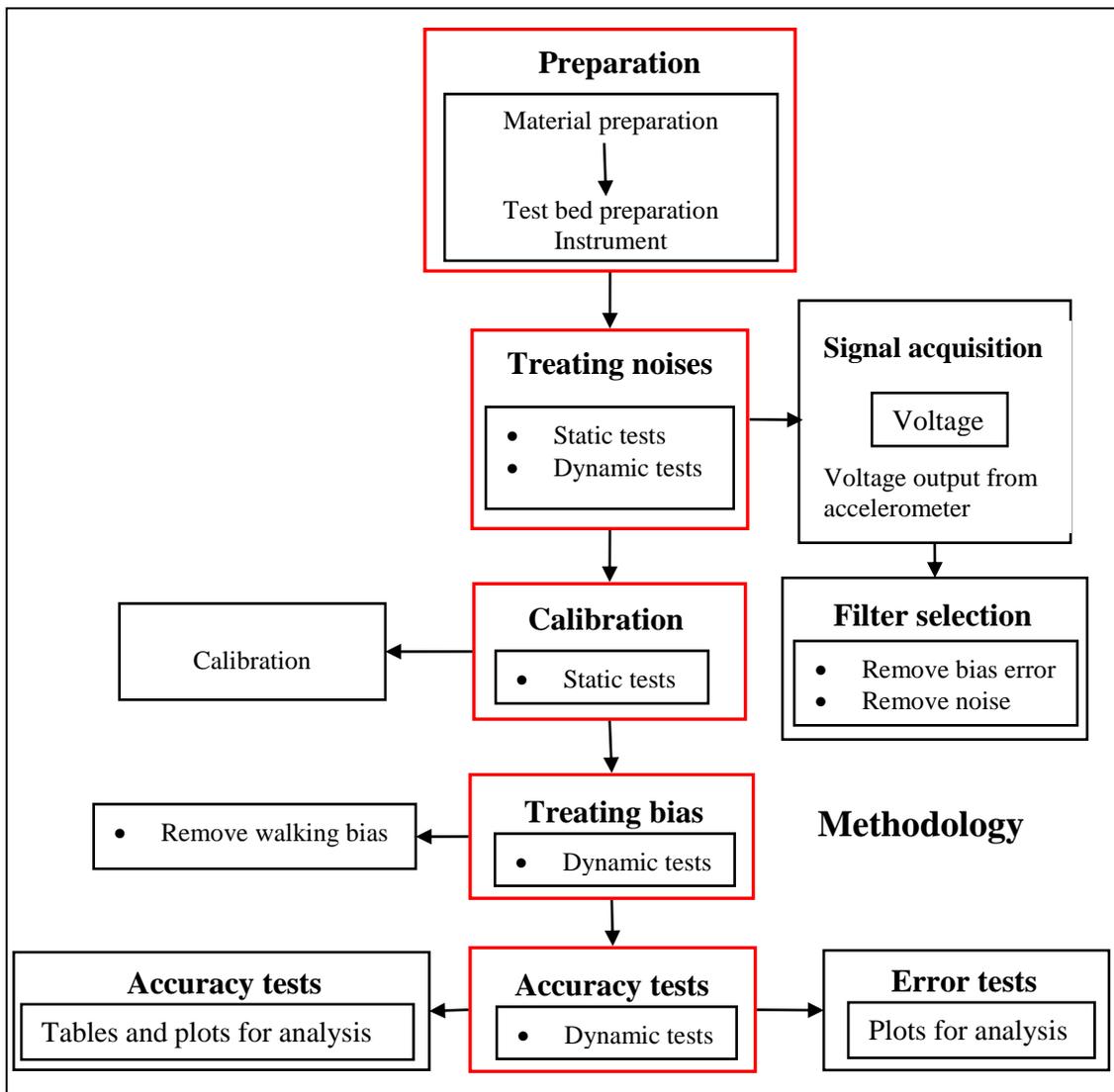
### **3.1.2 Methodology for testing and conditioning the accelerometer**

The plan is to setup the accelerometer test bed to carry out static and dynamic tests on the accelerometer while acquiring the signals. The objective is to certify that accelerometer can be used for measuring the motion of the mobile platform and the vibration at the tip of the flexible manipulator. The raw signals are first treated with signal processing algorithm to attenuate the errors and noises. Validation tests are then carried out to test the results. The phases of the methodology for the tests are as follows:

- I. Preparation: In the preparation step, setup the selected accelerometer with appropriate voltage input and data acquisition instrument. Then, prepare the test bed for the accelerometer to be tested on. Refer to section 3.1.3 for detail descriptions of the setup.
- II. Static and dynamic tests for filter selection: At this phase, perform attenuation of errors and noises for the accelerometer as follows:
  - a) Static test- compare smoothing filters: the raw accelerometer's signal is filled with noises, thus has to condition the raw signal to remove the noises.
  - b) Dynamic test- compare smoothing filters: select the optimal smoothing filter. Refer to section 3.1.4 for the detail descriptions of the test procedures.
- III. Static test— calibration: The purpose of this phase is to find the constant offset bias and acceleration gain errors, and then to determine the offset and gain values for the signal. Refer to section 3.1.5 for the detail descriptions of calibrating the accelerometer.
- IV. Dynamic test— treating bias error: The purpose of this phase is to remove the effect of walking bias. Test for dynamic errors and to reduce the errors. This is dealing with walking bias errors. Refer to section 3.1.6 for detail description of testing and dealing with walking bias.
- V. Accuracy tests— At this phase, convert the signal into acceleration, velocity and displacement outputs, and then measure accuracy and its errors:

- a. Accuracy tests: Test the accuracy of the acquired signals with a set of motions. Verify the accuracy of the accelerometer through dynamic tests. The purpose is the find the range of speeds that the accelerometer is capable to detect and how accurate can it measure.
- b. Dynamic errors: Test for the cross-axis affects that contribute to the measurement errors. This error cannot be treated or attenuated standard approach.

Refer to section 3.1.7 and 3.1.8 for the detail descriptions of tests.

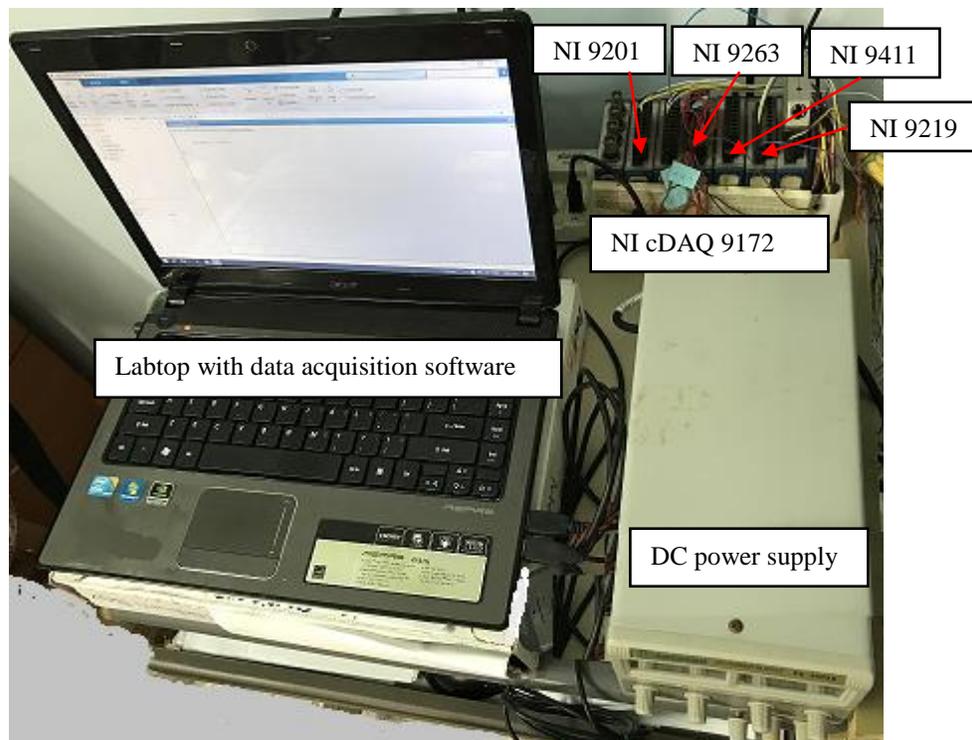


**Fig. 3.1** Flow chart of methodology for testing the accelerometer

The flow chart of the methodology for testing the accelerometer is illustrated in Fig. 3.1 below. Mathematical formula was developed for converting the acquired data into acceptable acceleration output. The test results are presented in section 4.1 in chapter 4.

### 3.1.3 Preparation of the accelerometer

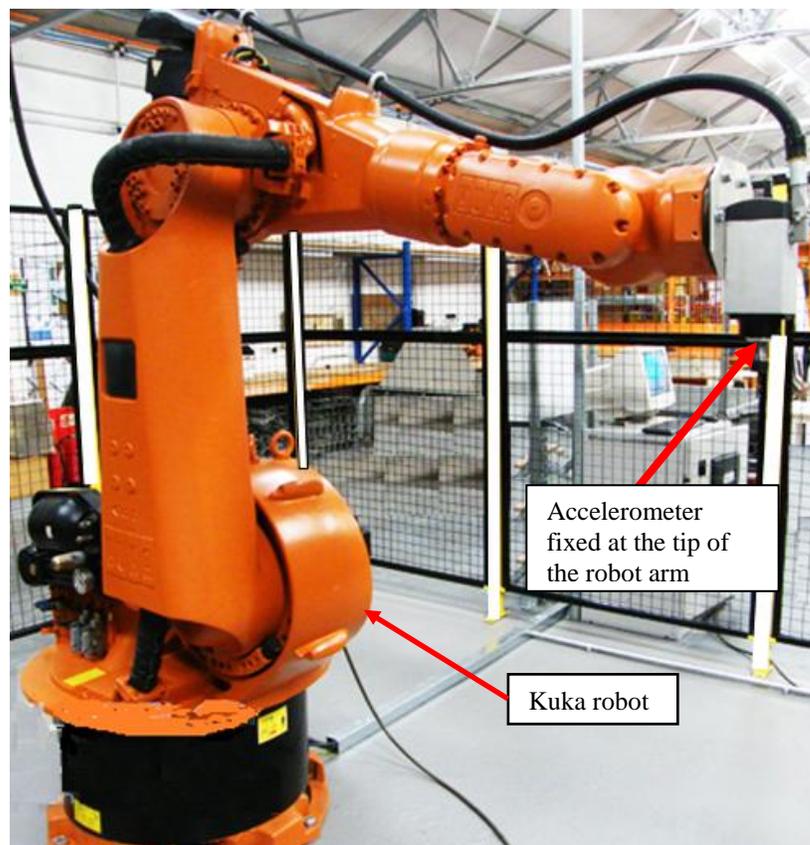
An off-the-shelf accelerometer to be evaluated is MMA7260Q, a low cost capacitive micromachined triaxial accelerometer, available at less than US\$20. It has selectable sensitivities of  $\pm 1.5g$ ,  $\pm 2g$  and  $\pm 6g$ . The bandwidth response is 350 Hz for X- and Y- axes, and 150 Hz for Z- axis. According to the data sheet in Appendix B.1, the operating voltages are between 2.2 V to 3.6 V. For maximum accuracy, the accelerometer was powered by the DC voltage of 3.6 V via NI 9263 analog output module. The acceleration signal was acquired via National Instruments NI 9201 acquisition module on NI cDAQ 9172 chassis. According to the data sheet of Appendix B.1, the recommended sampling frequency is 11 kHz, and thus 11 kHz data acquisition rate was set. The physical setup of the data acquisition hardware is depicted in Fig 3.2.



**Fig. 3.2** Hardware setup for the data acquisition modules  
(For acquiring signal from accelerometer)

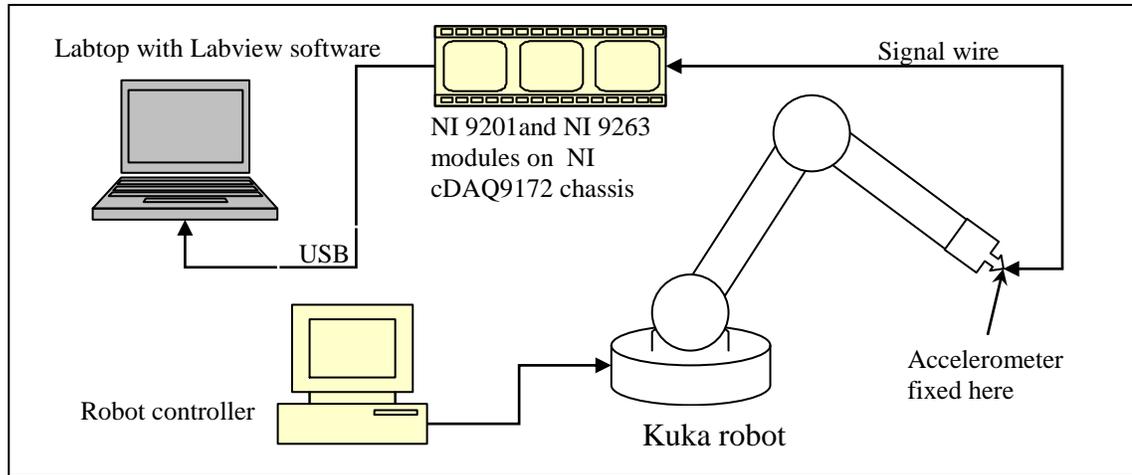
To test the accelerometer, the test bed needs to have accurate and stable movement, and has good repeatability. The KUKA robot available at the Mechanical lab of University of Nottingham, Malaysia campus had been select for preparing the test bed. This KUKA industrial robot, as shown in Fig. 3.3, is fenced in order to isolate it from the human user for safety purpose. The robot controller is situated outside the fence for robot programming and control. The robot was initially setup for robot welding demonstration class for the Engineering students in the Mechanical Engineering Faculty.

This KUKA robot is a highly accurate robot with 6-DOF movements; and it has a repeatability of less than  $\pm 0.05$  mm and a maximum reach of 2.033m. The accelerometer is to be used for sensing the tip movement of the robotic manipulator; thus the KUKA robot is an appropriate platform. To carry out the tests, the accelerometer was fixed at the tip of the robot arm.



**Fig. 3.3** Kuka robot and accelerometer fixed to the tip  
(This is the robot used in University of Nottingham, Malaysia campus)

Fig. 3.4 illustrates the schematics for the whole test setup for testing the accelerometer. The robot controller controls the arm movement at the end-effector. The signals to and from accelerometer are provided through National Instrument's NI modules. The Labview software controls the signal acquisition and voltage output to the NI cDAQ 9172 chassis.



**Fig. 3.4** Schematics of the test setup for the accelerometer

### 3.1.4 Filter selection

Static and dynamic tests were carried out to choose suitable filter to attenuate noises. The specific objectives of these tests are to determine the bias errors and noises through static and dynamic tests. The purpose is to determine the noises in the signal, and then to find the filter to attenuate the noise.

In the static tests, a single axis (z-axis) was placed facing gravity, while the other two axes (x- and y-axis) facing horizontally. The acquired signals were then tested with four types of filters. The selection criteria are effectiveness of smoothing the noise, processing speed and preservation of the signal.

The dynamic tests are to characterize the dynamic performances and aid selection of the most appropriate filter. The dynamic tests are carried out setting the accelerometer to vibrate by placing it at the tip of a cantilever beam, and acquire the signals.

The results for the above tests are depicted in section 4.1.1 for static tests and 4.1.2 for dynamic tests. When the signal content is contained in a certain bandwidth, the signal must be sampled at a frequency to prevent aliasing problem. Therefore, for maximum acquisition without aliasing and best signal reconstruction, it is recommended for 11 KHz sampling rate (refer to Appendix B.1) for signal acquisition.

### 3.1.5 Calibrating the accelerometer

Static calibration is used to compensate for systematic errors, such as offset bias and drift. Thus, the purpose of calibrating the accelerometer is to find the constant offset bias and acceleration gain error. We first determine the maximal and minimal for 1g and -1g. This is achieved by aligning the measuring axis (z-axis) with direction of gravity by placing the measuring axis facing statically downwards and then opposite direction upwards. The results for the maximum and minimum acceleration values are shown in Table 4.2 in section 4.1.3 of chapter 4.

The zero acceleration voltage value is assumed to be the centre of the maximum and minimum output ranges. This zero acceleration voltage value is the constant offset bias, which can be calculated as [83]:

$$\text{Constant offset bias, offset} = \frac{\text{max} + \text{min}}{2} \quad (3.7)$$

The acceleration gain can then be obtained by deducting constant offset by maximum value as follows [83]:

$$\text{Acceleration gain, gain} = \text{offset} - \text{max} \quad (3.8)$$

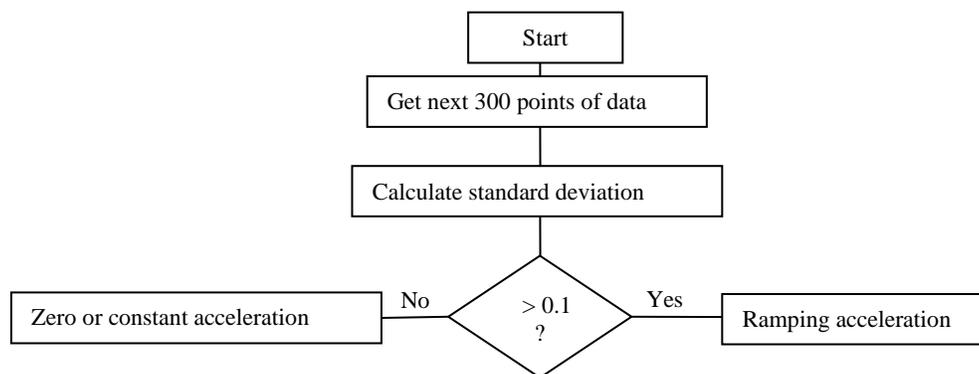
Using equation (3.1), we can then verify the actual zero and gravitational accelerations by placing the measuring axis parallel to the gravity. The results are shown in Table 4.3 in section 4.1.3 of chapter 4.

### 3.1.6 Effect of walking bias errors

Accelerometers possess unwanted drifts called walking bias associated with the acceleration signal caused by a small DC bias [45]. Ideally, the accelerometer should not contain any DC bias. The presence of drift due to walking bias can lead to large integration errors. If the acceleration signal from an accelerometer is integrated without

processing, the problem of walking bias can cause unbounded propagated error when acceleration measurements from the accelerometer are integrated into velocity and displacement.

Tests were carried out to test for the walking bias. The results for the walking bias are illustrated in Fig. 4.6 and 4.7 in section 4.1.4 of chapter 4. Standard deviation is used to measure the deviation of the data. From an arbitrary selection of 300 data points, it was found that a standard deviation of less than  $0.1 \text{ m/s}^2$  can be considered as constant acceleration. Thus, an algorithm was developed so that for every 300 points that if the standard deviation is less than 0.1 then it is considered to be a zero or a constant acceleration, else it is a changing acceleration. Fig. 3.5 illustrates the flow diagram for this algorithm to compute the standard deviation for every 300 points, and differentiates between a zero, a constant acceleration or a ramping acceleration.



**Fig. 3.5** Algorithm to compute standard deviation of 300 points to determine acceleration type

Fig. 4.8 and 4.9 in section 4.1.4 (chapter 4) illustrated the outcomes after the algorithm is implemented.

### 3.1.7 Dynamic verifying the accuracy of the accelerometer

In order to test the dynamic working range and error mode of the accelerometer, we put the accelerometer into dynamic tests. The tests were carried out on the Kuka robot by traversing the accelerometer with various travelling speeds to determine its working ranges. Only one axis (y-axis) of the accelerometer was tested.

The purpose of the tests is to determine the dynamic errors when the accelerometer moves at various speeds, and to determine the speeds that the accelerometer can perform well for motion estimation. We define the test factors and test responses as follows:

**a. Define the factors for moving the accelerometer:**

In this test, there are three factors:

- I. The speed of the travel: specifying the speed of moving the accelerometer.  
To maximize the information with minimum number of test points, a prescribed numbers of robot speeds were selected. It was found that the speed lower than 5 cm/s is too low for the accelerometer to detect the motion, while the maximum speed that can be set for the robot is 2m/s. Thus, the speed of travel were tested at 5 cm/s, 15 cm/s, 25 cm/s, 50 cm/s, 1 m/s and 2 m/s robot speeds for straight traversal.
- II. The distance to travel: specifying the distance to move accelerometer.  
The robot arm is constraint to stretch at a maximum distance of 0.8m horizontally and 0.6m vertically. We only test 0.8 m for horizontal traversal.
- III. The type of motion: specifying the travelling motion of which to move the accelerometer.  
In this research, we only performed the experiment on straight horizontal path. Only Y-axis is facing the direction of travel, with the Z-axis facing towards gravity and X-axis facing lateral direction.

**b. Define the responses from the accelerometer:**

In this test, the responses are:

- I. The acceleration: the acceleration output converted from the signals of the accelerometer.
- II. The velocity: the velocity output after first integration of the acceleration output.
- III. The displacement: the displacement output after second integration of the acceleration output.

**c. Repeatability tests**

25 sets of experimental samples were taken for each speed set-points of the robot tip extending, retracting, or turning. The tests results are exhibited in section 4.1.5 and 4.1.6 in chapter 4.

### 3.1.8 Cross-axis affects

Rotating the accelerometer 180° to determine the cross-axis errors. It is also to examine how much can the cross-axis error contribute to the measurement error. According to the datasheet for MMA7260Q (referring to Table 2 in Appendix B.1), this accelerometer has a cross-axis sensitivity of 0 to 5% (max). To test for this cross-axis affect two tests were carried out:

- I. Rotating the accelerometer 180°. This is to test for the cross-axis errors.
- II. Traverse test on the accelerometer. This is to examine how much can the cross-axis error contribute to the measurement error. ~~Only one axis~~

#### I. Rotating the accelerometer 180° to test cross-axis errors.

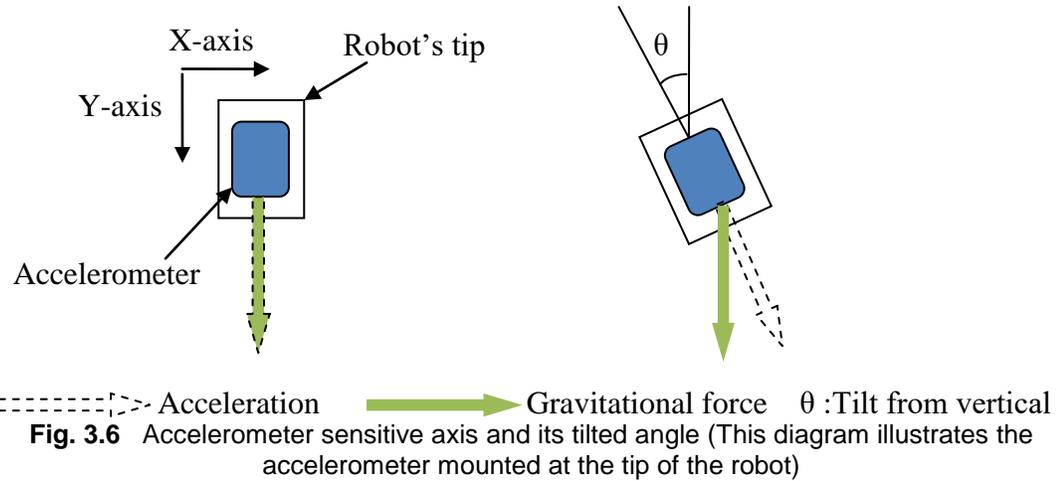
With the accelerometer mounted at the tip of the Kuka robot, the tip was programmed to rotate 180°, with the initial orientations: X-axis facing horizontal, Y-axis facing towards gravity, and Z-axis as centre of rotation. The final orientation being Y-axis facing upwards; as illustrated in Fig. 3.6. The true acceleration values were calculated based on the following equation [36]:

$$F = g_a * \cos(\theta) \tag{3.9}$$

where

- $F$  = the true value of the acceleration vector at various tilt angles,
- $g_a$  = the gravity vector, and
- $\theta$  = the angle of tilted.

The test result is presented in part I of section 4.1.7 in chapter 4 (result chapter).



## II. Traverse the accelerometer at the speed to 2 m/s.

We now analyse the cross-axis error when we traverse the accelerometer at maximum speed of 2 m/s. Considering the 5 % cross-axis sensitivity (refer to Table 2 in Appendix B.1), running the robot at 2 m/s speed would contribute to maximum error of 0.1 m/s<sup>2</sup>. The results illustrating the effect of this affect when the accelerometer traverses in a straight line are shown in Fig. 4 19 and 4.20 at part II of section 4.1.7 in chapter 4.

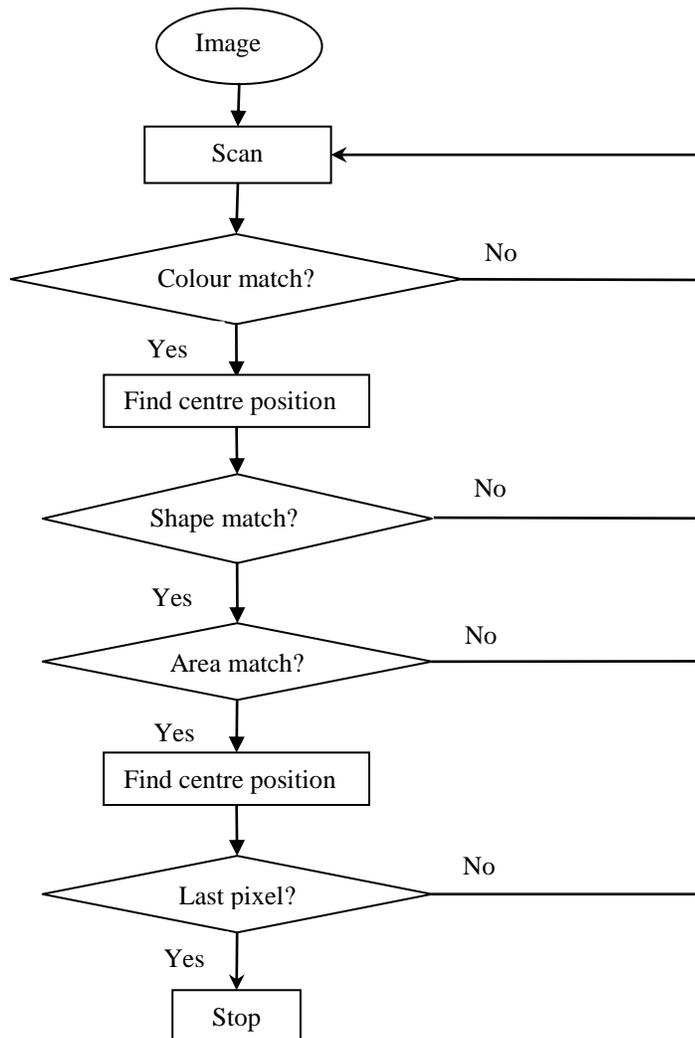
## 3.2 Characterization of camera

The localization of the mobile platform can be measured by using camera or webcam, while webcam is cheaper method. As for vibration tracking of the flexible beam, only camera can be used. Webcam will result in poor image as the vibrating beam moves in very speed, but webcam can be used for tracking mobile robot. Therefore, this section discusses the tests for localization of mobile platform using webcam, while vibration of flexible beam is measured by using high speed camera.

### 3.2.1 Algorithm for computing camera data

In the domestic environment, such as the living room, the feature that usually remains unchanged is the ceiling. As discussed in section 2.1.1 landmark recognition method for indoor mobile robots localization can be by fixing the artificial landmark on the ceiling [20, 21], while a camera is fixed on the robot to read the landmark. In this work, we

place a marker on the ceiling, where the webcam mounted on the mobile platform facing vertically upwards to capture the marker at the ceiling. Webcam is a cheaper method as compared to camera.



**Fig. 3.7** Flow chart for recognising marker position

Fig. 3.7 above shows the flow diagram for the marker recognising process. To detect the marker, the coordinates of the marker on the image plane are evaluated via an operation of blob detection. Next, the colour, shape and area matching method is used to recognise the marker. The program first filter off the background by matching the colour value of marker, anything not matching the colour threshold value will turn black while the rest

turns into white. A morphological structuring element method then finds the marker by fitting its shape with the marker. Next, the objects that match the area size of the marker will be identified. Finally, the centre point of the identified object is determined.

With the data obtained we compute the gain by dividing the actual displacement by the mean of the pixel counts between two adjacent marker's images [31]:

$$g_{disp} = \frac{d_{cam}}{\text{mean}\{(x_2-x_1)+\dots+(x_n-x_{n-1})\}} \quad (3.10)$$

where

$g_{disp}$  = gain for computing the marker's displacement using pixel count between adjacent image,

$d_{cam}$  = the displacement from previous camera position, and

$x$  = the pixel position of the marker.

The displacement calculation  $d_{est}$  using webcam/camera, is then calculated using pixel count  $p_{cnt}$  multiply by gain  $g_{cam}$  [31]:

$$d_{est} = g_{cam} \times p_{cnt} \quad (3.11)$$

where

$d_{est}$  = displacement in mm

$g_{cam}$  = pixel gain

$p_{cnt}$  = pixel count

### 3.2.2 Methodology for testing camera

In this research, webcam is used as an absolute method for tracking the position of the mobile platform, while the camera is used to sense the displacement at the tip of the flexible manipulator. The phases of the methodology for the tests are as follows:

- I. Preparation: In the preparation phase, the selected camera was setup, and the target object set to the position required to be captured. Then, the camera was adjusted to focus to the target object. Two setups were prepared; one is for webcam mounted on mobile platform, and the other one is for CMOS camera mounted at fixed end of a

flexible beam. Refer to section 3.2.3 for the details of preparation and setup of the webcam and CMOS camera.

II. Static test- calibration: The purpose of this phase is to apply mathematical algorithm with appropriate settings for converting image pixel into displacement information. Refer to section 3.2.4 for the details of calibration of the camera.

III. Test for mobile robot localization with webcam: The purpose of this test is to check the accuracy of the webcam in locating the mobile robot position at different locations. The tests are conducted as follows:

a. Static test— Move the mobile robot to different location. Capture the image of the marker at the ceiling. Convert the marker's position in the image into robot's position.

b. Dynamic localization test— Set to move the robot at various speeds. Capture the marker at the ceiling while the robot moves. Convert the marker's position in the image into robot's position.

Refer to section 3.2.5 for the details on the tests for robot localization using webcam.

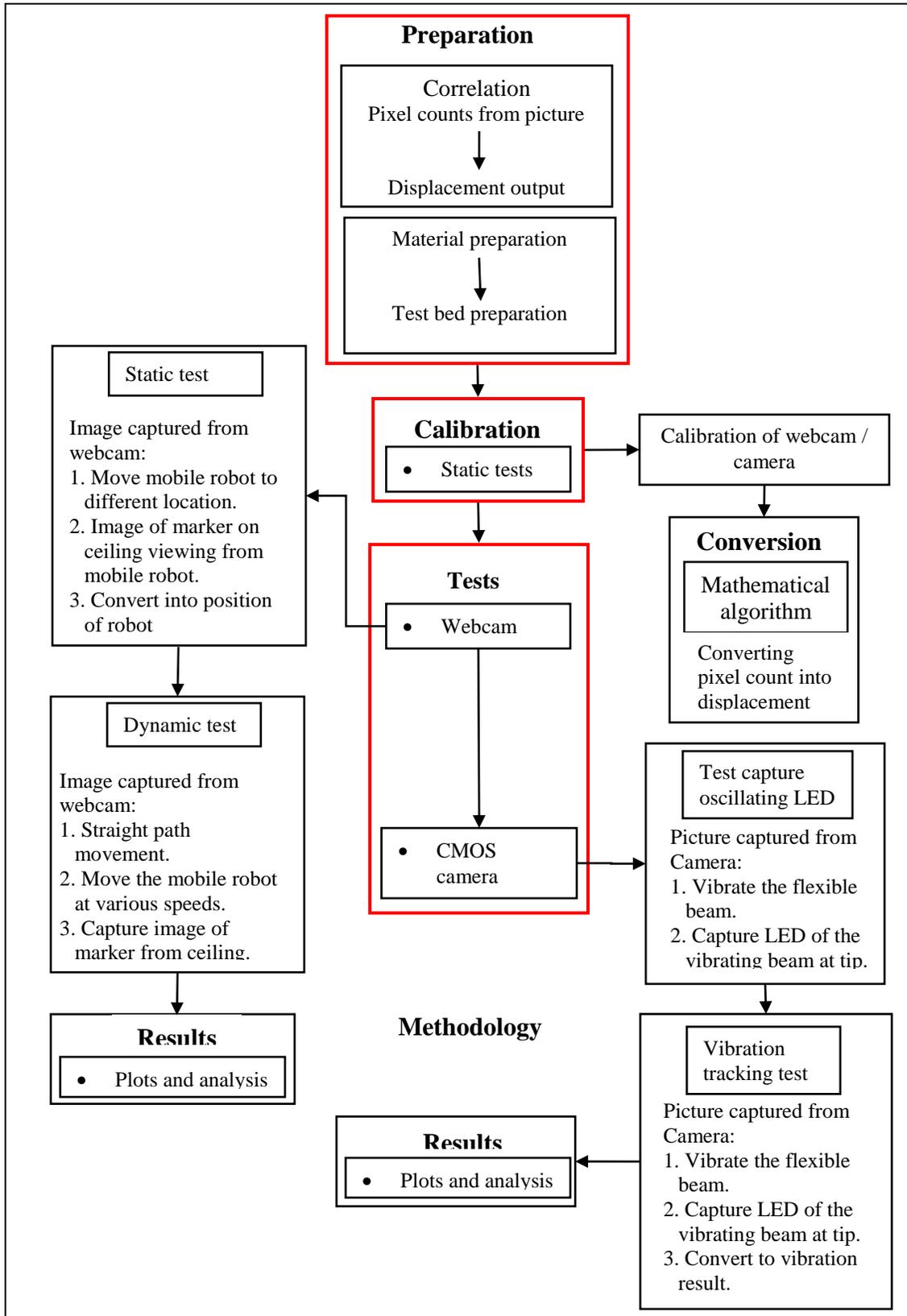
IV. Test for vibration tracking of flexible beam using camera: The purpose the tests the ability of the CMOS camera in capturing and measure the vibration at the tip of the flexible manipulator. The tests are carried out as follows:

a. Vibrate the flexible beam. Capture the oscillating LED using the CMOS camera.

b. Vibrate the flexible beam. Convert LED image into vibration result for the tip of the flexible beam.

Refer to section 3.2.6 for the details on the tests for vibration tracking using camera.

The flow chart of the methodology for testing the cameras is illustrated in Fig. 3.8 above. The test results are presented in section 4.2 in chapter 4.



**Fig. 3.8** Flow chart of methodology for testing the webcam / camera

### **3.2.3 Preparation and setup for testing web cam and CMOS camera**

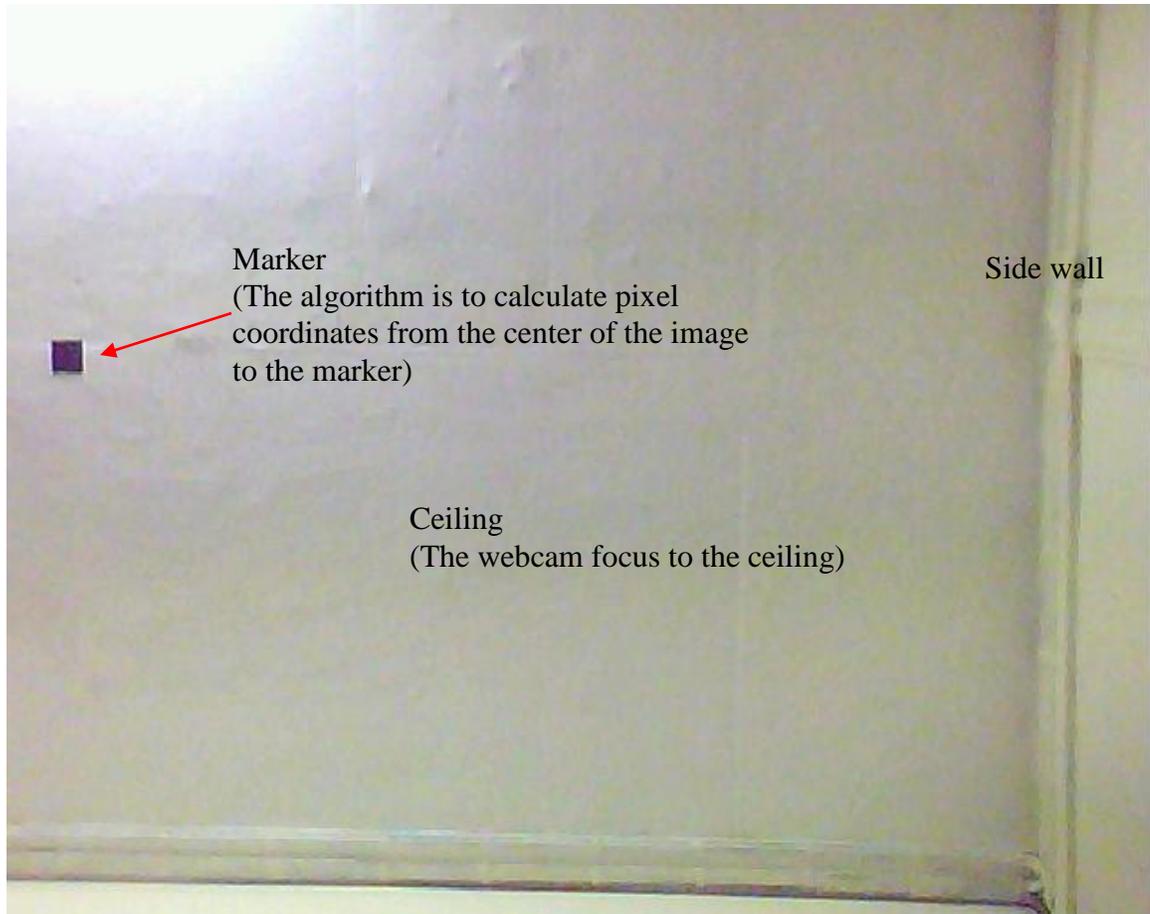
In this research, an Ultra-HD PC Video webcam and a Firefly series FMVU-03MTC-CS USB CMOS camera are proposed. This section evaluates the possibility of using proposed webcam for vehicle position estimation and the proposed CMOS camera for vibration estimation of the flexible beam. The webcam is available at less than US\$20 and 5g in weight. It is a 12 megapixel camera with a frame rate of 30 fps (frames per second). The CMOS camera cost at US\$295, weighing 37g. It has a resolution of 0.3 megapixels with a frame rate of 60 fps.

#### **a. Setup for mobile platform localization**

The Ultra-HD PC Video webcam is used for tracking the localization of the mobile platform. In order to capture relatively good image for moving object, the webcam is configured to the resolution of 640×480 for frame rate of 10 fps. The height from the lens of webcam to the ceiling is 2.4 m, and 11.7 cm above the ground. With this viewing distance, it has a wider viewing range, thus lesser markers are required on the ceiling.

In order to acquire the image that can be processed, the marker needs to be of reasonable dimension, and the shape of the marker need to be distinguishable. We use a black 48mm by 48mm square marker. As the marker is mounted to the ceiling its surface need to be anti-reflective so that it will not reflect light. The marker is placed at a position where the lighting condition is adequately lit.

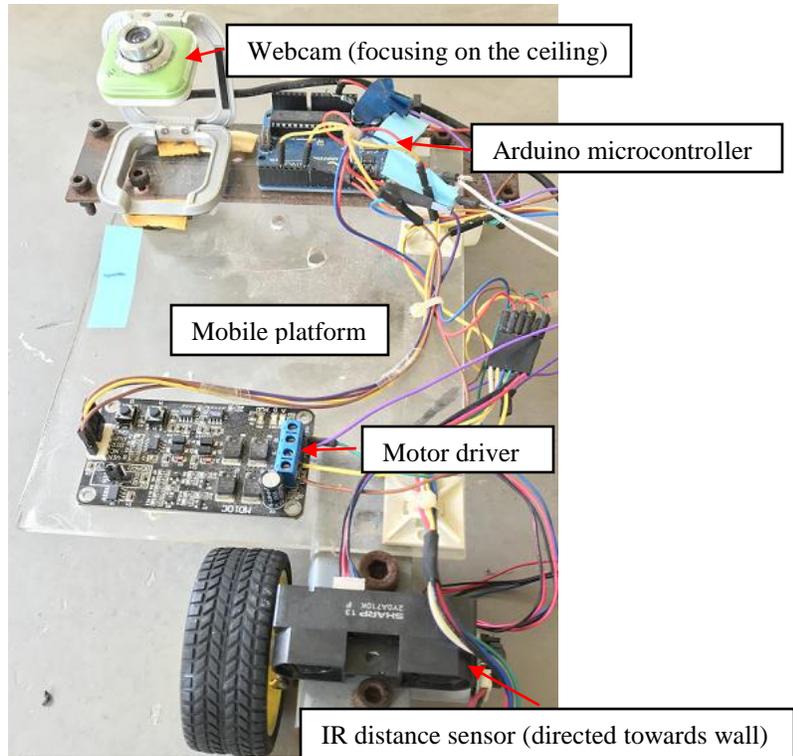
The image of the marker is shown in Fig. 3.9 below. It can be seen that one problem is the uneven illumination. The left top edge in the picture is very bright due to lighting, while the bottom right appeared darker. Thus, the choice of marker's colour and shape has to be chosen so as to distinguish it with the surrounding.



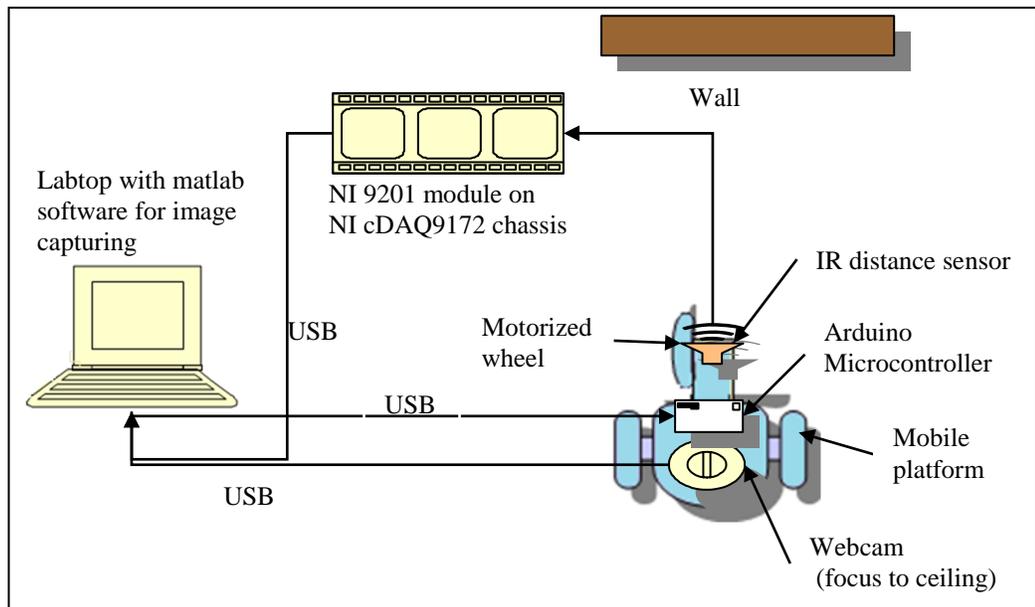
**Fig. 3.9** Image of marker on the ceiling

The webcam on mobile platform is depicted in Fig. 3.10 below. It consists of the mobile platform with motion driven by motorized front wheel controlled by the Arduino microcontroller to the motor amplifier. To measure the actual position of the mobile platform, a long range Sharp (GP2Y0A710F) IR distance sensor is mounted at the front of the mobile platform to track the position the mobile platform with respect to the wall. The signal of the distance sensor is acquired via the National Instrument NI 9201 module. The webcam is connected to the laptop and the image of the ceiling marker capturing using Matlab's image acquisition toolbox.

Fig. 3.11 illustrates the schematics for the test setup for testing the webcam on mobile platform.



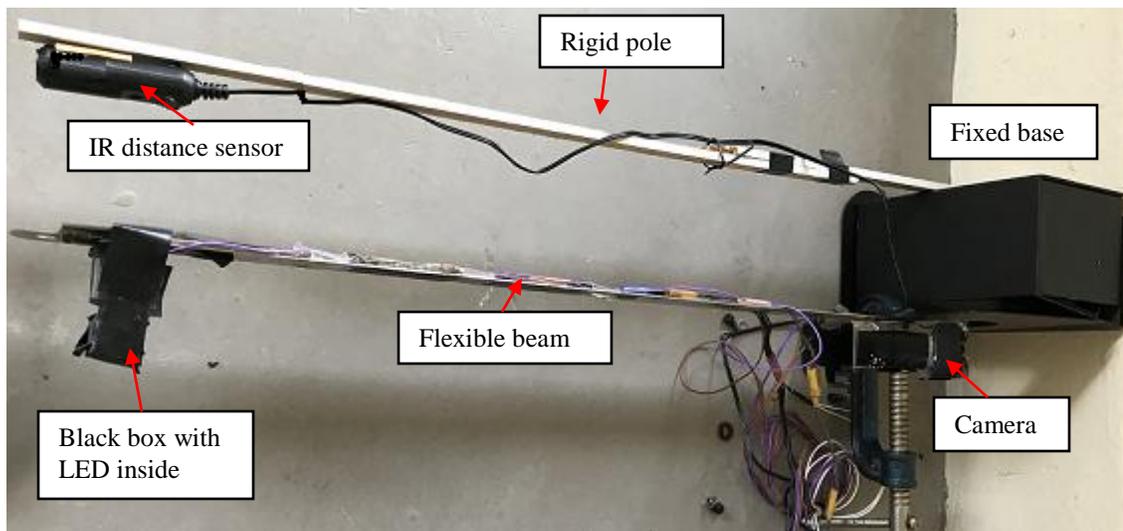
**Fig. 3.10** Setup for webcam and IR distance sensor on mobile platform



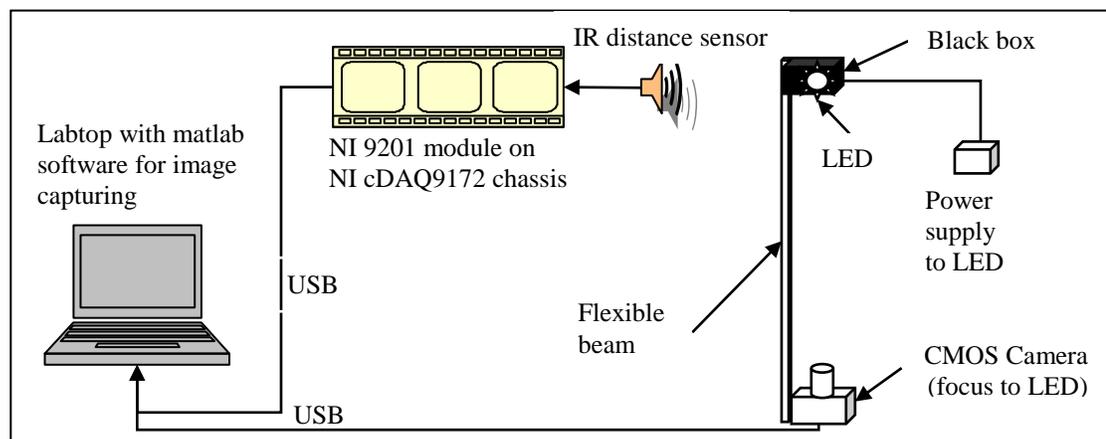
**Fig. 3.11** Schematics of the test setup for webcam to measure mobile platform position

## b. Setup for vibration tracking of flexible beam

The approach used here is by fixing the CMOS camera at the fixed end of the beam and an LED at the free end, so as to capture the image of oscillating LED. The LED is enclosed inside a black light-weight cardboard box with a small hole so that the light can only be seen from a one angle only. The distance from the lens of camera to the LED is 50 cm. To measure the actual vibration of the flexible beam, a short range (GP2Y0A21YK) IR distance sensor is mounted proximity to the tip of the flexible beam. Fig 3.12 depicts the physical set up, while Fig. 3.13 illustrates the schematics for the test setup for testing the CMOS camera on flexible beam.



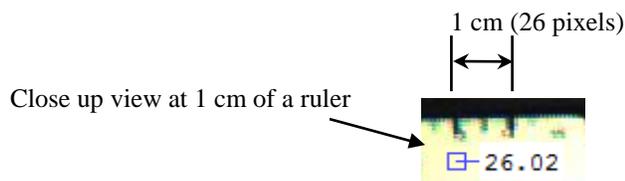
**Fig. 3.12** Physical setup for camera and distance sensor on flexible beam.



**Fig. 3.13** Schematics of the test setup for CMOS camera to measure flexible beam vibration

### 3.2.4 Calibration of camera for measuring vibration of flexible beam

The Firefly series FMVU-03MTC-CS USB CMOS camera is used for measuring the vibration of the flexible beam (refer to appendix B.2 for the datasheet about the camera). To calibrate the camera, a ruler was placed 50 cm from the lens of the camera. The camera is set to  $750 \times 480$  for maximum resolution. The image was then captured. Fig. 3.14 exhibits the number of pixels for 1 cm wide measurement. The measurement shows that 26 pixels equals 1 cm, which means 1 mm equals 2.6 pixels. The camera is thus has around 0.5 mm accuracy at static condition for maximum resolution. The maximum displacement that is within the view of the image is 576 cm.



**Fig. 3.14** Image displaying that there are 26 pixels for 1 cm wide measurement

### 3.2.5 Tests for mobile robot localization with webcam

The objective is to determine the accuracy of the webcam in computing the position of the robot. In considering the factors to be studied, the lighting condition, marker's surface and the high of the marker is assumed to be fixed, and thus the variables are resolution of the measurement and the speed of the robot.

Two modes of tests were carried out to characterize the sensor:

- I. Static tests: To determine the accuracy and to find a method to improve the accuracy. The images were taken at fixed distance intervals apart.
- II. Dynamic tests: To determine the maximum moving speed of the robot that the webcam is capable of capturing. The robot is made to move at various speeds while at the same time the camera records the images as it moves.

#### I. Static tests with webcam for localization of mobile platform

The images were captured at fixed distance interval apart as follows:

1. Starting from an origin, the camera advances 1 mm distance to the next point.

2. Starting from the origin, the camera advances 5 mm distance to the next point.
3. Starting from the origin, the camera advances 5 cm distance to the next point.

20 samples were taken for tests 1 and 2, and 30 samples were taken for test 3. The results are shown in section 4.2.1 in chapter 4.

The results showed that it has high error of up to 28 mm for the displacement measurement. To improve the accuracy, interpolation method is proposed, whereby the marker positions in the image are predetermined at fixed interval of the camera position. This means image teaching is required to store the marker's position into memory. We choose a 50 mm interval teaching. The results are shown in section 4.2.2 in Chapter 4.

## **II. Dynamic tests with webcam for localization of mobile platform**

In the dynamic experiments, the mobile platform is set to move in a straight path with the camera capturing the marker on the ceiling. An infrared range sensor is mounted in front of the robot to acquire robot's position. The results are shown in section 4.2.3 in Chapter 4.

### **3.2.6 Tests for vibration tracking of the flexible beam with camera**

As in the survey in section 2.2.2, there are numerous approaches that made use of camera for vibration estimation of flexible manipulators. In this section, we validate and compare the use of camera for estimating the beam vibration.

As explained in second part of the preparation phase, the setup is by fixing the Firefly series FMVU-03MTC-CS USB CMOS camera at the fixed end of the beam and an LED at the free end, so as to capture the image of oscillating LED. The distance from the lens of camera to the LED is 50 cm.

As for the webcam, the images of the LED light is very blur and there are frames drop issues, thus webcam is not suitable for vibration estimation of the flexible beam. Out of 3000 frames, more than half the frames were dropped. This caused position tracking of the oscillating LED light difficult for webcam. Therefore, webcam cannot be used for measuring vibration of flexible beam and thus only discussed the tests for vibration using camera only.

## **I. Test capturing of oscillating LED using camera**

The tests were carried by exciting the flexible beam to a deflection of 10 cm and then release. It was carried out in a dark room so that there is no surrounding light to effect the recognising of the LED position. Section 4.2.4 (chapter 4) illustrated the results for the tests.

## **II. Vibration tracking of the flexible beam using camera**

The vibration tracking tests were carried out by deflecting the tip of the beam by 4 cm and then release. For image processing operation, the coordinates of the LED light on the image plane are evaluated through the blob detection operation. An algorithm has been implemented to predict the next LED position, so that only a  $100 \times 20$  rectangular portion of the image is analyzed each cycle, which is the distance from the current LED position to the next LED position for highest image processing speed. This saves computational burden and image processing time. The algorithm is based on current position minus the previous position to obtain the velocity. And then use the velocity and current position to predict the next position. This method also solved the problem of distorted image due to high displacement speed as illustrated in Fig. 4.34 (section 4.2.4 in chapter 4). By knowing the heading direction of the LED, we can determine the current LED position from the distorted image. For example, by knowing the direction of motion we know that the current pixel position is for the middle LED position X: 427 and Y:211. The reference point is taken from the left top edge of the image, which is X: 0 and Y:0.

The results for the tests are depicted in section 4.2.5 in chapter 4.

## **3.3 Characterization of encoder**

As encoders and tachometers are well known devices for rotational displacement measurement, and the method for displacement measurement is relatively standard, thus only simple verifications were carried out here. Encoder data are accessed by computing the pulses count when the motor shaft turns.

### 3.3.1 Equation for computing encoder signal

For the wheel, the linear displacement of its circumference can be calculated through the encoder data using the following equation [27]:

$$d_{enc} = 2\pi r p_{enc} / n_{enc} \quad (3.12)$$

where

$d_{enc}$  = the linear displacement,

$r$  = the wheel radius,

$n_{enc}$  = the number of pulses per revolution, and

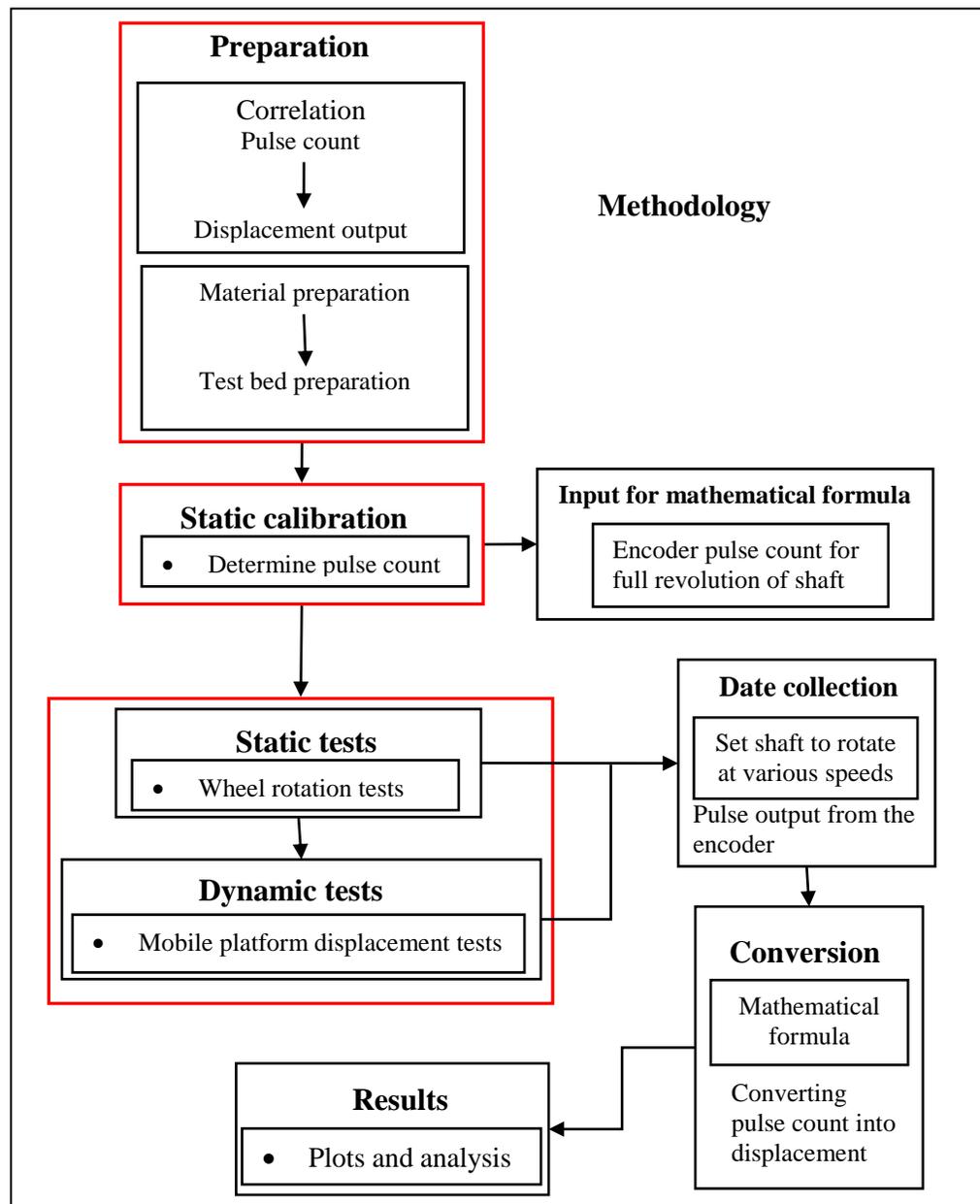
$p_{enc}$  = the number of pulses.

### 3.3.2 Methodology for testing encoder

In this research, encoder is used as a relative method for measuring the displacement of the mobile platform. The phases of the methodology for the tests are as follows:

- I. Preparation: In the preparation phase, a motor that comes with encoder was prepared. Setup the motor with encoder with appropriate voltage input and data acquisition instrument. Then, prepare the test bed for the encoder to be tested on. Refer to section 3.3.3 for the details of preparing the encoder.
- II. Calibration: Calibrate the encoder by accessing the number of pulse counts for the 360° turn of the shaft. Refer to section 3.3.4 for the details of calibrating the encoder.
- III. Static test: To test the accuracy of the conversion equation for the encoder pulse counts. Mount a wheel onto the motor shaft to test the accuracy of the encoder in computing the wheel's rotational displacements. Refer to part I of section 3.3.5 for details of the static tests.
- IV. Dynamic test: As a set objective in point 2 of section 1.2, we are required to the test the accuracy and repeatability of the sensors. To test the accuracy and repeatability

of the encoder in measuring the linear displacement of the mobile platform. Attach the motor with encoder onto the mobile platform. Carry out tests by applying voltage to drive the motor and measure the linear displacement of the platform that computed from the encoder output. This is to determine the accuracy of the encoder in computing the displacement of the mobile platform. Refer to part II of section 3.3.5 for details of dynamic tests.



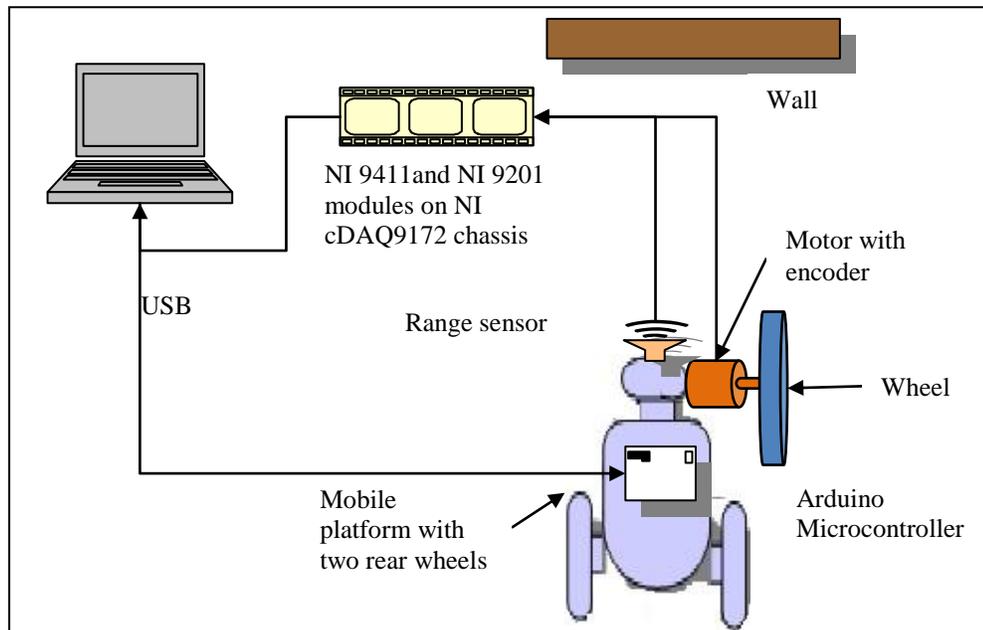
**Fig. 3.15** Flow chart of methodology for testing the encoder

The flow chart of the methodology for testing the encoder is illustrated in Fig. 3.15 above.

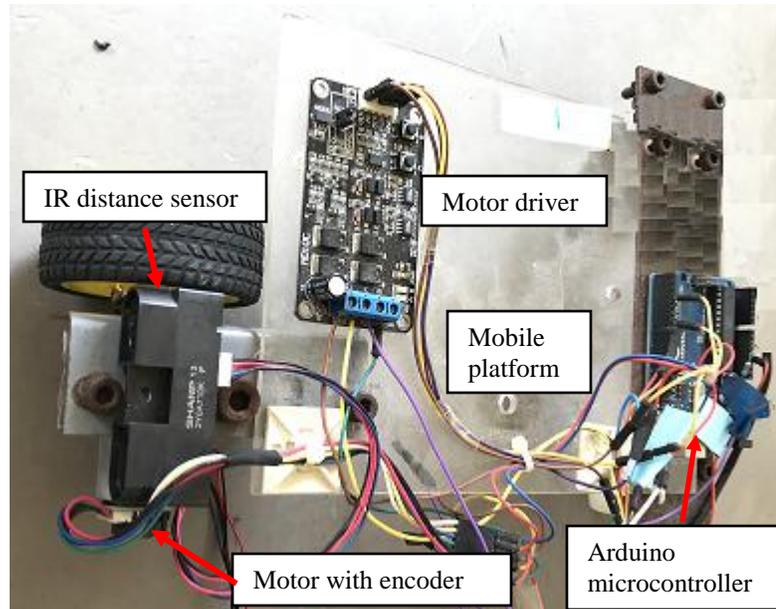
### 3.3.3 Preparation for testing the encoder

To test the encoder, a low cost brushed DC motor that comes with encoder was used. The motor is a geared motor, with gear ratio 46.85:1. The encoder is a rotary encoder with 2248.8 pulses per revolution. The motor comes with that sense the magnetic disk rotation of the motor shaft. The encoder's signal was acquired using NI 9411 via NI eDAQ 9172. A wheel is attached to the shaft of the motor where its angular displacement can be calculated with the encoder. The diameter of the wheel is 65 mm. The motor with the wheel is then attached to the mobile platform that has two follower rear wheels. A variable DC power of up to 5 V is used to drive the motor.

Fig. 3.16 illustrates the schematics for the test setup for testing the encoder on mobile platform. Fig. 3.17 depicts the setup for testing the encoder.



**Fig. 3.16** Schematics of setup for testing encoder



**Fig. 3.17** Setup for encoder test

### 3.3.4 Calibrate the encoder

The motor shaft was set to rotate  $360^\circ$ , while the pulse counts were collected. The pulse count is to verify the actual pulse per revolution, use for computing wheel displacement. Mathematical equation 3.12 is used to compute the linear displacement using the signal from the encoder.

### 3.3.5 Testing the encoder

To verify the encoder accuracy for use in robot localization, experiments were carried out. The static and dynamic tests for the encoder are as follows:

#### I. Static tests for the encoder – wheel angular displacement test

With a wheel attached to the motor shaft. The angular displacement of the wheel was computed from the encoder output. The wheel was allowed to turn freely without load, and tested with various input voltages to the motor. It was found that to prevent aliasing at higher motor speeds, the acquisition rate needs to be set to 1kHz.

The tests were carried out with the input voltages of 0.67V to 6V at increment of 0.67V. The results are illustrated in section 4.3.1 of chapter 4.

## **II. Dynamic tests for the encoder**

For the dynamic test, the motor with the wheel was affixed to a mobile platform. The objective is to determine the repeatability of the encoder in measuring the distance travelled by the platform. In the tests, the platform was programmed to move at various speeds (by applying various step input voltages) to move a fixed distance for each speed. It was tested with wheel rotating 370 mm linear displacement. The tests were carried out with 32 mm/s, 100 mm/s, 146 mm/s and 266 mm/s wheel's linear speeds. The results are illustrated in section 4.3.2 of chapter 4.

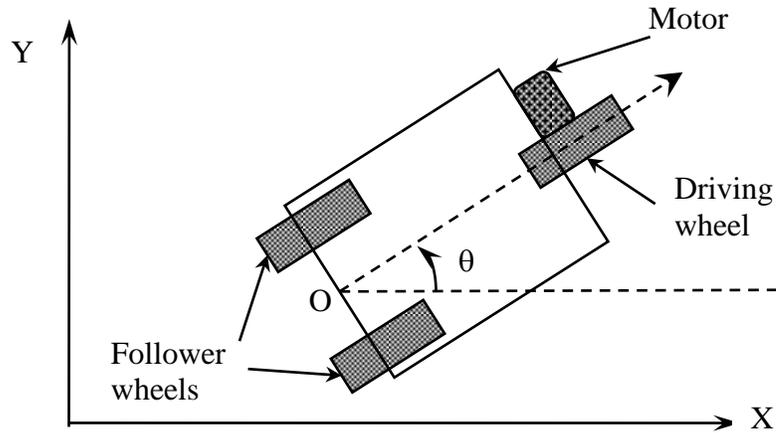
### **3.4 Modelling for Mobile Flexible Link Manipulator**

There are two parts for the mobile FLM: the mobile base and the flexible manipulator. The main task of the localization system is to provide accurate position estimation of the mobile platform, and accurately predict the vibrations of the flexible manipulator.

This and the next two sections covers the objective set in point 3 of section 1.2 in chapter 1, which is to model the mobile flexible link manipulator (MFLM) so that the selected sensors can be evaluated for vibration and motion trajectory prediction of the MFLM. First, the dynamic models of the mobile platform will be formulated, followed by the dynamic equation for the flexible beam.

#### **3.4.1 Dynamic Model of Mobile platform**

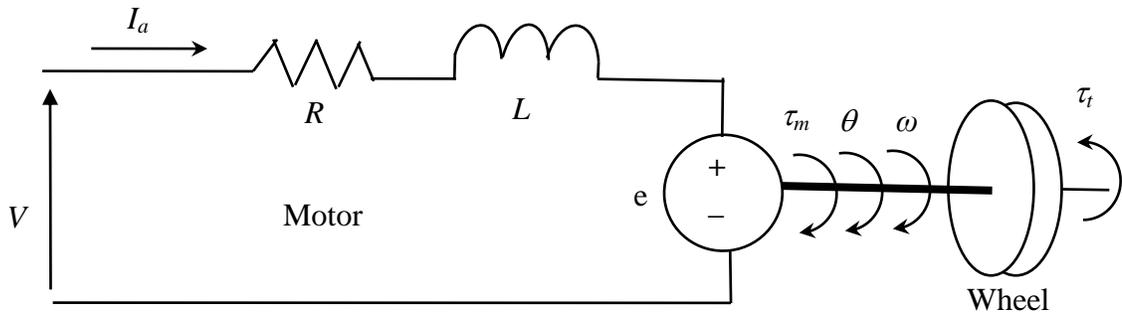
The mobile base used for the investigation is a three wheeled vehicle, as modelled in Fig. 3.16. The front wheel is the driver wheel (which is driven by a DC geared motor) that has tire. It composed of normal force, traction/braking force, rolling resistance force and lateral force. The rear follower wheels are two caster wheels, which are free rolling, thus the traction/braking and lateral forces are assumed be to negligible.



**Fig. 3.16** Free body diagram of a three wheels mobile platform (Note that this is the model of the mobile platform used in this research)

**a. Model of DC motor**

Assume that the load and input voltage to the brushed DC motor for the front wheel is equivalent to the DC motor circuit as shown in Fig. 3.17 below, where  $I_a$  is current,  $V$  is the input voltage,  $L$  is the armature inductance,  $R$  is the armature resistance and  $\tau_{coup}$  is the coupling torque from the wheel [84].



**Fig. 3.17** DC motor equivalent circuit [85]

Theoretically, a DC motor can be approximated as a circuit with back-EMF voltage, an inductance and a resistor. The resistor models the resistance of the motor windings. The back-EMF is the voltage generated by the moving electric current in the magnetic field. The back-EMF voltage is proportional to the speed of the motor. There is no back-EMF at stall. At no-load free rotation, the back-EMF can be assumed equal to the source input voltage. Therefore, we can assume that the back-EMF voltage  $e$  is proportional to the shaft angular velocity  $\omega_m$ , expressed as [84]:

$$e = k_e \omega_m \quad (3.13)$$

where

$k_e$  = the back-EMF constant of the motor,

$e$  = back-EMF of the motor,

$\omega_m$  = shaft angular velocity.

The motor torque is proportional to the armature current, expressed as [84]:

$$\tau_m = k_i I_a \quad (3.14)$$

where

$k_i$  = the torque constant of the motor (Nm/A),

$\tau_m$  = the torque generated by the motor,

$I_a$  = input the current to the motor.

Based on Kirchhoff's voltage law for the electrical characteristics of Fig. 3.17, yielded the equation [84]:

$$V = L \frac{dI}{dt} + RI_a + e = L \frac{dI}{dt} + RI_a + k_e \omega_m \quad (3.15)$$

where

$V$  = input voltage,

$L$  = armature inductance,

$R$  = armature resistance.

For the mechanical characteristics of Fig. 3.17, Newton's second law can be applied, which yields the equation [85]:

$$\tau_m = J_m \frac{d\omega_m}{dt} + T_{fr} + T_{coup} = k_i I_a \quad (3.16)$$

where

$J_m$  = moment of inertia of the motor,

$T_{coup}$  = coupling torque where the motor torque must exceed to begin the wheel movement,

$T_{fr}$  = friction which act against the movement.

$T_{fr}$  can be a static or kinetic friction [85]. It is dependent on the mass of the manipulator by the normal force  $N$ . Static friction is realized at zero velocity as a force threshold  $F_s$ , which must be overcome to set the mobile manipulator moving. This is a phenomenon concerning friction where the rotational system will only start to move when the driving torque is high enough to break the static frictional torque [85, 86]. Such characteristics of friction form a dead zone nonlinearity output with respect to the input voltage. The static friction can be represented by a coefficient  $\mu_s$  and normal force  $N_F$  as [87]:

$$F_s = \mu_s \cdot N_F \quad (3.17)$$

where

$F_s$  = static frictional force,

$\mu_s$  = coefficient of friction,

$N_F$  = normal force acting on the wheels.

Upon overcoming the static friction, the friction force becomes kinetic friction. The kinetic friction consists of stribek friction, Coulomb friction and viscous friction. The frictional torque of  $T_{fr}$  in equation (3.16) can thus be written as [87]:

$$T_{fr} = T_{static} + T_{stribek} + v_c \omega_m + T_{coul} \quad (3.18)$$

where

$T_{fr}$  = friction torque

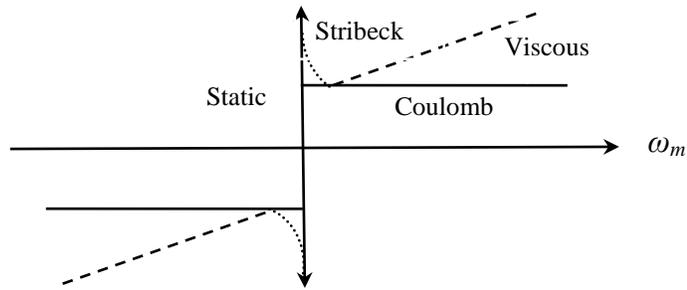
$v_c$  = viscous friction coefficient,

$T_{coul}$  = coulomb friction which affects the motor dynamics,

$T_{static}$  = static friction,

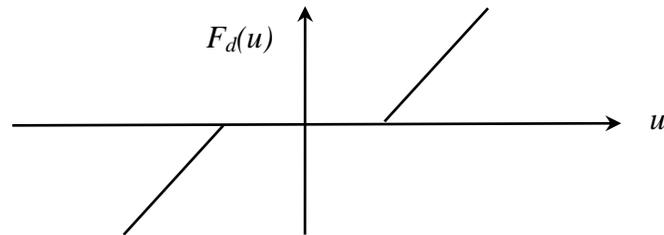
$T_{stribek}$  = stribek friction.

Fig. 3.18 illustrates the combination of static friction, stribek friction, viscous friction and coulomb friction [85]. In ideal case, the solid line is sufficient to represent the nonlinear friction which consists of coulomb friction.



**Fig. 3.18** Viscous friction, Coulomb friction and static friction [85]

Fig. 3.19 depicts the ideal dead zone nonlinearity, where the electro-mechanical system exhibits a nonlinear characteristic [87] when the input voltage to the motor undergoes zero volts, the vehicle stays still for some time until it achieved the breakthrough voltage. This means that at zero velocity, the striction (due to static friction) will oppose all motions until the torque is higher in magnitude than the striction torque. This is due to the fact that the electro-mechanical system cannot respond immediately to the input voltage from rest.



**Fig. 3.19** Dead zone characteristic [87]

The Coulomb friction causes the vehicle to be resistant to move, which is not dependent on shaft velocity of the motor [87]. Coulomb friction torque is a nonlinear friction, and is considered as a perturbation, depending only on the sign of the motor angular velocity (or direction of velocity). The following equation can be assumed for the Coulomb friction torque model [87]:

$$T_{coul} = \begin{cases} T_c \cdot \text{sign}(\dot{\theta}_m), & |\dot{\theta}_m| > 0 \\ \tau_m - T_{coup}, & |\dot{\theta}_m| = 0 \text{ and } |\tau_m - T_{coup}| \leq |T_c| \\ T_c \cdot \text{sign}(\tau_m - T_{coup}), & |\dot{\theta}_m| = 0 \text{ and } |\tau_m - T_{coup}| > |T_c| \end{cases} \quad (3.19)$$

where

$T_c$  = an unknown constant value which is different for each motor, which is also dependent on mass of the manipulator,

$\dot{\theta}_m$  = angular displacement rate of motor shaft.

Thus,  $T_{coul} = \mu_{coul}N$ , where  $\mu_{coul}$  is coulomb friction coefficient. The first case of equation (3.19) shows that Coulomb friction torque when the motor is turning, and the other two cases show the same torque when the motor is stopped. The Coulomb friction causes the vehicle to be resistant to the direction of motion [87].

The viscous friction is dependent and is a linear function of the rotational velocity of the motor shaft, which is represented as  $\nu\omega_m$  in equation (3.18). It is also dependent on the mass of the mobile manipulator, thus  $\nu = \mu_v N$ , which is the viscous coefficient multiply by normal force  $N$ . The stribeck friction occurs only at low velocity, and it is the apparent drop in the friction force as the velocity increases [85]. Stribeck friction is usually expressed by the following equation [86]:

$$T_{fr} = (T_{coul} + (T_{static} - T_{coul})e^{-|w_m|/v_s})\text{sign}(w_m) + \nu\omega_m \quad (3.20)$$

where

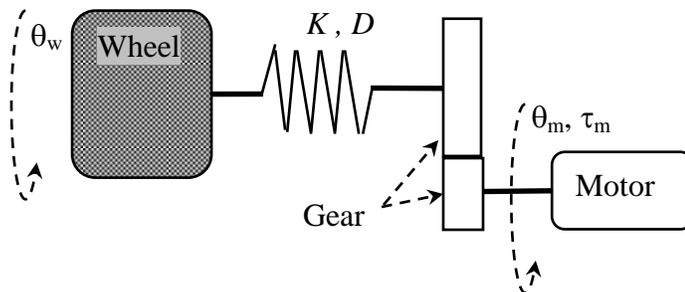
$T_{coul}$  = torques for coulomb friction,

$T_{static}$  = static friction,

$v_s$  = stribeck velocity,

$\nu\omega_m$  = viscous friction

$i$  = an exponent.



**Fig. 3.20** Model of front motor and its wheel [78]

With a geared DC motor, it is common to visualize that the transmission from the motor to the driving wheel can be approximated with a model of a two-mass resonant system connected by a spring and damping system [78], coupling the motor and wheel, having the constants of  $K$  for the spring and  $D$  for the damper, as represented in Fig. 3.20. Equating the coupling torque of the motor  $T_{coup}$  in terms of  $K$  and  $D$ , equation (3.16) becomes:

$$\tau_m = K_i I_a = J_m \ddot{\theta}_m + T_{fr} + \frac{K}{\eta^2} (\theta_m - \eta \theta_w) + \frac{D}{\eta^2} (\dot{\theta}_m - \eta \dot{\theta}_w) \quad (3.21)$$

where

$K$  = constant for the spring,

$D$  = constant for the damper,

$\eta$  = the gear ratio,

$\theta_w$  = the wheel angle,

$\theta_m$  = the shaft angle,

$J_m$  = motor's moment of inertia.

$T_{coup}$  in equation (3.16) is  $\frac{K}{\eta^2} (\theta_m - \eta \theta_w) + \frac{D}{\eta^2} (\dot{\theta}_m - \eta \dot{\theta}_w)$ . Let  $J_w$  be the wheel inertia, the dynamic equation for the shaft of driving wheel is [85]:

$$J_w \ddot{\theta}_w = K \left( \frac{\theta_m}{\eta} - \theta_w \right) + D \left( \frac{\dot{\theta}_m}{\eta} - \dot{\theta}_w \right) = \tau \quad (3.22)$$

where

$\tau$  = coupling torque measured in the shaft connecting to the wheel,

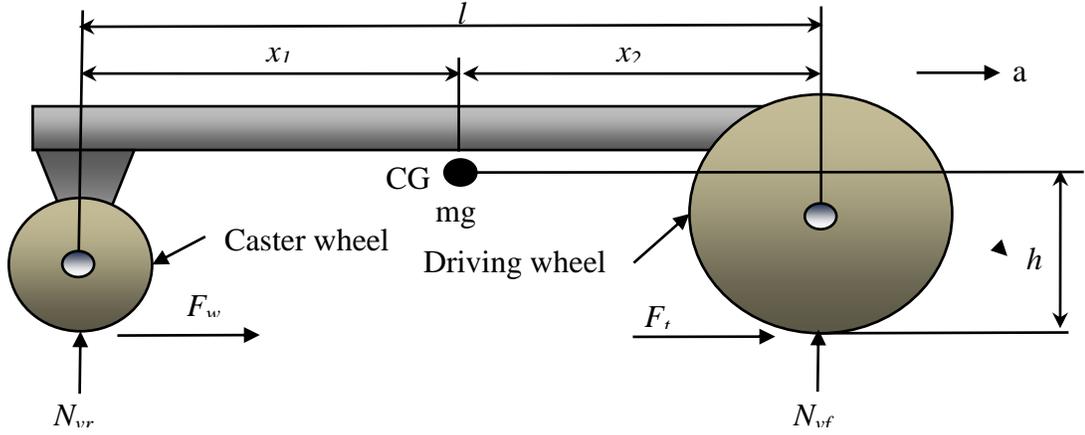
$J_w$  = wheel's moment of inertia.

Thus, substituting equation (3.22) into (3.21), leads to:

$$\tau_m = K_i I_a = J_m \ddot{\theta}_m + T_{fr} + \frac{\tau}{\eta} \quad (3.23)$$

## b. Model of wheels

The possible contributions to motion inaccuracy for a vehicle are wheel slippages and backlashes due to reduction gears. Here, we investigate the wheel dynamics with the ground, as illustrated in Fig. 3.21.



**Fig. 3.21** Mobile platform side view illustrating centre of gravity

(Note that this diagram illustrates the model of the mobile platform used in this research)

As we limit our study to the robot moving only longitudinally, thus the steering dynamics of the wheel are not considered. Only the rolling dynamics of the wheel are considered. Referring to Fig. 3.21, summing forces in the vertical direction, leads to [88]:

$$N_{vr} + N_{vf} - mg = 0 \quad (3.24)$$

where

- $N_{vr}$  = normal force acting on rear wheel,
- $N_{vf}$  = normal force acting on front wheel
- $m$  = mass of the mobile platform
- $g$  = gravity

Summing forces in the horizontal direction leads to [88]:

$$F_t + F_w = ma \quad (3.25)$$

Summing moments about the centre of gravity CG, leads to [50]:

$$N_{vr} \cdot x_1 - N_{vf} \cdot x_2 - F_t \cdot h - F_w \cdot h = 0 \quad (3.26)$$

where

- $F_t$  = traction force acting on front wheel due to rolling,
- $F_w$  = friction force for rear wheel,
- $N_v$  = normal force from the ground,

$f$  = subscript represent front wheel,  
 $r$  = subscripts represent rear wheel,  
 $x_1$  = distance from centre of gravity to rear wheel,  
 $x_2$  = distance from centre of gravity to front wheel,  
 $h$  = height from ground to centre of gravity.

Rearranging equation (3.24) leads to:

$$N_{vr} = mg - N_{vf} \quad (3.27)$$

Rearranging equation (3.26) leads to:

$$N_{vf} \cdot x_2 = N_{vr} \cdot x_1 - F_t \cdot h - F_w \cdot h \quad (3.28)$$

Substituting equation (3.27) to (3.28), and simplifying for the normal force on the front wheel, leads to:

$$N_{vf} = \frac{mg \cdot x_1 - F_t \cdot h - F_w \cdot h}{l} \quad (3.29)$$

Where

$l$  = length measured from front wheel to rear wheel.

Substituting equation (3.28) to (3.27) to solve for normal force  $N_{vr}$  on the rear wheel, and simplifying leads to:

$$N_{vr} = \frac{mg \cdot x_2 + F_t \cdot h + F_w \cdot h}{l} \quad (3.30)$$

Accordingly, the normal forces for each of the two rear wheels are [50]:

$$\begin{aligned}
 N_{vrr} &= \frac{mg \cdot x_2 + (F_t + F_w) \cdot h}{2l} \\
 N_{vrl} &= \frac{mg \cdot x_2 + (F_t + F_w) \cdot h}{2l}
 \end{aligned} \quad (3.31)$$

where

$rr$  = subscripts represent rear right wheel,

$rl$  = subscripts represent left wheel.

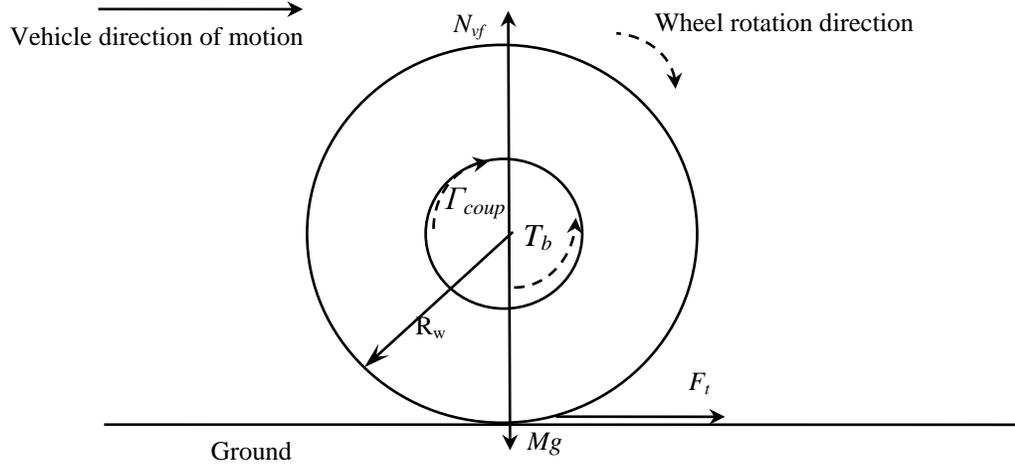
The friction for the front wheel (Fig. 3.22) with respect to ground is expressed as [89]:

$$F_t = \mu_d(\lambda) N_{vf} \quad (3.32)$$

where

$\lambda$  = the wheel slip value,

$\mu_d$  = the adhesion coefficient.



**Fig. 3.22** Driving wheel dynamic with the ground [89]

With  $R_w$  as the radius of front wheel, the torque at the front wheel to move the vehicle will be [89]:

$$\tau_f = F_t R_w = \mu_d(\lambda) N_{vf} R_w \quad (3.33)$$

For the rear wheels, the torque needs to break the force that acts with a moment equal to the radius of the wheel. As illustrated in Fig. 3.23, for the vehicle to move we need to overcome the frictions acting at the caster wheel.  $F_w$  is the friction force between the wheel and the ground, while  $F_a$  is the friction force between the wheel and the axle. Consequently, we can determine the torque between the wheel and ground as  $T_w$ , and the torque between the wheel and axle as  $T_a$ .  $\mu_a$  and  $\mu_w$  are the coefficient of friction for the axle and ground, respectively.  $\mu_a$  needs to be smaller than  $\mu_w$  so that the wheel will not slide across the ground, and to roll. Therefore, we need only  $F_a$  to compute the required torque at the caster wheels. In order to relate the axle force  $F_a$  to required wheel torque to rotate the caster wheel without sliding along the ground, virtual radius  $R_v$  of the wheel/axle combination is used to compute as follows [90]:

$$R_v = R_w - R_a \quad (3.34)$$

where,

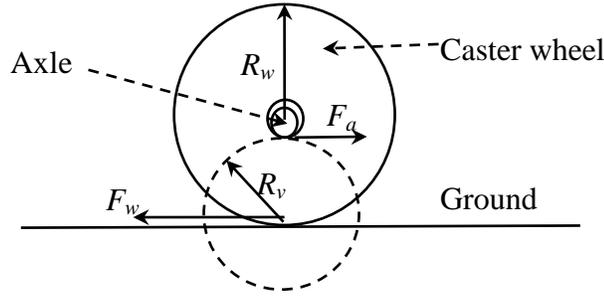
$R_w$  = the wheel radius,

$R_a$  = the axle radius,

$R_v$  = the fictitious radius which  $F_a$  would act to rotate the wheel about the tangent point in contact with the ground.

The equation for the torque at the caster wheels  $\tau_c$  is [89]:

$$\tau_c = F_a R_v = F_a (R_w - R_a) = N_{vrr} \mu_a(\lambda) (R_w - R_a) \quad (3.35)$$



**Fig. 3.23** Caster wheel with respect to ground [88]

If there is no wheel slip, the motor torque must be less than the maximum torque that can be supplied by the friction with the ground [88], then the rewrite equation (3.22) for dynamic equation of motion for the driving (front) wheel with respect to ground as:

$$J_w \ddot{\theta}_w = \Gamma - R_w F_t - R_c F_w \quad (3.36)$$

where

$R_w$  = the front wheel radius,

$R_c$  = the rear wheel radius

$\Gamma$  = the shaft torque at the wheel.

### c. Vehicle dynamics

As we only consider the robot moving longitudinally; hence the Coriolis and centrifugal forces acting on the vehicle can be ignored. The platform travels in slow speed at indoor

environment, thus the aerodynamic effect on the mobile platform can be ignored. The lateral forces for rear wheels can also be neglected because they spin without friction and thus usually align with their velocity vectors. Only the normal forces acting on the wheel are considered. For the front wheel, the tire forces are composed of a normal resultant force, traction/braking torque, rolling resistance and lateral force. We only consider the vehicle travelling in flat ground, thus rolling resistance is negligible. Accordingly, the gravitational force is also not considered. The transverse movement is negligible, so the lateral force can also be ignored. A flexible link manipulator is mounted on the platform, thus there could be a disturbance force ( $F_{disturb}$ ) due to vibration of the manipulator. The vehicle dynamics equation is [90]:

$$\dot{v}_{el}m = N_w F_{toq} - F_{disturb} \quad (3.37)$$

where

$v_{el}$  = velocity of the mobile platform,

$m$  = mass of the mobile platform,

$N_w$  = the number of driving wheels during acceleration or braking.

By referring to Newton's second law for the motion of a particle, we can state that the resultant force is the external forces ( $F_x$ ,  $F_y$  and  $F_z$ ) on the system of mass equals the total mass of the system multiplies the acceleration of the centre of mass. The governing equations of the vehicle for longitudinal, lateral, and vertical, motions can be expressed as [50, 88]:

$$F_{toq} = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = m \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = m \begin{bmatrix} \ddot{X} + \omega_y \dot{Z} - \omega_z \dot{Y} \\ \ddot{Y} + \omega_z \dot{X} - \omega_x \dot{Z} \\ \ddot{Z} + \omega_x \dot{Y} - \omega_y \dot{X} \end{bmatrix} = \begin{bmatrix} XF_f + XF_{rl} + XF_{rr} \\ YF_f + YF_{rl} + YF_{rr} \\ ZF_f + ZF_{rl} + ZF_{rr} \end{bmatrix} \quad (3.38)$$

The terms  $XF_i$ ,  $YF_i$  and  $ZF_i$ , which is  $\frac{J_w \ddot{\theta}_w}{R_w}$  (where  $J_w \ddot{\theta}_w$  is from equation (3.36)), are the wheel forces in the x, y and z directions respectively. Subscript  $i$  represent  $f$ ,  $rl$  or  $rr$ , denote front, rear left and rear right. In this equation, we ignored the roll, pitch and yaw moments of the vehicle since it will be travelling on levelled ground and assuming no turning movements. Therefore, equation (3.38) can be used to determine the acceleration and velocity of the mobile platform.

The total required torque at the three wheels to move the mobile platform is the summation of torque at the front wheel (equation (3.33)) and the torque at the rear wheels (equation (3.35)), which yields:

$$\tau_t = \tau_f + 2\tau_c \quad (3.39)$$

where

$\tau_t$  = total torque acting on the three wheels of the mobile platform,

$\tau_f$  = torque action on front wheel,

$\tau_c$  = torque action on rear wheel.

Next, we determine the acceleration force necessary to accelerate the vehicle from the stop to maximum speed. As the force is driven at the wheel, we multiple the acceleration force (equation (3.37)) by the front wheel radius  $R_w$ , we get the acceleration torque for the mobile platform [89]:

$$\tau_{acc} = \dot{v}_{el} m R_w \quad (3.40)$$

Where

$\tau_{acc}$  = acceleration torque for the mobile platform.

Adding equation (3.39) and (3.40) and substitute to  $\tau$  in equation (3.23) for the front wheel torque, we get the final motor torque equation as:

$$\tau_m = k_i I_a = J_m \ddot{\theta}_m + T_{fr} + \frac{(\tau_t + \tau_{acc})}{\eta} \quad (3.41)$$

#### d. Motor speed calculation

Taking Laplace transform [91] for equation (3.15), gives:

$$V(s) = LsI_a(s) + RI_a(s) + k_e s \theta(s) \quad (3.42)$$

The frictional torque  $T_{fr}$  contains viscous friction which is dependent on angular velocity of motor shaft, which will affect the motor speed calculation. Thus, we split equation (3.18) into constant friction and varying friction, where constant friction is  $T_{frc} = T_{static} + T_{stribek} + T_{coul}$ , leading to:

$$T_{fr} = T_{frc} + v_c \omega_m \quad (3.43)$$

Substituting equation (3.43) into (3.41), leads to:

$$k_i I_a = J_m \ddot{\theta}_m + T_{frc} + v_c \omega_m + \frac{(\tau_t + \tau_{acc})}{\eta} \quad (3.44)$$

Let  $T_m = T_{frc} + \frac{(\tau_t + \tau_{acc})}{\eta}$ , then taking Laplace transform of Equation (3.44), yields:

$$k_i I(s) = J_m s^2 \theta(s) + v_c s \theta(s) + T_m \quad (3.45)$$

where

$T_m$  = load torque from the mobile platform.

Substituting equation (3.42) into (3.45), and rearranging leads to:

$$J_m s^2 \theta(s) + v_c s \theta(s) = k_i \left[ \frac{V(s) - k_e s \theta(s)}{Ls + R} \right] - T_m(s) \quad (3.46)$$

For the DC motor alone, we can assume that  $T_m = 0$ . Thus, rearranging equation (3.46) to with input voltage,  $V(s)$ , to the output angle  $\theta$ . With  $W$  as angular velocity of  $\theta$ , we obtains the transfer function for the motor as:

$$\frac{\dot{\theta}(s)}{V(s)} = \frac{W(s)}{V(s)} = \frac{k_i}{(J_m s + v_c)(R + Ls) + K_e k_i} \quad (3.47)$$

When the motor is attached to the robotic system for driving the front wheel, then  $T_m \neq 0$ . Consequently, the transfer function for the entire robotic system is:

$$G(s) = \frac{W(s)}{V(s)} = \left[ \frac{k_i}{(J_m s + v_c)(R + Ls) + K_e k_i} \quad \frac{-(R + Ls)}{(J_m s + v_c)(R + Ls) + K_e k_i} \right] \quad (3.48)$$

where

$\mathcal{G}(s)$  = Laplace transforms of output  $\omega(t)$ ,

$V(s)$  = Laplace transforms of voltage inputs  $v(t)$ .

The inputs are  $V(s)$  and  $T_m(s)$  for voltage input and load torque from the mobile platform, respectively. The output is the motor angular velocity,  $\omega(s)$ . The moment of inertia  $J_m$  due to the entire robotic system will then be the moment of inertia of motor plus the moment of inertia due to the weight of the entire system. We've assumed that the mass of the spokes is negligible compared to the mass acting on the wheel, therefore the moment of inertia of the front wheel can be written as [88]:

$$J_m = J_m' + m_{total} R_w^2 \quad (3.49)$$

where

$J_m'$  = moment of inertia for the motor,

$m_{total}$  = total mass of the entire robot,

$R_w$  = radius of the front wheel.

The speed of the mobile platform can be calculated by using equation (3.47) for  $\omega(s)$ , multiplying with the front wheel radius leads to [88]:

$$v_{el} = \omega(s)R_w \quad (3.50)$$

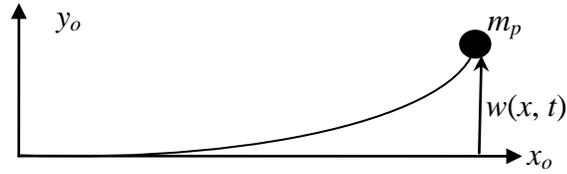
### 3.5 Model of flexible link manipulator

In this work, the dynamic models for the mobile FLM are derived. The flexible manipulator is a nonlinear system that can be described using partial differential equations. It has infinite number of degrees-of-freedom. However, it is not realistic to use infinite dimensional models for real-time applications, thus finite dimensional models having minimum number of parameters is preferred. Detail discussion of the available models is beyond the scope of this work, but can be found in Kiang. et al. [14]. Instead, only the three important elastic models are briefly introduced as follows:

1) Finite Element Models [22]: These models are the most accurate, but due to their complexity it is usually only used for simulation purpose. Finite element method is a general numerical technique for solving differential equations.

2) Assumed Modes Models [92]: These models are derived from the partial differential equation by model truncation. In assumed mode method (AMM), by selecting a few low order modes can establish a lower dimensional system dynamics model. AMM are useful for simulation and control design and thus are used in this work for designing the model of the system.

3) Lumped Parameter models [93]: The discrete, localized springs is used to model the elasticity of the FLM. Non-actuated joints link together a number of rigid bodies to form a flexible link. Lumped parameter model is not used in this work.



**Fig. 3.24** Schematic representation of the flexible manipulator [94]

In this section, a mathematical model is derived for the flexible link manipulator (FLM) using the AMM. The FLM is a flexible beam as shown in Fig. 3.24, with one end fixed and a payload of  $m_p$  at the free end [94]. When acted by a force at the free end of the flexible beam, the beam deflection can be determined using the following equation [95]:

$$w_{max} = \frac{PL^3}{3EI} \quad (3.51)$$

where

- $w_{max}$  = maximum deflection of the beam,
- $P$  = force acting at the free end of the beam,
- $E$  = Young's modulus,
- $I$  = area moment of inertia.

The gravity effect can be neglected because is placed horizontally so that the flexible beam can only vibrate freely horizontally. The area along the beam is constant, and the beam is thin and long, so the deformation due to the rotary inertia, shear and the effects of axial forces are assumed negligible, so can be neglected as well. Therefore, the Euler-Bernoulli beam theory (which applies for thin beam theory, where the rotary inertia and shear deformation are neglected) can be used to model the elastic behaviour of the beam. One end is fixed, and the displacement at the tip of the flexible link is designated as  $w(x, t)$ . Based on Euler-Bernoulli beam model, the governing equation of motion is given by [96, 97]:

$$m \frac{\partial^2 w}{\partial t^2} + EI \frac{\partial^4 w}{\partial x^2} = f(x, t) \quad (3.52)$$

where

- $w$  = transverse deflection of flexible beam,
- $m$  = mass of beam,

$E$  = Young modulus of elasticity,

$I$  = second moment of area of the cross-section of the beam,

$f(x, t)$  = load at the tip.

In the case of free vibration of the beam,  $f(x, t) = 0$ , and the equation (3.52) reduces to:

$$m \frac{\partial^2 w}{\partial t^2} + EI \frac{\partial^4 w}{\partial x^2} = 0 \quad (3.53)$$

Equation (3.53) is a classical undamped Euler-Bernoulli beam theory [96]. Using the assumed model solution [95] for separating the variables, we assume that the displacement can be separated into two parts; one depends on time and another on position. Thus, the equation for the beam deflection [96, 98],

$$w(x, t) = \Phi(x)Q(t) \quad (3.54)$$

where

$w(x, t)$  = deflection of beam,

$\Phi(x)$  = a function which depends only on space,

$Q(t)$  = a function which depends only on time.

For a cantilever beam subjected to free vibration, and for a prismatic homogeneous beam, the equation of motion can be written as [98],

$$\frac{d^2}{dx^2} \left\{ EI(x) \frac{d^2 \Phi(x)}{dx^2} \right\} = \omega_n^2 m(x) \Phi(x) \quad (3.55)$$

Substituting equation (3.54) into equation (3.53), and together with equation (3.55) and rearranging, leads to:

$$C \frac{\Phi_i''''(x)}{\Phi_i(x)} = \frac{\ddot{Q}_i(t)}{Q_i(t)} = \omega_{ni}^2, \quad C = \sqrt{\frac{EI}{m}} \quad (3.56)$$

Next, we define [96]:

$$\beta_i^4 = \frac{\omega_i^2}{c^2} \quad (3.57)$$

This leads to two ODEs (ordinary differential equations) [95]:

$$\frac{d^2 Q}{dt^2} + \omega_n^2 Q = 0 \quad (3.58)$$

$$\frac{d^4 \Phi}{dx^2} - \beta^4 \Phi = 0 \quad (3.59)$$

The general solutions to the above equations are given as [99]:

$$\Phi(x) = C_1 \sin(\beta x) + C_2 \cos(\beta x) + C_3 \sinh(\beta x) + C_4 \cosh(\beta x) \quad (3.60)$$

and

$$Q(t) = C_5 \sin(\omega_d t) + C_6 \cos(\omega_d t) = C \sin(\omega_d t + \phi) \quad (3.61)$$

where

$$\omega_d = \omega_n \sqrt{1 - \zeta^2} \text{ is damped natural frequency,}$$

$$\phi = \text{phase shift.}$$

For the equation of motion for the flexible beam given by equation (3.53), the boundary conditions can be written as [99]:

$$y(0, t) = 0, \quad \frac{\partial y}{\partial x}(0, t) = 0, \quad \frac{\partial^2 y}{\partial x^2}(L_b, t) = 0, \quad EI \frac{\partial^3 y}{\partial x^3}(L_b, t) = m \frac{\delta^2(L_b, t)}{\delta t^2} \quad (3.62)$$

Substituting equation (3.54) into equation (3.53) and (3.62), the boundary-value problem can be obtained as follows [99]:

$$\Phi = 0, \quad \frac{\partial \Phi(0)}{\partial x} = 0, \quad \frac{\partial^2 \Phi(L_b)}{\partial x^2} = 0 \quad \frac{\delta^3 \Phi(L_b)}{\delta x^3} = -\frac{m \omega_n}{EI} \Phi(L_b), \quad \frac{\partial^4 \Phi}{\partial x^2} - \frac{m \omega_n \Phi}{EI} = 0, \quad (3.63)$$

where

$$L_b = \text{length of the beam.}$$

Applying equation (3.63) into equation (3.60), and performing mathematical manipulations, yields [95, 99]:

$$\Phi(x) = \sin(\beta x) - \sinh(\beta x) - \hat{C} (\cos(\beta x) - \cosh(\beta x)) \quad (3.64)$$

where

$$\hat{C} = -\frac{\sin(\beta L_b) + \sinh(\beta L_b)}{\cos(\beta L_b) + \cosh(\beta L_b)} \quad (3.65)$$

This is an undamped system. Next, we show that the dynamic behaviour of the cantilever beam can be modelled as a second order system, governed by the second-order ODE as [98, 100]:

$$a_2 \frac{d^2 w}{dt^2} + a_1 \frac{dw}{dt} + a_0 w = F(t) \quad (3.66)$$

where  $w$  is the response of the system (which is the beam deflection) subjected to an input force  $F$ , and  $a_0$ ,  $a_1$  and  $a_2$  are the system parameters. We rewrite equation (3.66) as [98, 100]:

$$\ddot{w} + 2\zeta\omega_n\dot{w} + \omega_n^2 w = k_g\omega_n^2 F \quad (3.67)$$

where

$$\omega_n = \sqrt{\frac{a_0}{a_2}} \quad \text{undamped natural frequency} \quad (3.68)$$

$$\zeta = \frac{a_1}{a_2} \quad \text{damping ratio} \quad (3.69)$$

$$k_g = \frac{1}{a_0} \quad \text{gain} \quad (3.70)$$

Considering the case where  $F(t) = 0$ , equation (3.67) becomes

$$\ddot{w} + 2\zeta\omega_n\dot{w} + \omega_n^2 w = 0 \quad (3.71)$$

Comparing equation (3.71) with equation (3.58), we can see that the vibrating deflection of equation (3.58) is a second-order system with an undamped natural frequency  $\omega_n$  without damping included in the model. Equation (3.71) has solutions of the form [95]:

$$w = Ae^{\lambda t} \quad (3.72)$$

Equation (3.71) has two roots in the following form [98, 100]

$$\lambda = -\zeta\omega_n \pm \omega_n\sqrt{\zeta^2 - 1} \quad (3.73)$$

Our system is an under-damped system with  $0 < \zeta < 1$  [98, 100]. Substituting equation (3.73) into equation (3.72), leads to [95]:

$$w = A_1 e^{\left(-\zeta\omega_n + \omega_n\sqrt{\zeta^2 - 1}\right)t} + A_2 e^{\left(-\zeta\omega_n - \omega_n\sqrt{\zeta^2 - 1}\right)t} \quad (3.74)$$

where  $A_1$  and  $A_2$  can be determined from the initial conditions. Equation (3.74) can be expressed as [95]:

$$w = Ae^{-\zeta\omega_n t} \sin(\omega_d t + \phi) \quad (3.75)$$

where

$$A = \frac{\sqrt{(v_o + \zeta\omega_n w_o)^2 + (y_o \omega_d)^2}}{w_d^2} \quad (3.74)$$

$$\phi = \tan^{-1} \left( \frac{y_o \omega_d}{v_o + \zeta\omega_n x_o} \right) \quad (3.77)$$

$$\omega_d = \omega_n \sqrt{1 - \zeta^2} \quad (3.78)$$

$$\omega_n = \sqrt{\frac{k}{m}} \quad (3.79)$$

$$\zeta = \frac{D}{2\sqrt{km}} \quad (3.80)$$

$$k_{stiff} = \frac{3EI}{L^3} \quad (3.81)$$

where

$v_o$  = initial velocity at tip of beam,

$w_o$  = initial displacement at tip of beam,

$\omega_n$  = natural frequency at tip of beam,

$\omega_d$  = damped natural frequency,

$m$  = mass of beam,

$k_{stiff}$  = beam stiffness,

$D$  = damping coefficient,

$E$  = Young's modulus,

$I$  = area moment of inertia of the beam.

Finally, by substituting equation (3.64) and (3.75) into equation (3.54), leads to:

$$w(x, t) = A e^{-\zeta\omega_n t} \sin(\omega_d t + \phi) \times (\sin(\beta x) - \sinh(\beta x) - \hat{C} (\cos(\beta x) - \cosh(\beta x))) \quad (3.82)$$

The Rayleigh's method [95] is used to find the natural frequency, which can be summarized as [96]

$$(KE)_{\max} = (PE)_{\max} = \text{total energy of the system} \quad (3.83)$$

where,

$KE$  = kinetic energy

$PE$  = potential energy.

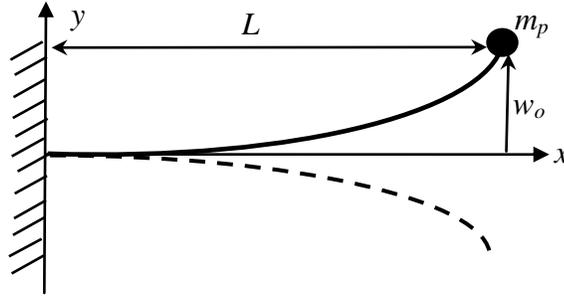
For the case of a beam, potential energy is also referred to as strain energy. Equation (3.83) can only be satisfied if the system is vibrating at its natural frequency. First, we propose a quarter cosine wave solution [101] (see Fig. 3.25) for equation (3.53), yielding [98, 100]:

$$w(x) = w_o \left[ 1 - \cos\left(\frac{\pi x}{2L}\right) \right] \quad (3.84)$$

$$\frac{dw}{dx} = w_o \left(\frac{\pi}{2L}\right) \sin\left(\frac{\pi x}{2L}\right) \quad (3.85)$$

$$\frac{d^2w}{dx^2} = w_o \left(\frac{\pi}{2L}\right)^2 \cos\left(\frac{\pi x}{2L}\right) \quad (3.86)$$

$$\frac{d^3w}{dx^3} = -w_o \left(\frac{\pi}{2L}\right)^3 \sin\left(\frac{\pi x}{2L}\right) \quad (3.87)$$



**Fig. 3.25** Quarter cosine wave was a Rayleigh shape for a cantilever

The manipulator system considered in the current work have the payload deforming only in horizontally, thus the gravity effect can be neglected. The total potential energy  $PE$  in the beam is [98, 100, 101]

$$PE = \frac{EI}{2} \int_0^L \left(\frac{d^2w}{dx^2}\right)^2 dx \quad (3.88)$$

Substituting equation (3.84) into equation (3.86), leads to

$$PE = \frac{EI}{2} \int_0^L \left[ w_o \left(\frac{\pi}{2L}\right)^2 \cos\left(\frac{\pi x}{2L}\right) \right]^2 dx \quad (3.89)$$

After performing mathematic simplification, yields

$$PE = \frac{1}{64}\pi^4 \left[ \frac{EI}{L^3} \right] (w_o)^2 \quad (3.90)$$

This expression states the elastic deformation forms the internal energy of the link as it bends. Because only motion in the plane perpendicular to the gravitational field is considered, the potential energy due to gravity is not accounted. Considering tip payload and mobile base movement, the total kinetic energy  $KE$  is [98, 100, 101]

$$KE = \frac{1}{2}\rho A \omega_n^2 \int_0^L (w)^2 dx \quad (3.91)$$

Substituting equation (3.84) into equation (3.91), yields

$$KE = \frac{1}{2}\rho A \omega_n^2 \int_0^L (w_o \left[ 1 - \cos\left(\frac{\pi x}{2L}\right) \right])^2 dx \quad (3.92)$$

Again, performing mathematic simplification, yields

$$KE = \frac{1}{4}\rho A \omega_n^2 L \left[ 3 - \frac{8}{\pi} \right] (w_o)^2 \quad (3.93)$$

Applying equation (3.83), we equate the  $PE$  and  $KE$

$$\frac{1}{64}\pi^4 \left[ \frac{EI}{L^3} \right] (w_o)^2 = \frac{1}{4}\rho A \omega_n^2 L \left[ 3 - \frac{8}{\pi} \right] (w_o)^2 \quad (3.94)$$

Solving for natural frequency, yields

$$\omega_n = \sqrt{\left\{ \frac{\pi^4 \left( \frac{EI}{L^3} \right)}{16\rho AL \left[ 3 - \left( \frac{8}{\pi} \right) \right]} \right\}} \quad (3.95)$$

$$f_n = \frac{1}{2\pi} \sqrt{\left\{ \frac{\pi^4 \left( \frac{EI}{L^3} \right)}{16\rho AL \left[ 3 - \left( \frac{8}{\pi} \right) \right]} \right\}} \quad (3.96)$$

Performing mathematical simplification, leads to [95]:

$$f_n = \frac{1}{2\pi} \sqrt{\left\{ \frac{\pi^4 \left( \frac{EI}{L^3} \right)}{16\rho AL \left[ 3 - \left( \frac{8}{\pi} \right) \right]} \right\}} \quad (3.97)$$

$$f_n = \frac{3.664}{2\pi L^2} \sqrt{\frac{EI}{\rho A}} \quad (3.98)$$

where

$f_n$  = natural frequency

We need to determine the effective mass of the beam in order to compute the beam vibration with varying payload at the tip, where a discrete, end-mass represent the distributed mass of the beam. Solving equation (3.79) for effective mass  $m_e$ , yields

$$m_e = \frac{k}{\omega_n^2} = \frac{k}{(2\pi f_n)^2} \quad (3.99)$$

Substituting equations (3.81) and (3.98) into equation (3.99), and performing mathematical simplification yields

$$m_e = 0.2235m_b \quad (3.100)$$

where

$m_b$  = mass of the beam,

$m_e$  = effective mass of the beam.

To include the payload at the tip of the beam, we sum the effective mass plus the payload to form total mass at the tip, leading to

$$m = 0.2235m_b + m_p \quad (3.101)$$

where

$m_p$  is the payload.

To compute the natural frequency of the beam we substituting equation (3.81) into (3.99), this yields the following equation:

$$f_n = \frac{1}{2\pi} \sqrt{\frac{3EI}{(0.2235m_b + m_p)L^3}} \quad (3.102)$$

### 3.6 Model of mobile flexible link manipulator

This section derives the model of entire system consists of the flexible link mounted on top of the mobile platform. Newton's method that is directly related to Newton's 2nd law and Lagrange's method that has its root in the classical work of d'Alembert and Lagrange on analytical mechanics are generally two major methods that can be used to derive the dynamic equations of mechanical systems. Lagrange's method provides systematic procedures for eliminating the constraints from the dynamic equations, and come out with simpler system equations. Majority of the models that we found in the literature survey were developed using Lagrange's as a method of choice. Thus, to derive the

dynamic equations of the system, the Euler-Lagrange's equations are used. The Lagrange's function is [102]:

$$\mathcal{L} = E_k - E_p \quad (3.103)$$

where

$\mathcal{L}$  = Euler-Lagrange's equation,

$E_k$  = Kinetic energy,

$E_p$  = Potential energy.

Thus, we need to determine the kinetic energy and the potential energy. The total kinetic energy of the system is given by [102]:

$$E_k = E_{kL} + E_{kB} + E_{kP} \quad (3.104)$$

where

$E_{kL}$  = Kinetic energy for flexible link,

$E_{kB}$  = Kinetic energy for mobile base,

$E_{kP}$  = Kinetic energy for the payload.

The kinetic energy of the flexible link  $E_{kL}$  can be found by substituting equation (3.84) into (3.91). The total displacement of a point on the link is  $y = d + w$ , where  $d$  is the displacement of the mobile platform and  $w$  is the deflection at the tip of the manipulator. The speed of total displacement at the tip of the manipulator is  $\dot{y} = \dot{d} + \dot{w}_o$ . After performing mathematical manipulation of the equation (3.93), leads to:

$$E_{kL} = \frac{1}{2} \rho A \int \dot{y}^2 dx = \frac{AL\rho(-8\dot{w}_o(v+\dot{w}_o)+\pi(2v^2+4v\dot{w}_o+3\dot{w}_o^2))}{4\pi} \quad (3.105)$$

The kinetic energy of mobile base  $E_{kB}$  and the kinetic energy of tip payload  $E_{kP}$  are [101]:

$$E_{kB} = \frac{1}{2} m_B v_B^2 + \frac{1}{2} I_B \omega_B^2 \quad (3.106)$$

$$E_{kP} = \frac{1}{2} m_P \dot{w}^2 \quad (3.107)$$

where  $w$  is displacement vector of the flexible link,  $\omega_n$  is the natural frequency of the beam vibration,  $m_B$  is the weight of the mobile base,  $v_B$  the velocity of the mobile base which can be calculated by equation (3.50), while  $I_B$  and  $\omega_B$  are respectively the

moment of inertial and angular velocity of the base.  $m_p$  is the mass of the payload and  $p_p$  is the position vector of the payload. Next, the potential energy of the flexible link caused by the internal bending strain energy of the beam can be found by substituting equations (3.86) into (3.90) leads to

$$E_{pL} = \frac{1}{2}EI \int \left(\frac{d^2w}{dx^2}\right)^2 dx = \frac{1}{64}\pi^4 \left[\frac{EI}{L^3}\right] (w_o)^2 \quad (3.108)$$

Applying equations (3.104) and (3.108) to the Lagrange's equation (3.103) leads to

$$\mathcal{L} = \frac{1}{2}\rho A \int \dot{y}^2 dx + \frac{1}{2}m_B v_B^2 + \frac{1}{2}I_B \omega_B^2 + \frac{1}{2}m_p \dot{w}^2 - \frac{1}{2}EI \int \left(\frac{d^2w}{dx^2}\right)^2 dx \quad (3.109)$$

or

$$\begin{aligned} \mathcal{L} = & \frac{AL\rho(-8\dot{w}_o(v+\dot{w}_o)+\pi(2v^2+4v\dot{w}_o+3\dot{w}_o^2))}{4\pi} + \frac{1}{2}m_B v_B^2 + \frac{1}{2}I_B \omega_B^2 + \frac{1}{2}m_p \dot{\delta}^2 - \\ & \frac{1}{64}\pi^4 \left[\frac{EI}{L^3}\right] (w_o)^2 \end{aligned} \quad (3.110)$$

There are two generalized coordinates, the deflection for the flexible manipulator  $\delta$ , and the linear displacement  $d$  of the mobile platform, respectively. Using Euler-Lagrange equation, we have the equations of motion [102]:

$$\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{w}} - \frac{\partial \mathcal{L}}{\partial w} = 0 \quad (3.111)$$

$$\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{d}} - \frac{\partial \mathcal{L}}{\partial d} = F_B \quad (3.112)$$

where  $F_B$  is the input force to the mobile base. Differentiating the Lagrangian, the following equations are obtained:

$$\frac{\partial \mathcal{L}}{\partial \dot{w}_o} = AL\rho \left( \frac{-8\dot{w}_o - 8(w_o + v) + \pi(6\dot{w}_o + 4v)}{4\pi} \right) + m_p \dot{w} \quad (3.113)$$

$$\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{w}_o} = AL \left( \frac{-16 + 6\pi}{4\pi} \right) \rho \ddot{w}_o + m_p \ddot{w} \quad (3.114)$$

$$\frac{\partial \mathcal{L}}{\partial w_o} = -\frac{EI\pi^4 w_o}{32L^3} \quad (3.115)$$

$$\frac{\partial \mathcal{L}}{\partial v_B} = m_B v_B \quad (3.116)$$

$$\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial v_B} = m_B \dot{v}_B \quad (3.117)$$

Putting together the equation of motion (3.110), (3.111) and (3.112), leads to

$$0 = AL \left( \frac{-16+6\pi}{4\pi} \right) \rho \ddot{w}_o + m \ddot{w} + \frac{EI\pi^4 w_o}{32L^3} \quad (3.118)$$

$$F_B = m_B \dot{v}_B \quad (3.119)$$

The dynamic equations of the mobile flexible link manipulator obtained from the Euler-Lagrange's equations above can be re-written in matrix vector form as follows [80]:

$$M(q)\ddot{q} + R(q) = \tau \quad (3.120)$$

where  $R(q)$  is the repulsive matrix,  $M(q)$  is the resultant forces matrix and  $\tau$  is the input torque matrix.

$$M(q) = \begin{bmatrix} AL \left( \frac{-16+6\pi}{4\pi} \right) \rho + m_P & 0 \\ 0 & m_B \end{bmatrix} \quad (3.121)$$

$$\ddot{q} = \begin{bmatrix} \ddot{w} \\ \dot{v}_B \end{bmatrix} \quad (3.122)$$

$$R(q) = \begin{bmatrix} \frac{EI\pi^4 w_o}{32L^3} \\ 0 \end{bmatrix} \quad (3.123)$$

$$\tau = \begin{bmatrix} 0 \\ F_B \end{bmatrix} \quad (3.124)$$

Where  $m_B$  is the total mass of the MFLM. From equation (3.120), the accelerations of the mobile base and deflection rate of the flexible beam can be obtained as:

$$\ddot{q} = M(q)^{-1}[\tau - R(q)] \quad (3.125)$$

From equation (3.125) we can see that there is zero torque input for the first row of the matrix. This means that the deflection of the beam is constant, which is not the case. The deflection of the beam is effected by the acceleration of the mobile base. Therefore, the acceleration output becomes the input to the dynamics of the flexible beam. The amount of deflection of the beam depends on the elasticity of the beam and the acceleration of the mobile platform. Thus, we compute equation (3.120) for the mobile base and the beam's deflection separately. From equation (3.119), the dynamic for the mobile base gives:

$$m_B \dot{v} = F_B \quad (3.126)$$

$$\dot{v} = \frac{F_B}{m_B} \quad (3.127)$$

From equation (3.118), the dynamic for the flexible beam yields:

$$\left\{ AL \left( \frac{-16+6\pi}{4\pi} \right) \rho + m_p \right\} \ddot{w} + \frac{EI\pi^4 w}{32L^3} = m\dot{v} \quad (3.128)$$

where

$$m = m_p + \rho AL,$$

$m_p$  = mass of the payload,

$\rho AL$  = the mass of the flexible beam.

It is desirable to transform the dynamic equation (3.125) into state-space form. Defining the following state vector

$$X = [q^T \dot{q}^T]^T = [w \ d \ \dot{w} \ \dot{d}]^T \quad (3.129)$$

We have the corresponding linear state-space model,

$$\dot{X} = \begin{bmatrix} 0_{3 \times 2} & I_{3 \times 2} \\ -M(q)^{-1}R(q) & 0_{1 \times 3} \end{bmatrix} X + \begin{bmatrix} 0_{3 \times 1} \\ M(q)^{-1} \end{bmatrix} \tau \quad (3.130)$$

$$Y = [I_{1 \times 2} \ 0_{1 \times 2}] X \quad (3.131)$$

The model of this mobile flexible link manipulator simulated using Matlab through m-file code in Appendix D.

### **3.7 Methodology for identifying the constants for the motor and simulations to test the model**

With the equations of the motor defined in section 3.4.1, this section presents the methodologies for experimental identifying the EMF constant, armature resistance, armature inductance, torque constant and moment of inertia. The methodologies for identifying the viscous friction coefficient, Coulomb friction and static friction are also included. This section fulfils objective point 4 that is set in section 1.2 in chapter 1.

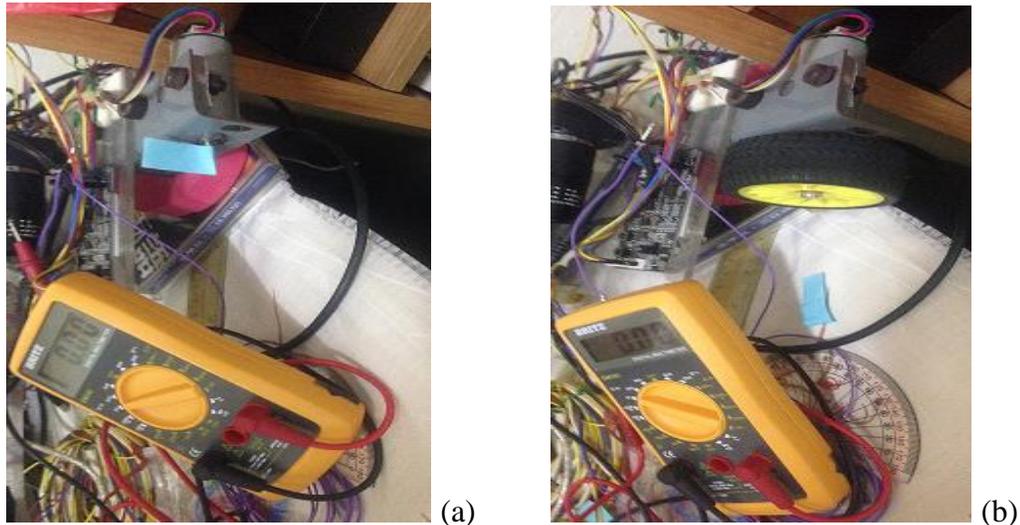
Once the motor constants are obtained, a simulation of the model will then carried out and the result compare with the actual motor output to test the accuracy of the model.

### 3.7.1 Methodology for identifying back EMF constant and armature resistance

In developing the methodology for the identifying the constants for the motor, we examine equation (3.15). At steady state conditions, the  $\frac{dI_a}{dt}$  of equation (3.15) is zero. Thus, at steady state the equation becomes:

$$V = RI_a + k_e \omega_m \quad (3.132)$$

Thus, we need to only measure the steady state values of the input voltage, current and motor shaft velocity.



**Fig. 3.26** Test setup to measure input voltage VS angular velocity of motor shaft.  
(a) motor without wheel (b) motor with wheel mounted at shaft

The test setup is shown in Fig. 3.26, where a digital multimeter set to measure the input voltage to the motor, another multimeter set to measure the current, and the rotational speed determined by the number of revolution of the shaft per and wheel using a digital rpm meter. The tests were carried out with and without wheel mounted at the motor shaft, as shown in Fig. 3.26a and 3.26b. The result for the test is depicted in part I of section 4.4.1 in chapter 4.

### 3.7.2 Methodology for identifying torque constant

In developing the methodology for the identifying the torque constant for the motor, we examine equation (3.16). Equation (3.16) provides the torque constant in terms of torque proportional to current,  $\tau_m = k_t I_a$ . This requires only the torque and current measurements.

To measure the torque, the test set-up in Fig. 3.27 is used. It consists of a wheel mounted on the shaft of geared DC motor. A rope was tight around the rim of the wheel, while the weight tight to other end of the rope. The motor torque was calculated by multiplying the force of the weight with the radial distance from wheel hub to the rope, i.e.  $\tau_m = Fr$ . Where  $F$  is the force from the weight and  $r$  is the radial distance from wheel hub to the rope. The result for the test is depicted in part II of section 4.4.1 in chapter 4.

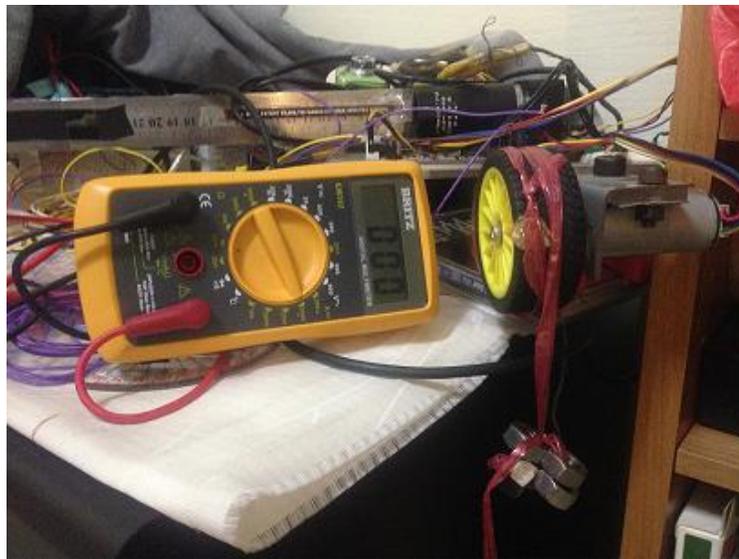


Fig. 3.27 Test setup for current VS torque measurement

### 3.7.3 Methodology for identifying armature inductance and moment of inertia

In developing the methodology for identifying the armature inductance and moment of inertia we examine equation (3.15) and (3.16).

To measure the armature inductance, we measure the rate of current rise with the motor at stall. It is the measure of time it takes for the current to go from zero to steady state. A

current sensor (ACS712, refer to Appendix B.5) connected to measure the real-time changes, by acquiring the voltage outputs from the current sensor using National Instruments NI 9201 module. A fixed voltage was applied to the motor at-rest by locking the rotor/wheel. We acquire the rate of current rise with respect to time. The back-emf is zero since there is rotor does not rotate. Within the first instance of motor start-up the initial current is zero, and the subsequently negligible when compared with rate of change of current. Hence,  $RI_a$  and  $e$  in equation (3.15) is zero resulting in

$$V = L \frac{dI_a}{dt} \quad (3.133)$$

To measure the motor inertia  $J_m$ , we need measure the rate of change of the motor rotor speed from zero to constant speed. According to equation (3.15), the rate of voltage change can be used to calculate the rate of change in rotational speed. The rate of voltage change is measured by connecting the input voltage to National Instruments NI 9201 module. The obtained current and speed change is then used to calculate the moment of inertia  $J_m$  using equation (3.16). The test was carried out by locking the motor rotor until it reaches a set current level to overcoming the initial effects of rotor inductance. Then release the motor rotor.

The test results are presented in part III of section 4.4.1 in chapter 4.

### **3.7.4 Methodology for identifying viscous friction coefficient and Coulomb friction**

In developing the methodology for identifying the viscous friction coefficient and Coulomb friction, we examine equation (3.16).

To determine the viscous friction of the motor, we carry out tests without load. From equation (3.16), at steady state condition,  $\frac{d\omega_m}{dt} = 0$ . At no load condition, it is assumed that the coupling torque is zero, i.e.  $\tau_{coup} = 0$ . Thus, we left with friction torque and input torque. According to Fig. 3.18 (pg. 78), at steady state the coulomb friction is constant, while the viscous friction increases with rotational speed. Hence, tests were carried out without load at varying speeds.

The test results are presented in part IV of section 4.4.1 in chapter 4.

### **3.7.5 Methodology for identifying static friction and stribreak effect**

In developing the methodology for identifying the static friction and stribreak effect, the combination of equation (3.16) and (3.18) are examined.

The static friction was obtained by determining the maximum torque before the motor rotor/wheel starts to rotate. It is by determining the static friction torque at no load condition, and then measure the peak current for the static friction and stribek effect. By using a load of 20 g at a step input voltage of 1.34V we can determine the static friction before the shaft rotates, and then measure the stribek when the current decay. The measurement for the current is by using the current sensor (ACS712, refer to Appendix B.5) and data acquisition device NI 9201 module.

The test results are presented in part V of section 4.4.1 in chapter 4.

### **3.7.6 Testing the motor model**

With the motor parameters being identified in the above section, a series of perturbation simulations were carried out to validate the accuracy in terms of angular speed of the motor shaft based on the model developed for the motor in section 3.4.1. The purpose of the simulation and motor test is to verify the accuracy of the model for:

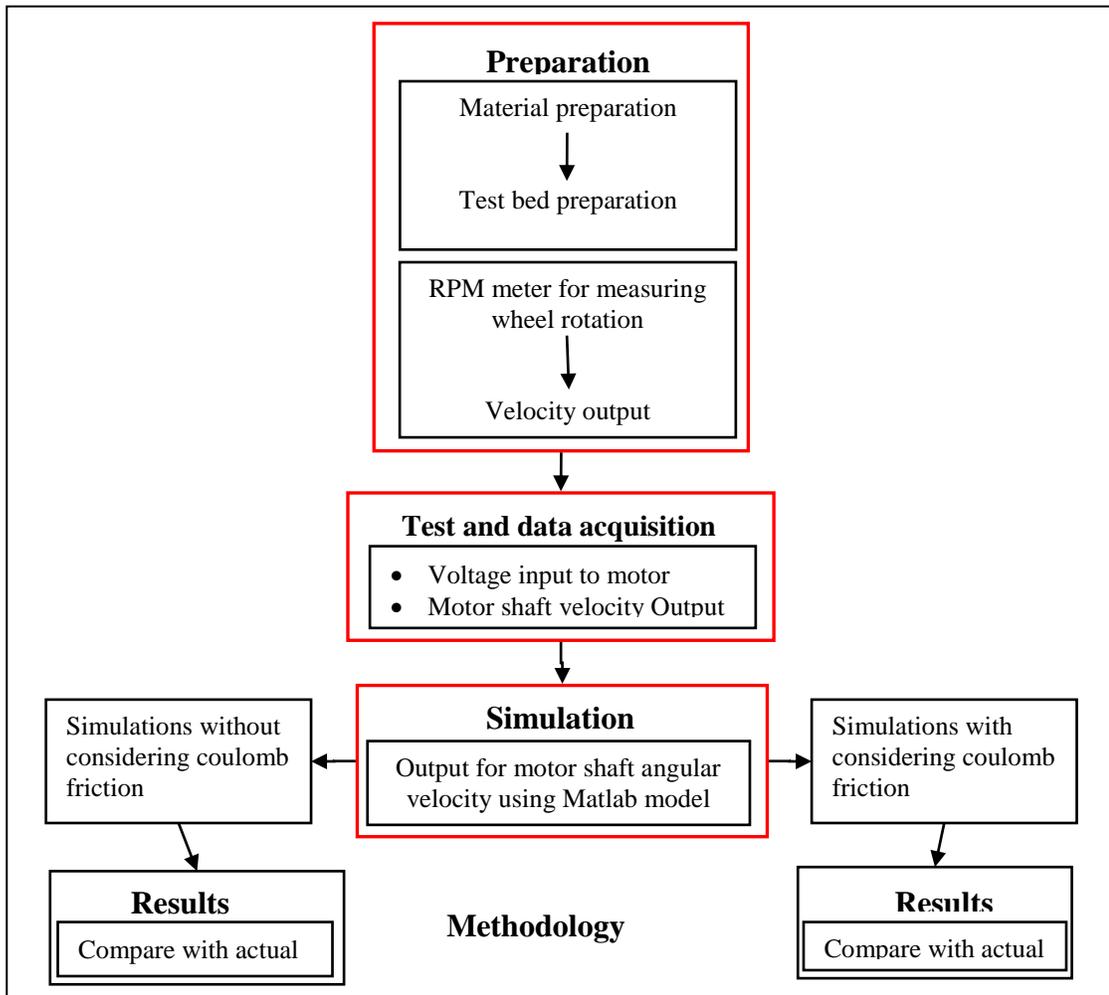
- I. Angular velocity of the motor shaft without considering coulomb friction. The model of equation (3.47) is used for the simulations.
- II. Angular velocity of the motor shaft with the consideration of the coulomb friction. The model of equation (3.48) is used for the simulations.

The phases of the methodology for the tests are as follows:

- I. Preparation: In the preparation phase, the wheel is mounted onto the motor shaft. The setup includes the preparation of the RPM meter for measuring the wheel rotation and the varying voltage input to the motor.
- II. Tests: Perform the tests by setting the various voltage input to the motor. Then measure the speed of rotation at the wheel.
- III. Simulations:
  - a. Perform simulation without considering coulomb using model of equation (3.47).

- b. Perform simulation with considering coulomb using model of equation (3.48).
- c. Compare both results with actual results taken at step II above.

The methodology for testing the model is illustrated in Fig. 3.28 above. Section 4.4.2 illustrates the result of the simulation comparing with actual motor shaft angular velocity.



**Fig. 3.28** Flow chart of methodology for testing the motor's mathematical model

### 3.8 Chapter summary

This chapter started with the mathematical formulation for the respective sensors to be used in experimental tests, and followed by their methodologies for conducting the tests.

Accelerometer, webcam, camera and encoder are the sensors to be tested, thus the methodologies for carrying out the tests were formed. The test results for the respective tests are presented in the next chapter.

In modelling of the mobile flexible link manipulator, a complete and systematic procedure was carried out to develop the model of the MFLM. The MFLM has a flexible beam mounted on a three wheeled mobile platform. The driving wheel is driven by a DC motor that drives the entire robot. The model considered viscous friction and coulomb friction.

The front wheel is connected to the motor via coupling, and the rear wheels are free rolling follower wheels. The electrical and the total torque equations for the motor were formulated. Frictional reactions such as coulomb friction, viscous friction and static frictions are considered in the total torque equation.

Next, the dynamics equation for the front wheel and rear wheels are formulated based on the overall weight and 2-dimensional view of the mobile platform. The equation for acceleration torque of the vehicle is based on the front wheel radius and vehicle acceleration. The acceleration torque of the vehicle and the total wheels torques are then substituted as load torque into the motor model formulated above to calculate the final motor torque. The vehicle acceleration is calculated based on acceleration torque, mass of vehicle and front wheel radius.

The flexible manipulator is modelled based on the Euler-Bernoulli beam theory as a flexible cantilever beam. Then, by using the total energy method, the equation for the beam's vibration frequency is derived, for the end-effector tip with variable mass. Euler-Lagrange's equations are then used to form the equation of motion for the mobile flexible link manipulator based on the vehicle and flexible beam models obtain above.

Methodologies were then formulated to identify the constants and friction coefficients for the motor. Finally, a methodology for testing the derived mathematical model for the motor was formed. The test results are presented in the next chapter.

In conclusion, this chapter has fulfilled objectives points 2, 3 and 4 that are set in section 1.2 in chapter 1.

## 4. Tests results for the sensors and simulations for the mobile flexible link manipulator

With the model of the mobile flexible manipulator developed in the previous chapter, as well as the methodologies developed for the various tests, this chapter delivers the simulations, experimental test and the results. Sections 4.1 to 4.3 present the results for the accelerometer, camera and encoder. Section 4.4 presents the results for determining the constants for the motor and the accuracy of the model.

### 4.1 Test results of testing and conditioning the accelerometer

In section 3.1.2 and 3.1.3 of chapter 3, the methodologies and preparation for testing the accelerometer were set. In this section, we flourish the tests that were being carried out and illustrate the corresponding results.

#### 4.1.1 Static tests and noises attenuation

The test results for the static tests as described in section 3.1.4 are depicted here. The first tests were carried out by placing the accelerometer's x-axis and y-axis horizontal, while the z-axis facing gravity. Fig. 4.1 shows the raw signal from the accelerometer's x-axis. The raw signal coming from the accelerometer is very noisy. It has a high frequencies peak-to-peak noise of  $3.36 \text{ m/s}^2$  at zero acceleration.

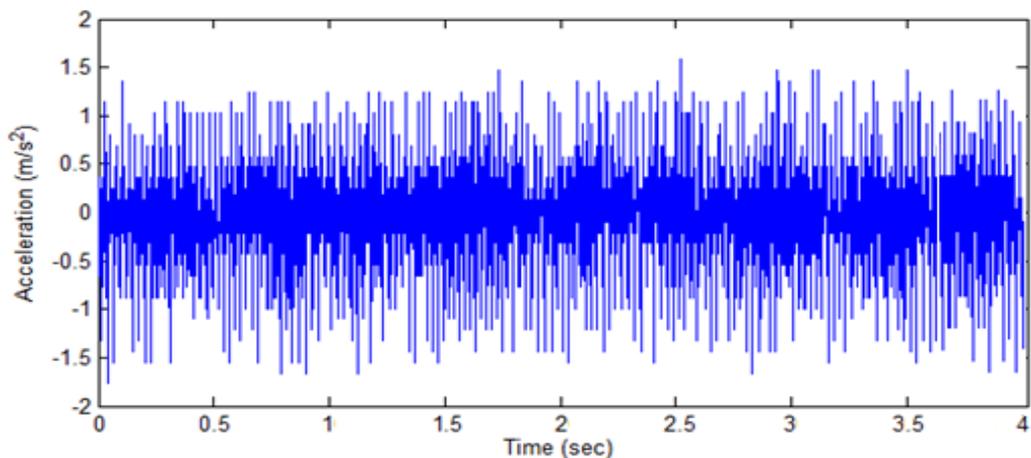
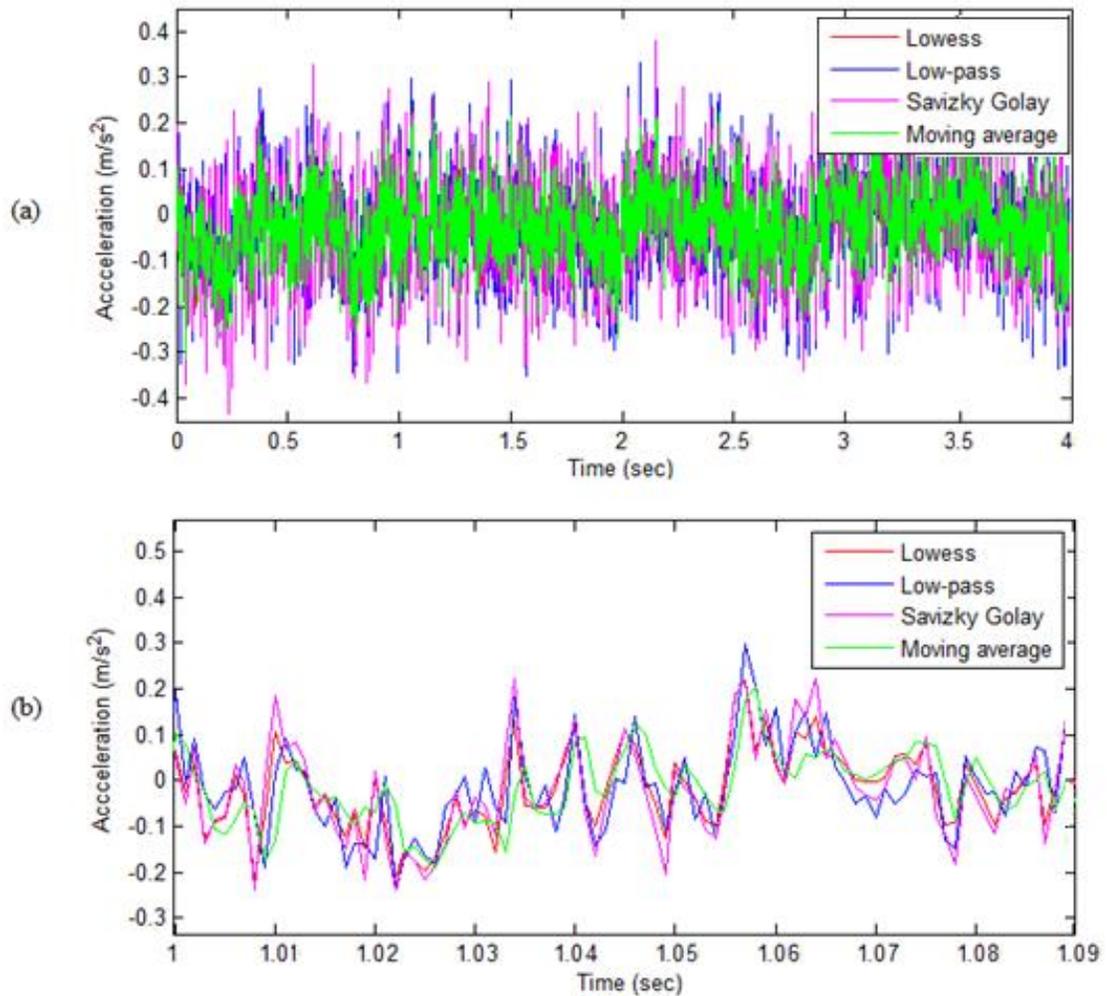


Fig. 4.1 Raw signal with noise

In order to attenuate noisy signals, smoothing filter is proposed. Four types of smoothing filters were compared. They are local regression filters (Lowess), low-pass filters, Savizky Golay filters and moving average filters. Appendix C describes the Lowess, Savizky Golay and moving average filters.

Fig. 4.2a illustrates the comparisons for the four types of filters with plots at 50 data points. Fig. 4.2b depicts the close-up view for 0.1 seconds (1 s and 1.1 s). It can be seen that moving average has the smoothest curve, followed by Lowess and Savizky Golay filers.



**Fig. 4.2** Comparison of filters, (a) filtered data, (b) zoom in 0.1 second between 1 and 1.1 second

The standard deviation, error range and processing time for each of the filters are illustrated on Table 4.1. Lowess is lowest for standard deviation and error ranges, but highest in processing time. Low-pass filter has a lowest processing time.

Due to the fact that all the four filter types does not variances very much in term signal shape and noise reduction, low-pass has been recommended due to its low processing time. In the next section, we would compare the four filters in dynamic tests.

**Table 4.1** Standard deviation and error range for the filters

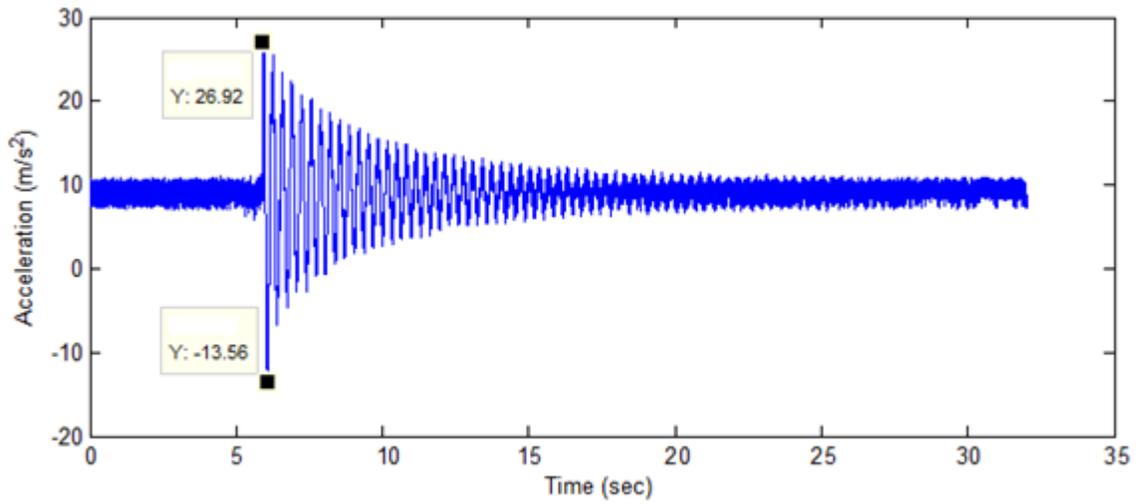
<b>Filters</b>	<b>Standard deviation</b>	<b>Range</b>	<b>Processing time (s)</b>
Lowess	0.086	0.067	0.048
Low-pass	0.104	0.753	0.007
Savizky Golay	0.104	0.822	0.047
Moving average	0.077	0.552	0.031

#### **4.1.2 Vibration tests and selection of curve smoothing filter**

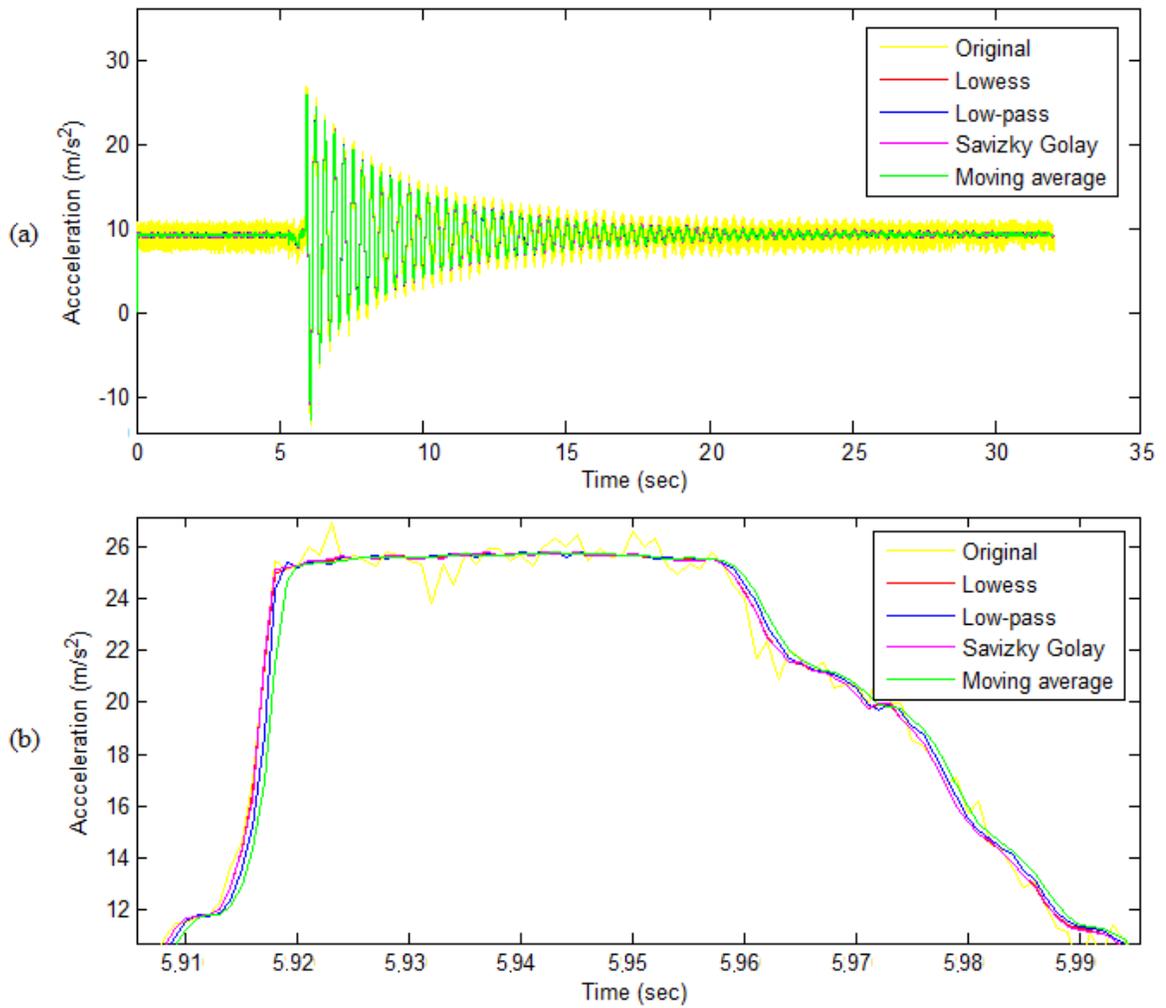
The results for the dynamic tests as described in section 3.1.4 are depicted here. The vibration tests were carried out by placing the accelerometer on the tip of a cantilever beam, and then exciting the tip. The effectiveness of each filter is compared for the data on vibrations. Fig. 4.3 shows the signal for the z-axis of the accelerometer subject to vibrational input. Fig. 4.4a shows the filtered signal for the four types of filters.

Fig. 4.4b depicts the close-up view of the plot for first acceleration cycle. We can see that Savitzky-Golay and Lowess filters are able to preserve the shape at the peak of the ramp. Moving average filters have about 2 ms of lag and low-pass filter has 1 ms lag. Low-pass filter has an overshoot of  $0.28 \text{ m/s}^2$ .

Based on this experiment, all four types of filters have similar output in terms of smoothness and response. Lowess appears to be the best as it preserves the shape of the curve, and has a smoothest curve with no output lag at sharp acceleration ramps. Moving average filter has a slight lag in the output. However, Lowess has the longest processing time compared to other three filters. Low-pass filter has shortest processing. Again, due to the fact that all the four filter types does not variances very much in term signal shape and noise reduction, low-pass has been recommended due to its low processing time.



**Fig. 4.3** Raw acceleration signal for z-axis (parallel to gravity)



**Fig. 4.4** Comparison of filters (a) Filtered data, (b) close up view for first peak of the vibration

### 4.1.3 Results for calibrating accelerometer

The static calibration tests defined in section 3.1.5 carried out on the accelerometer and the results are depicted here. The maximum and minimum voltage values for gravitation acceleration sensing by the accelerometer are obtained for the three orthogonal axes. This is done by placing the X, Y or Z axis towards direction of gravity. Table 4.2 are the results shown in voltage outputs, which shows the constant offset and gain to be used for the accelerometer signals.

**Table 4.2** Maximum and minimum stationary voltage

Axis	X (V)	Y (V)	Z (V)
Maximum (Max)	1.401	1.457	1.702
Minimum (Min)	0.5367	0.5804	0.841
Constant offset (Offset)	0.96885	1.0187	1.2715
Acceleration gain (gain)	0.43215	0.4383	0.4305

By using equations (3.7) and (3.8) of chapter 3, we can obtain the gain for the accelerometer by using the values from Table 4.2.

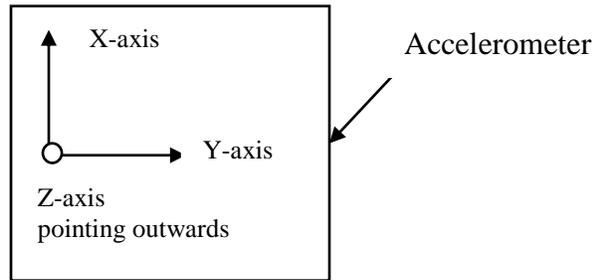
Table 4.3 tabulates the results, each data were taken from the mean value of the raw accelerometer output. It can be seen that there is less than 1 m/s<sup>2</sup> of error for each axis. This error is due to placement error when orienting the accelerometer's respective axes towards the gravitational force, as well as the misalignment error of the orthogonal axes.

**Table 4.3** Calculated accelerations x-, y- and z- axis

Axis	x (m/s <sup>2</sup> )	y (m/s <sup>2</sup> )	z (m/s <sup>2</sup> )
Z-axis face towards gravity	0.04389	0.2131	9.799
Z-axis face opposing gravity	0.4413	0.5122	-9.81
Y-axis face towards gravity	0.1518	9.808	0.2096
Y-axis face opposing gravity	-0.7741	-9.809	-0.0739
X-axis face towards gravity	9.802	0.3027	0.06336
X-axis face opposing gravity	-9.81	0.9549	0.934

#### 4.1.4 Results for dealing with walking bias errors

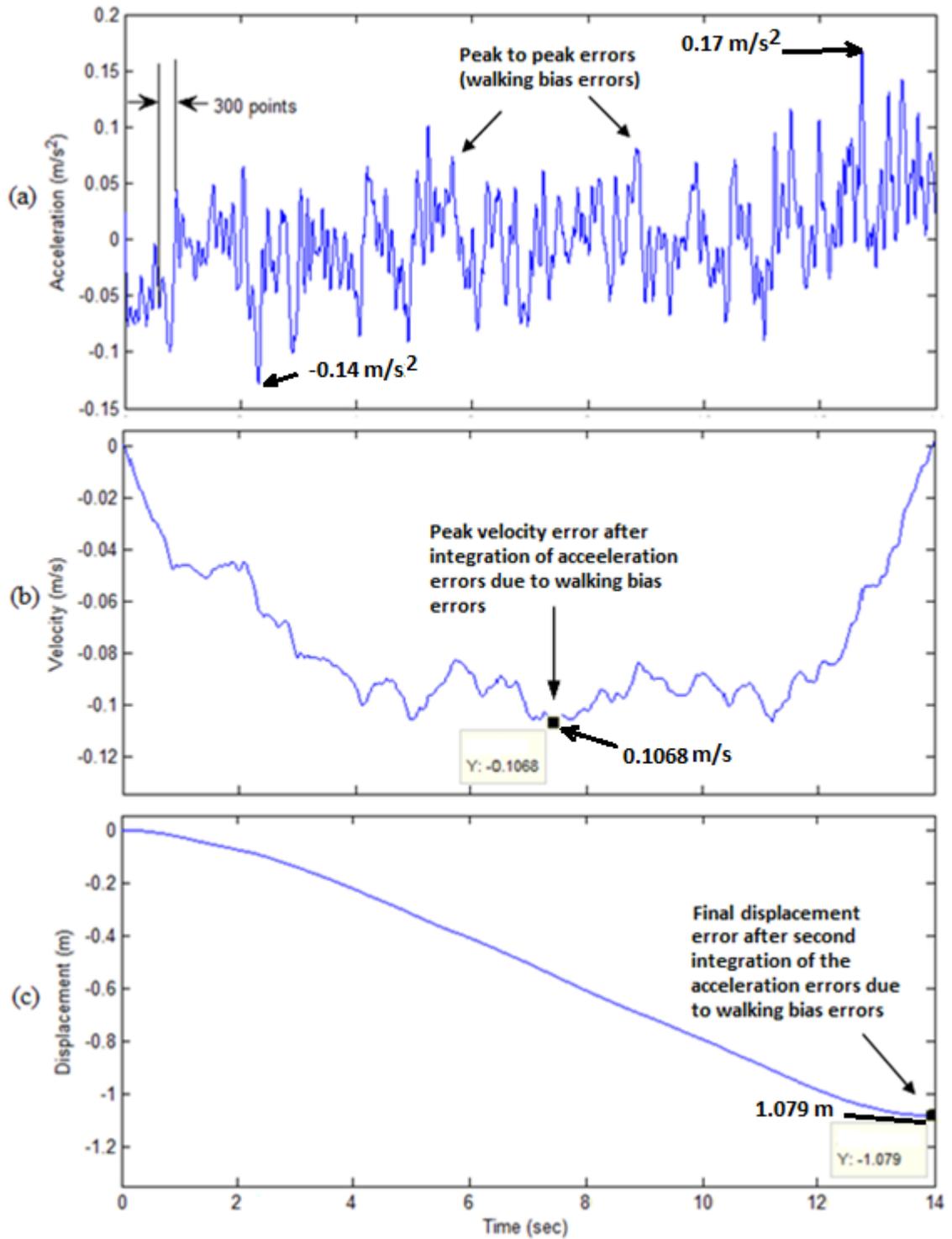
The test as defined in 3.1.6 for the walking bias is presented in this section. The layout of the stationary accelerometer is illustrated in Fig. 4.5 below for carrying out the tests.



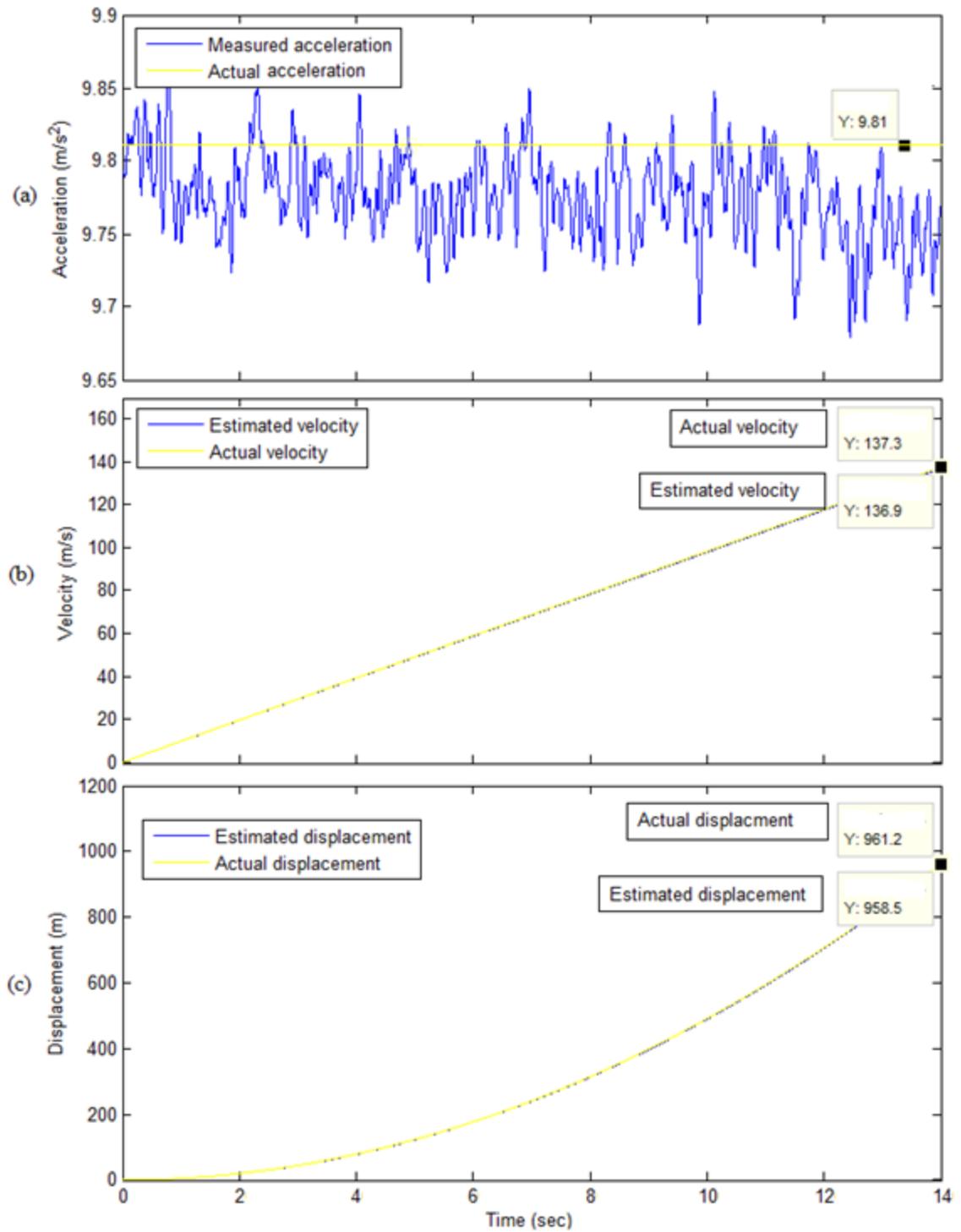
**Fig. 4.5** Layout of the accelerometer to test for its walking bias error

Fig. 4.6 shows an accelerometer stationary at zero acceleration. The acceleration plot exhibits walking bias error, where the acceleration signal drifts by between  $-0.14\text{m/s}^2$  and  $0.17\text{m/s}^2$  (demonstrated in Fig. 4.6a). This resulted in maximum velocity errors of up to  $0.1068\text{ m/s}$  (refer to Fig. 4.6b), which resulted in final displacement error of  $1.079\text{ m}$  (refer to Fig. 4.6c) at the end of 14 seconds. Fig. 4.6a shows that for an arbitrary selection of 300 data points, we can consider it as a constant acceleration if the standard deviation is less than  $0.1\text{ m/s}^2$ .

For the accelerometer facing vertically towards the gravity, it is subjected to a constant acceleration force of  $9.81\text{m/s}^2$  ( $1g$ ). Fig. 4.7 depicts the accelerometer subjected to constant acceleration of  $9.81\text{m/s}^2$ . The effect of walking bias error on this acceleration signal (see Fig. 4.7a) resulted in velocity error of  $0.4\text{m/s}$  (see Fig. 4.7b) with the displacement error of  $2.7\text{m}$  (see Fig. 4.7c) at the end of the 14 seconds trial. Thus, the effect of walking bias errors on the accelerometer on zero acceleration or constant acceleration caused significant amount displacement estimation errors.

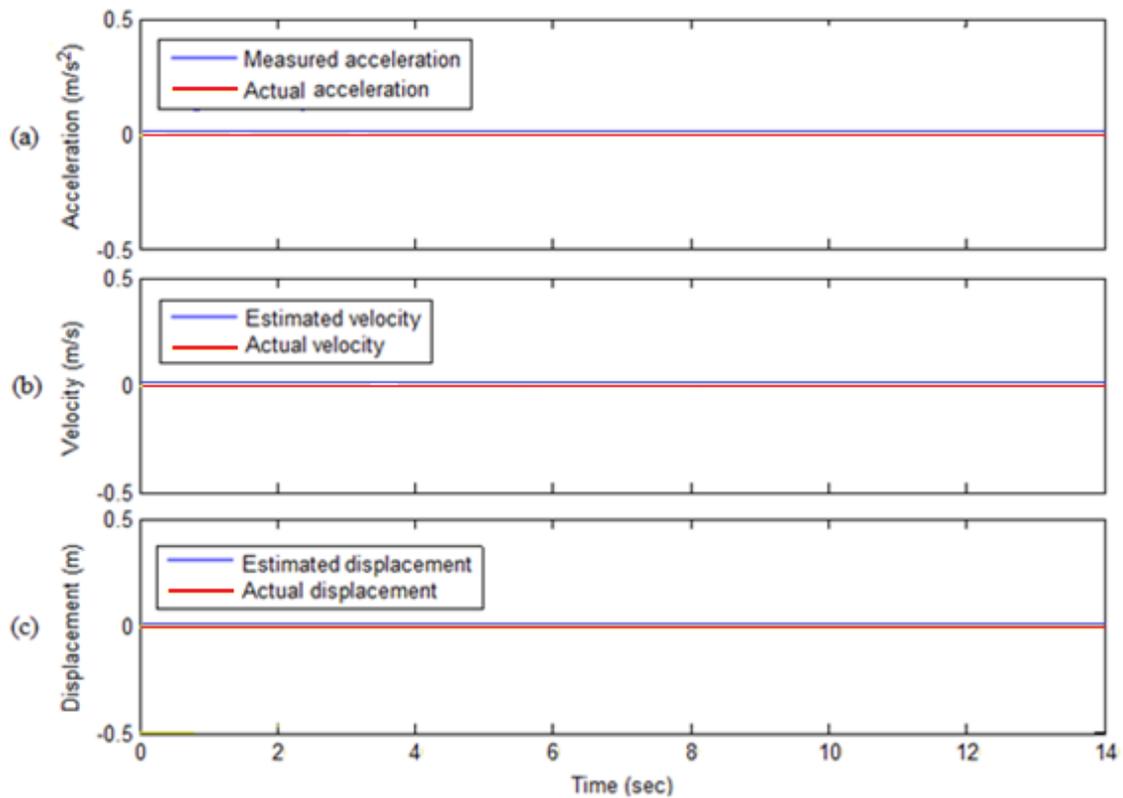


**Fig. 4.6** The effects of walking bias at zero acceleration. (a) Acceleration error, (b) Velocity error, (c) Displacement error.

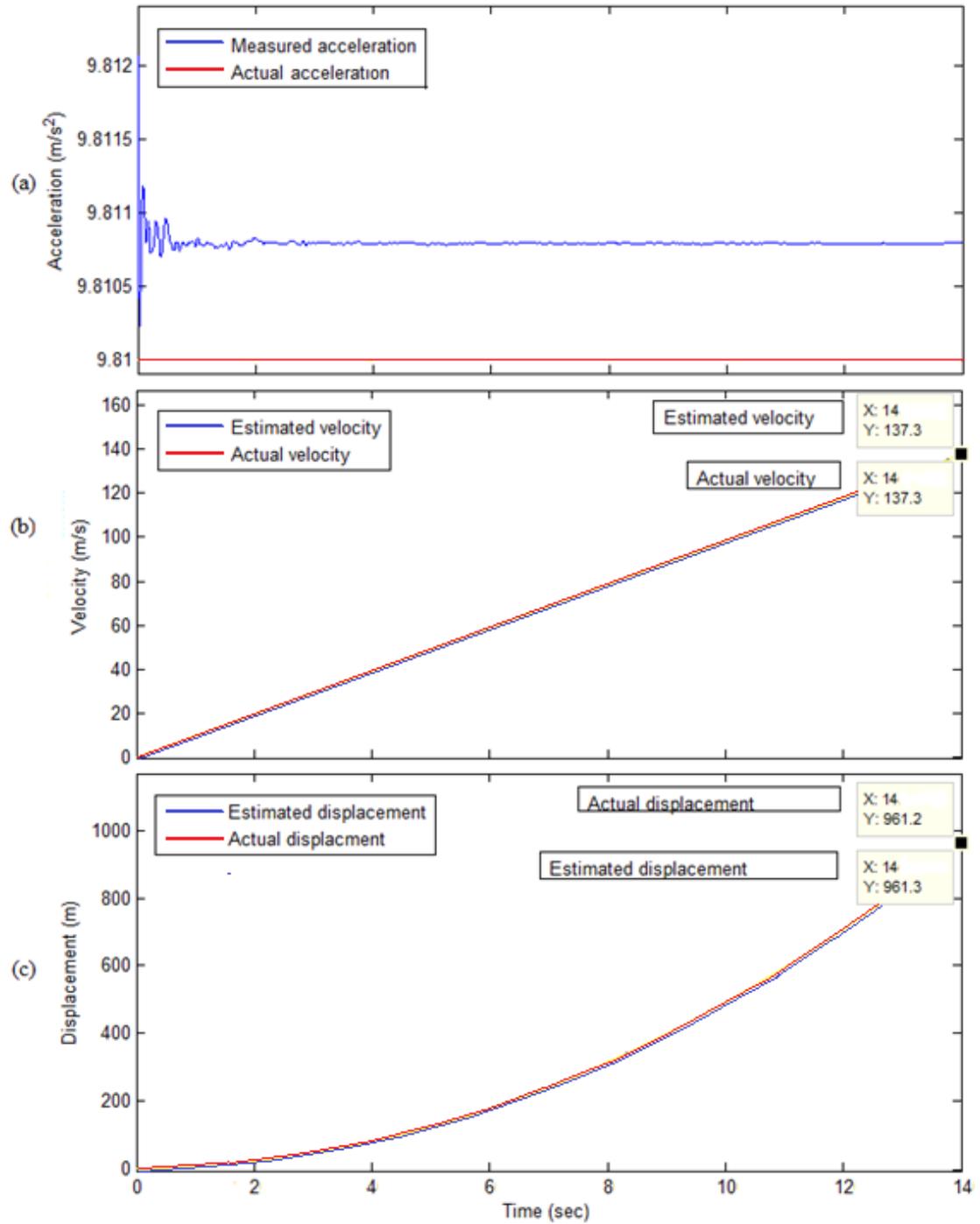


**Fig. 4.7** Effects of walking bias at constant acceleration. (a) Acceleration, (b) Velocity, (c) Position.

Fig. 4.8 illustrates the outcome of the acceleration, velocity and displacement estimations for zero acceleration after implementing the algorithm as shown in Fig. 3.3 in section 3.1.6 that make use of standard deviation to solve the walking bias problem. Compared to Fig. 4.6, it can be seen that all acceleration, velocity and displacement errors are now zero. As for the constant acceleration estimation (refer to Fig. 4.9), comparing it to Fig. 4.7, it can be seen that the velocity error is now approximately zero, and the displacement error is now 0.1 m.



**Fig. 4.8** Results of walking bias being eliminated at zero acceleration. (a) Acceleration, (b) Velocity, (c) Displacement



**Fig. 4.9** Results of walking bias being reduced at constant acceleration. (a) Acceleration, (b) Velocity, (c) displacement

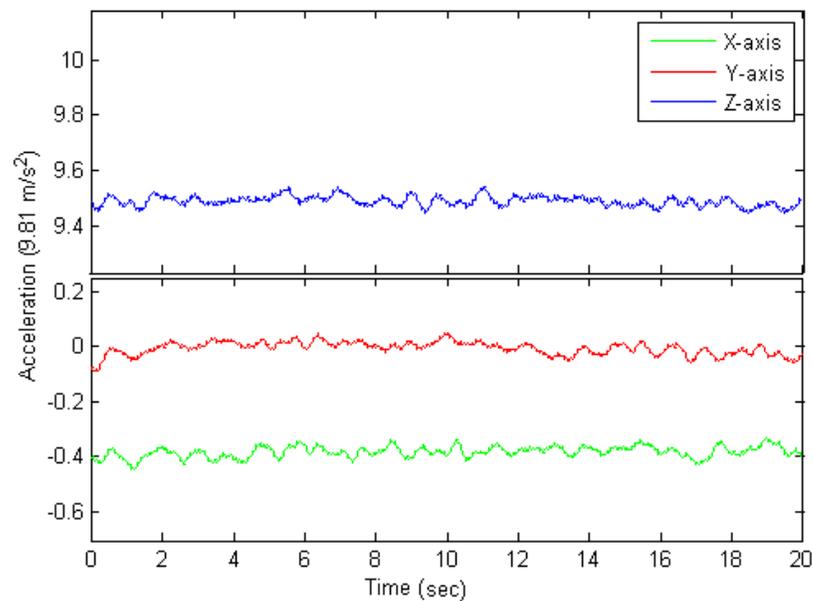
Therefore, there is a 100% improvement for zero acceleration and 90% improvement for a constant acceleration. By applying the algorithm of Fig. 3.3 (in section 3.1.6) before integrating, the walking bias errors are eliminated. In the next section, we will verify the measurement for to dynamic motion.

#### 4.1.5 Experimental results for dynamic tests

The results for the dynamic tests for section 3.1.7 for traversing the accelerometer with various speeds are presented in this section. The experimental test carried out are 5 cm/s, 15 cm/s, 25 cm/s, 50 cm/s, 1 m/s and 2 m/s of robot speeds.

##### i. Results for 5 cm/s speed

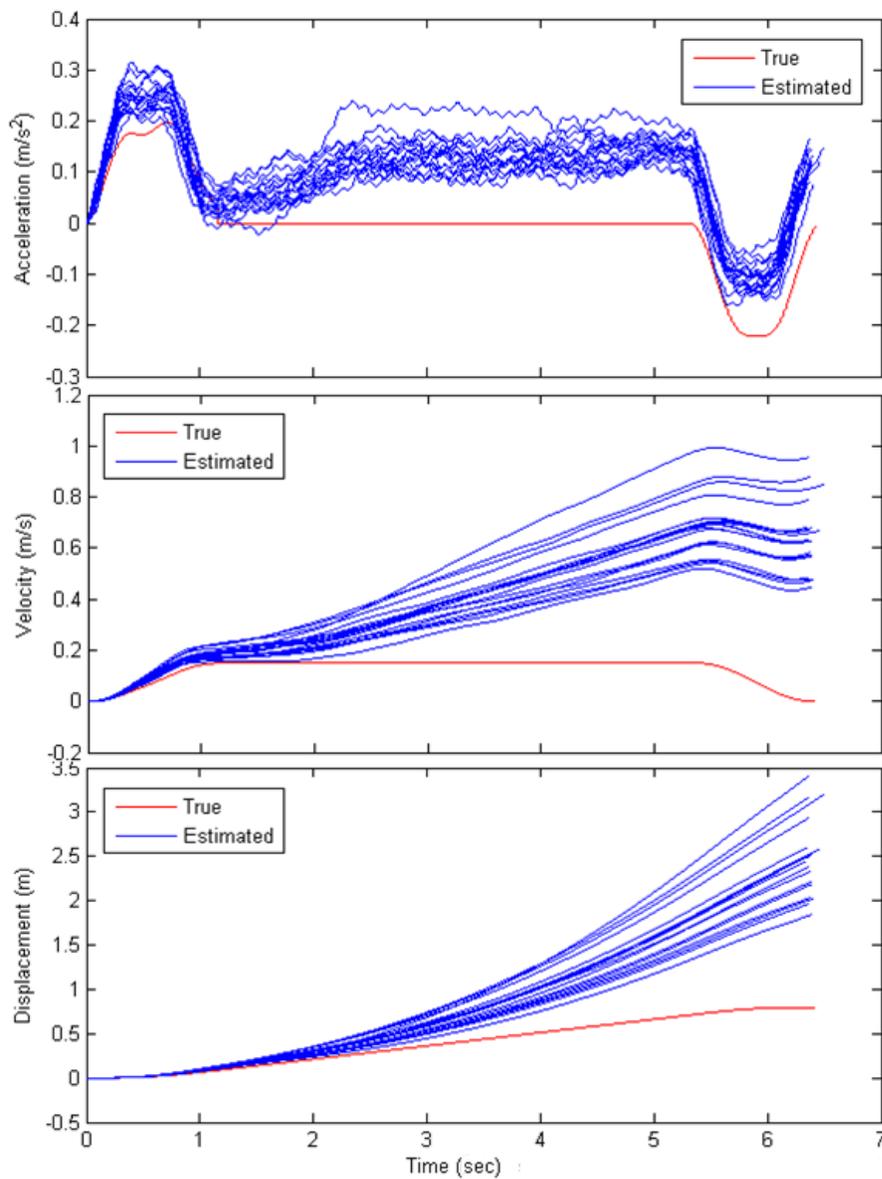
Fig. 4.10 shows the responses for the 5 cm/s robot tip speed. It can be seen that the acceleration is too low to be visible, causing a very low signal-to-noise ratio, thus resulting in imperceptible acceleration signals. The noise amplitude of  $0.114 \text{ m/s}^2$  concealed the low acceleration signals. Thus, the accelerometer is not recommended for sensing in this speed 5 cm/s speed.



**Fig. 4.10** Accelerometer response to 5 cm/s speed

## ii. Results for 15 cm/s speed

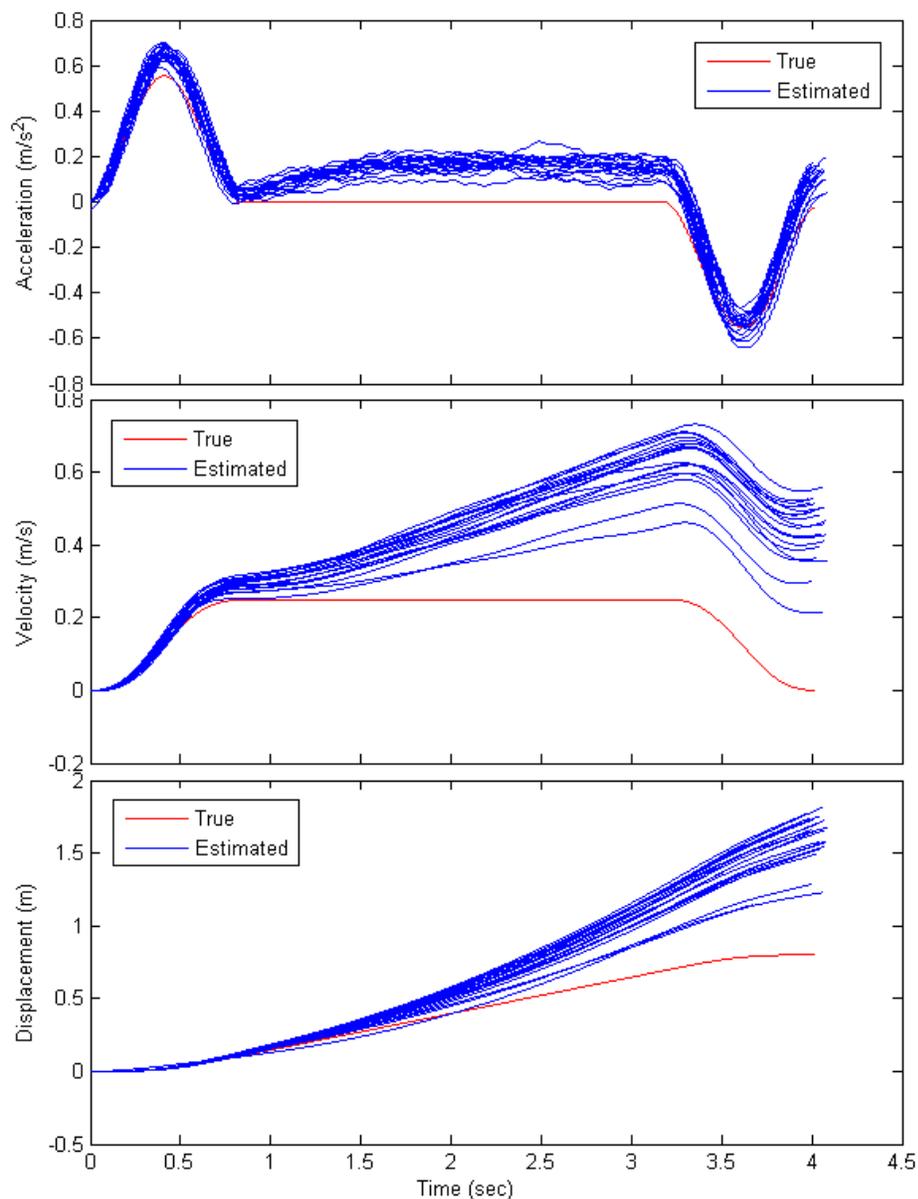
Fig. 4.11 is the response for 15 cm/s robot tip speed. The noise is up to about  $0.2 \text{ m/s}^2$ , while the acceleration is also around  $0.2 \text{ m/s}^2$  to  $0.3 \text{ m/s}^2$ . The signal-to-noise ratio is still low. This contributed to a propagated velocity and displacement estimation errors cumulated at the end of the 0.8 m travel. The displacement estimation error is between 1.014 m and 2.6 m, which is 127% to 325% errors. Thus, it would have high error if the accelerometer is used to measure motion at 15 cm/s travelling speed.



**Fig. 4.11** Accelerometer response to 15 cm/s speed

### iii. Results for 25 cm/s robot speed

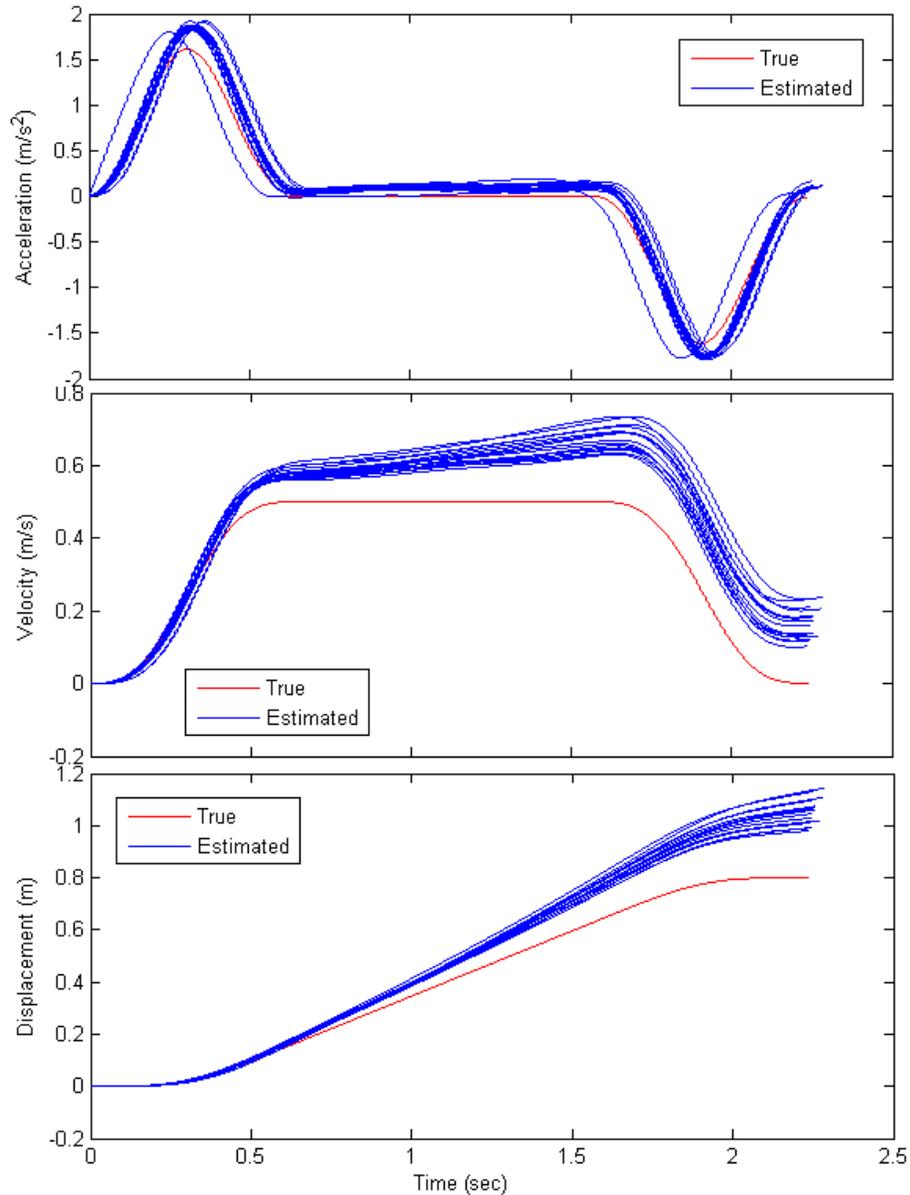
Fig. 4.12 is the response for 25cm/s robot tip speed. It has a higher acceleration signal, thus better signal-to-noise ratio. This contributed to a lesser displacement error of between 0.428m to 1.012m, which is between 53.5% to 126.5% errors. Compared to that of 15cm/s speed, it is 50 % improvement in the displacement estimation. Thus, the accelerometer used for measuring motion at 25 cm/s travelling speed would have acceptable accuracy.



**Fig. 4.12** Accelerometer response to 25 cm/s speed

#### iv. Results for 50 cm/s robot speed

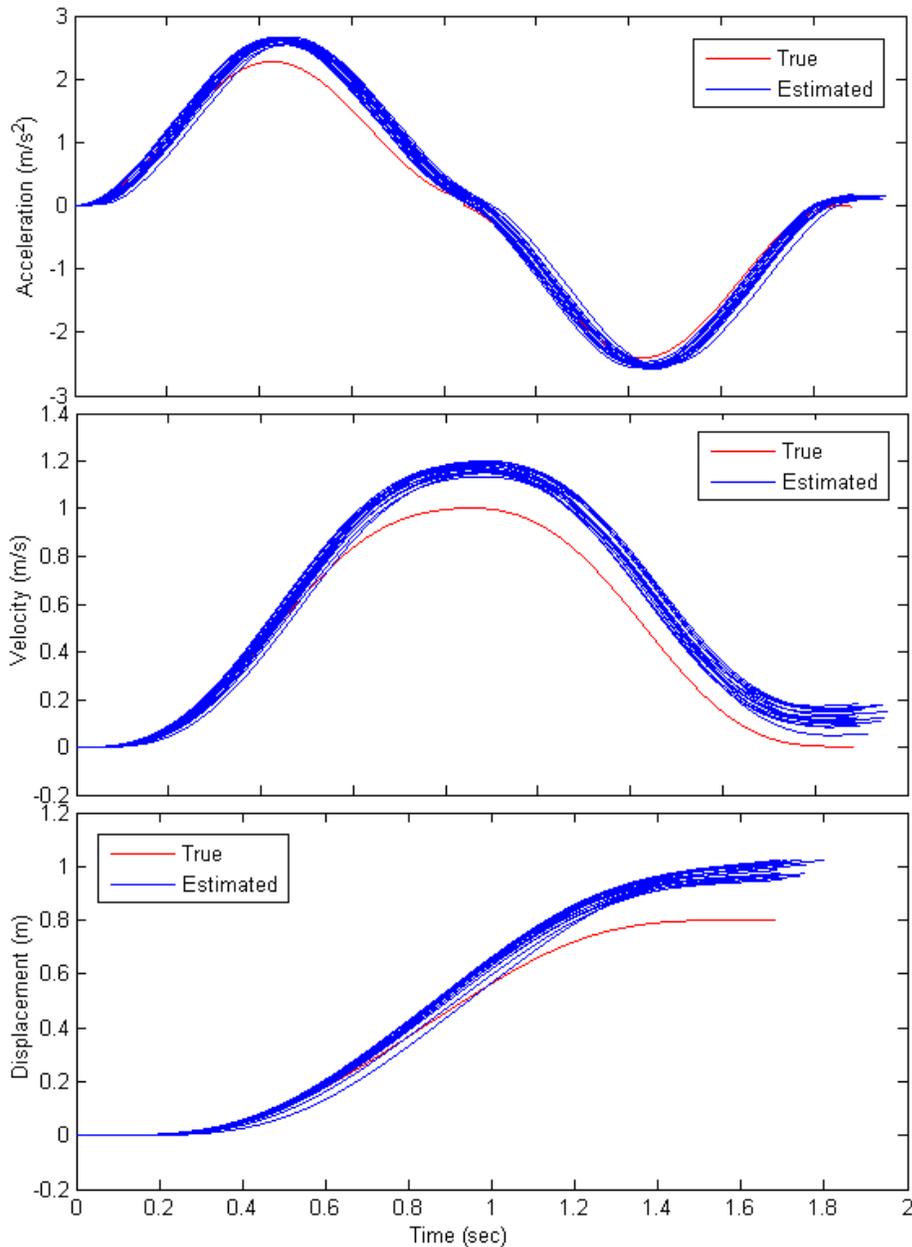
At 50 m/s speed the affect of the  $0.2 \text{ m/s}^2$  acceleration noise is much lesser, resulting in better measurement accuracy (see Fig. 4.13). The displacement estimation error is between 0.1798m to 0.345m, yielded a 22.5% to 43.1% errors. This showed that the accelerometer can achieve a better accuracy at higher speed.



**Fig. 4.13** Accelerometer response to 50 cm/s speed

### v. Results for 1 m/s robot speed

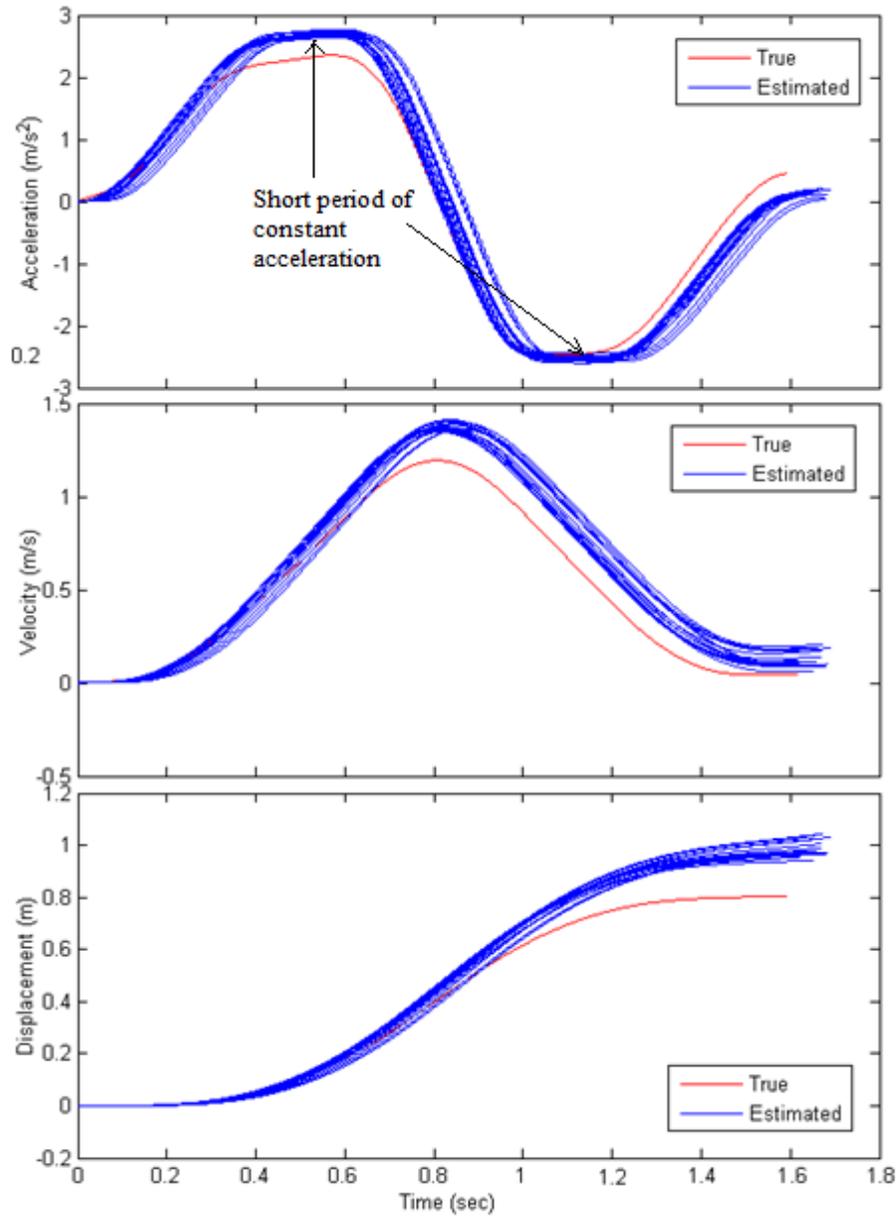
At 1 m/s robot speed, there is no constant acceleration phase, as shown in Fig. 4.14. This yielded a good signal-to-noise ratio, having much better displacement estimation. The displacement error is between 0.1498m to 0.224m, which is only 18.7% to 28% errors. A good accuracy is achieved for the accelerometer to measure the motion at travelling speed of 1 m/s.



**Fig. 4.14** Accelerometer response to 1 m/s speed

### vi. Results for 2 m/s robot speed

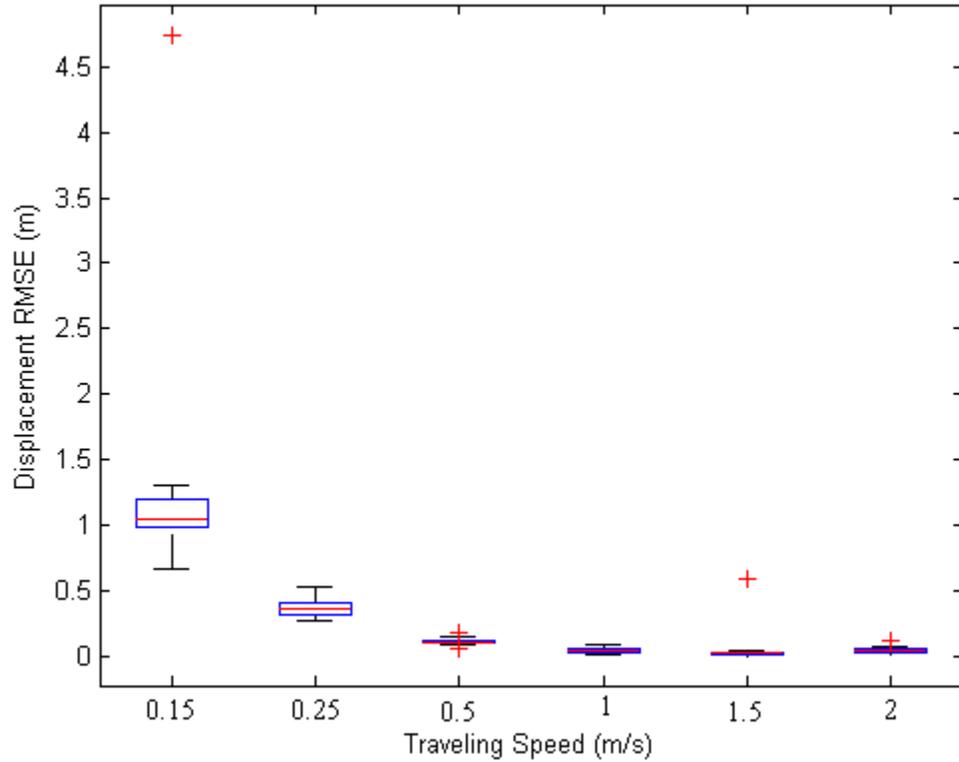
For the robot speed of 2 m/s, there holds a small amount of constant acceleration at the peak acceleration (see Fig. 4.15). But the signal-to-noise ratio remains to be good as compared to that of 1m/s speed. The displacement estimation error is between 0.13 m to 0.24 m, which is between 16.5% to 30% errors. A good accuracy is achieved for the accelerometer to measure the motion at travelling speed of 2 m/s.



**Fig. 4.15** Accelerometer response to 2 m/s speed

#### 4.1.6 Evaluating the results

With the results as shown in section 4.1.5 above, we first statistically analyse the estimation performance of the accelerometer for range of speeds that were being tested.



**Fig. 4.16** Boxplot for displacement estimation RMS errors at various speeds

RMS (root mean square) error is used to compute the error distance measurement for each test. The equation for RMSE is [103]:

$$RMSE = \sqrt{\frac{\sum_{t=1}^n (\hat{d}_n - d_n)^2}{n}} \quad (4.1)$$

Where

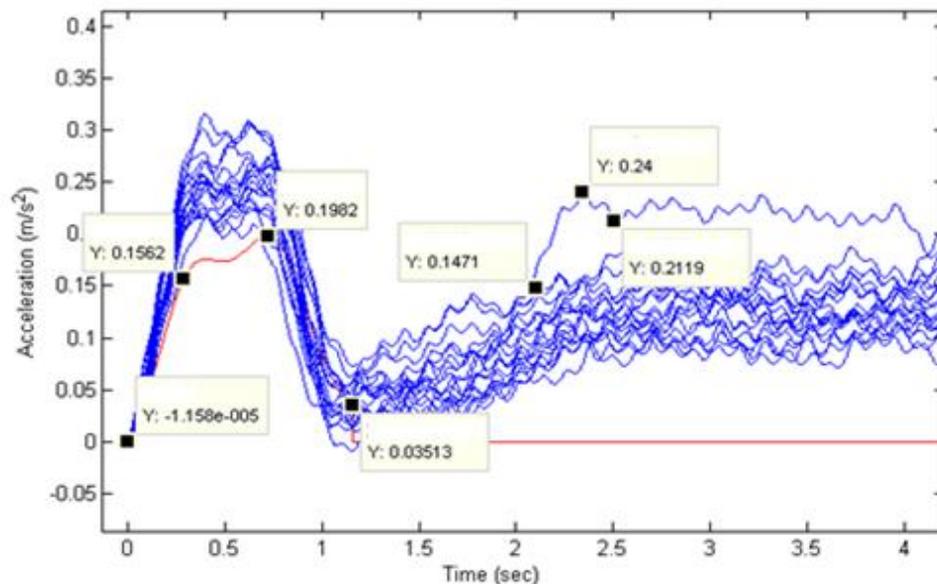
$\hat{d}$  = accelerometer estimated displacement

d = actual displacement.

The statistical data about the error for each speed is represented in Fig. 4.16 above. It can be seen that at higher travelling speed, RMS error is smaller, with the speed of 1 m/s being the best. The error range is also narrower at higher speed, implying that a better

repeatability at higher travelling speed. However, the speed of 2 m/s has error and error range slightly higher than that of 1 m/s. This is believed to be due to the constant acceleration happened at peak of the acceleration (as shown Fig. 4.15). This means that the displacement measurement error is mainly due to low acceleration or constant acceleration.

The analysis deduced that the signal-to-noise ratio is very poor for the speed of less than 15 cm/s. At higher speed, the estimation accuracy becomes better. Fig. 4.17 is the close-up view at the first 4 seconds for 15 cm/s speed. The peak acceleration during ramping is  $0.156 \text{ m/s}^2$  and  $0.198 \text{ m/s}^2$ , while the peak noise is around  $0.147$  to  $0.24 \text{ m/s}^2$ , revealing a poor signal-to-noise ratio of around 1:1. This accelerometer is best used for motion estimation at speed of around 25 cm/s and above and, as illustrated in Fig. 4.12 above.



**Fig. 4.17** Bias errors of about 20 ms for 15 cm/sec robot speed

#### 4.1.7 Cross-axis affects

This section presents the results for the cross-axis tests being set in section 3.1.8 of chapter 3. The results for rotating the accelerometer  $180^\circ$  are first presented, followed by the results for traversing the accelerometer 2 m/s speed.

### I. Rotating the accelerometer 180° to test cross-axis errors.

Fig. 4.18 shows the acceleration responses for the three axes when rotating the accelerometer about the Z-axis. The solid lines are the acceleration values acquired from accelerometer, while the dashed lines are the actual acceleration values. The Z-axis has a maximum of  $2\text{m/s}^2$  cross-axis sensitivity error, while X-axis has indication of hysteresis and deadband errors when X-axis turned towards opposite direction. Double integrating these errors results in amplification of the errors. These errors cannot be eliminated through calibration. Thus, other methods, such as sensor fusion can be used to improve the errors.

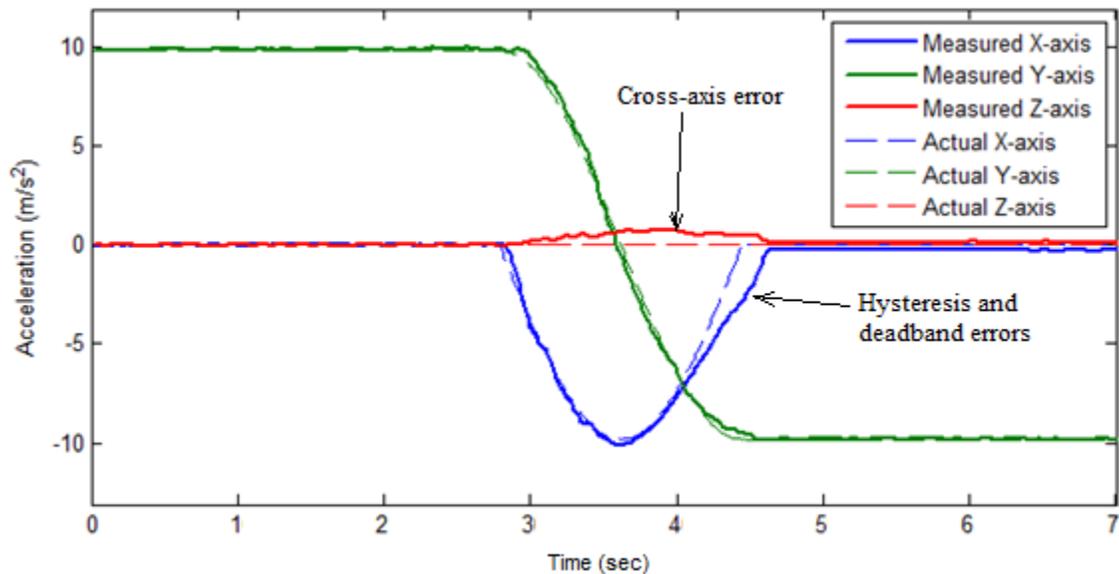
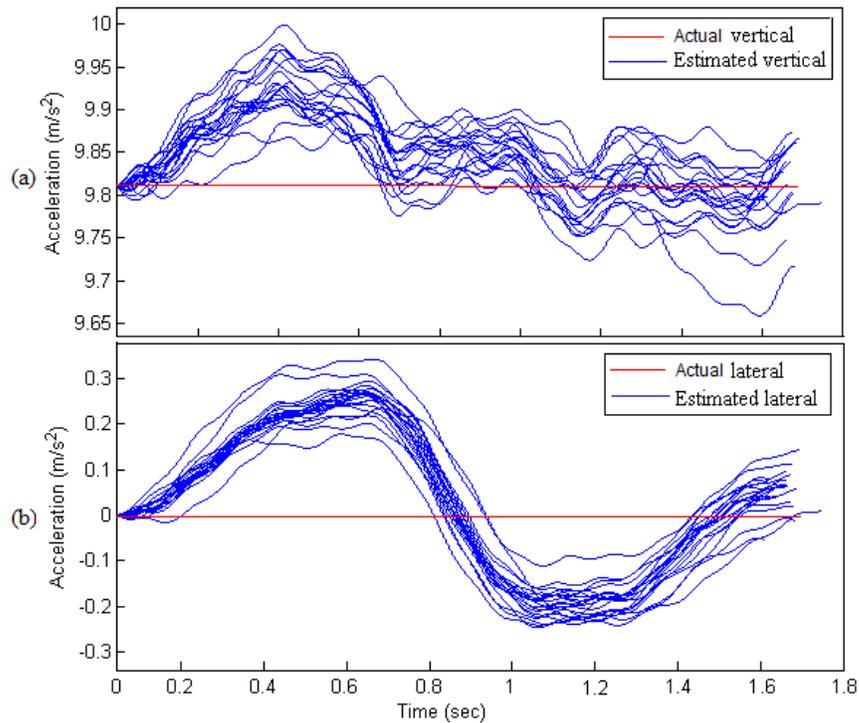


Fig. 4.18 Accelerometer responses when rotated 180° about Z-axis

### II. Traverse the accelerometer at 2 m/s speed to analysis the cross-axis errors.

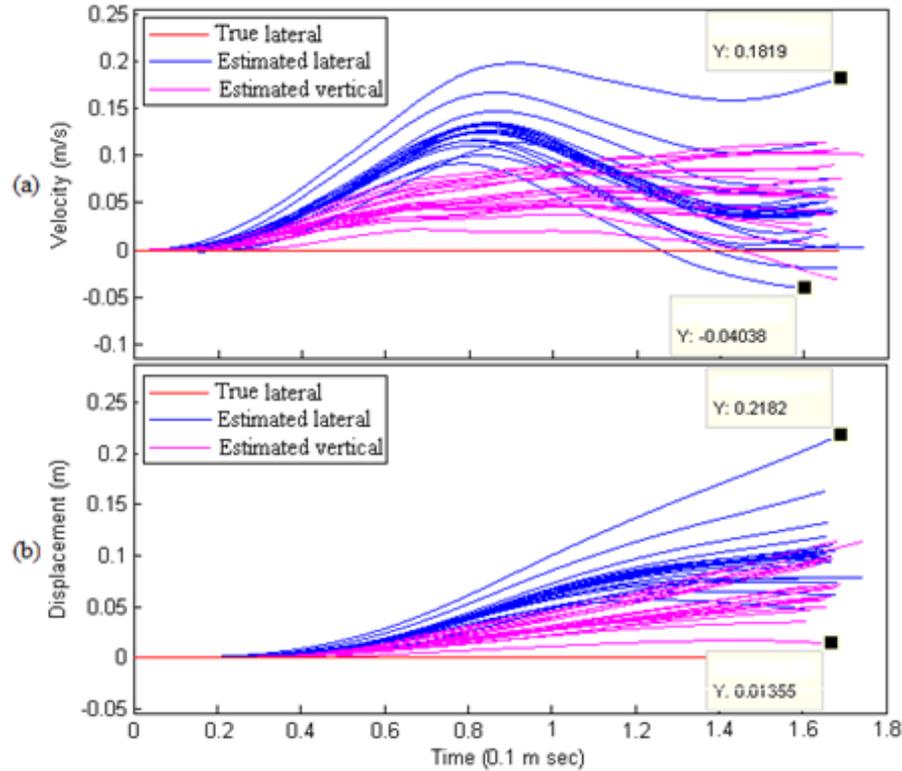
We analyse the errors due to cross-axis sensitivity. Considering the 5 % cross-axis sensitivity (refer to Table 2 in Appendix B.1 on cross-axis sensitivity), running the robot at 2 m/s speed would contribute to maximum error of  $0.15\text{ m/s}^2$  as shown in Fig. 4.19. The response for accelerometer's vertical-axis appears to have around  $\pm 0.14\text{m/s}^2$  (see Fig. 4.19a) cross-axis sensitivity. The accelerometer's lateral-axis has a cross-axis

sensitivity of around  $\pm 0.2 \text{ m/s}^2$ . This contributed to approximately 5% to 8% cross-axis error. However, it is possible that the contribution to this error could also be due to some slight misalignment of the sensor mounting at the tip of the robot arm.



**Fig. 4.19** Cross-axis acceleration sensitivity at 2 m/s speed. (a) Vertical axis, (b) lateral axis

This cross-axis sensitivity resulted in a displacement error of 0.014 m to 0.2 m, as shown in Fig. 4.20. Therefore, we need to take cross-axis error into consideration for position estimation. A solution to minimise this cross-axis problem (to overcome this systematic error) that present in the accelerometer is to replace the 3-axis accelerometer with three single axis accelerometers. However this would increase cost and increased weight for the sensor system. Thus, other methods, such as sensor fusion with camera using Kalman filter model can be used to improve the errors. The accelerometer provides the high frequency measurement at realtime with errors; while camera provides accurate low rate of measurement to correct the errors from the accelerometer at a delay. The fusion of these two sensors can thus improve the measurement accuracy.



**Fig. 4.20** Cross-axis velocity (top) and displacement (bottom) for 2 m/s speed

## 4.2 Tests results for webcam and camera

In this section, the tests as established in section 3.2.2 of Chapter 3 are carried out, and the corresponding test results are illustrated. The test results for webcam are discussed first, followed by the camera. First are the static tests as set in point I of section 3.2.5, followed by the dynamic test as set in point II of section 3.2.5.

### 4.2.1 Tests results for initial displacement measurement using webcam

We present here the static tests set in point I in section 3.2.5. The results for the tests explained in part I in depicted in Fig. 4.21 to 4.23 with the position computing and their corresponding errors for 1 mm displacement sampling, 0.5 cm displacement sampling and 5 cm displacement sampling, respectively.

From the result, it can be seen that maximum error is around 28 mm. The error is due to the effects of: 1. auto-focusing lens which cause changes in focal length, 2. the perspective effects, 3. angle of focus, and 4. poor quality lenses that cause distortion.

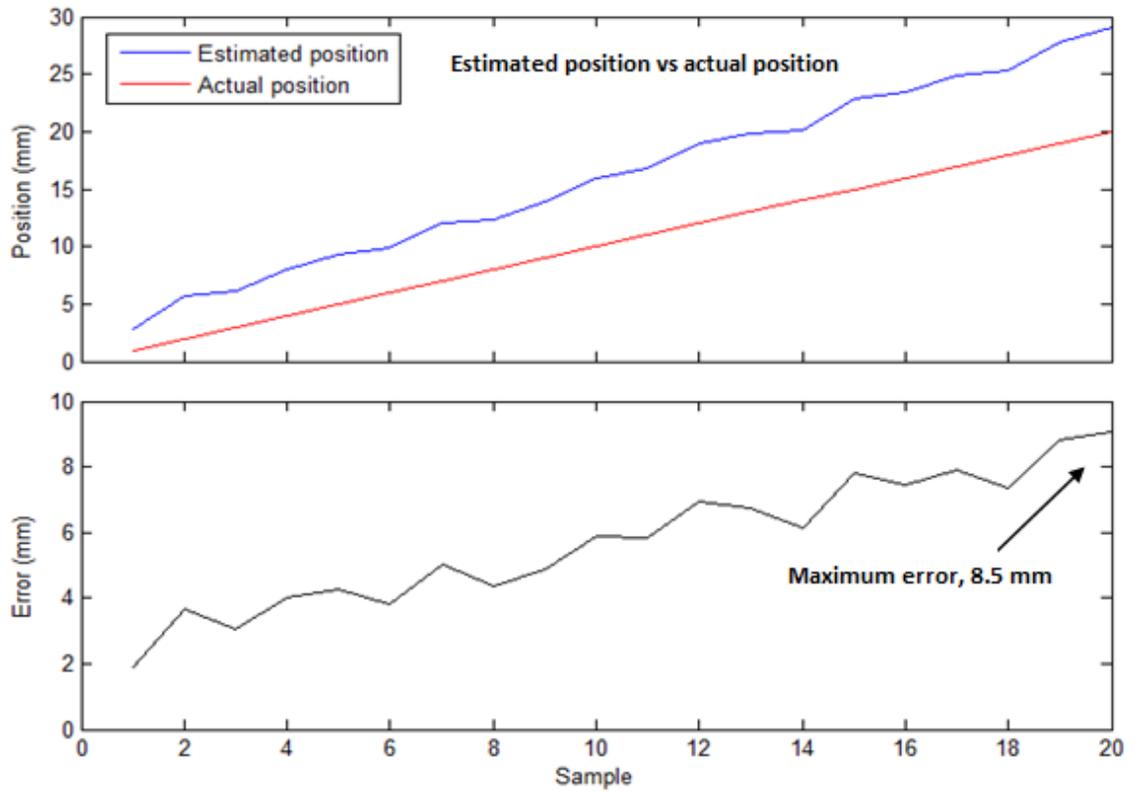


Fig. 4.21 20 samples of position estimation for 1 mm displacement sampling and its error

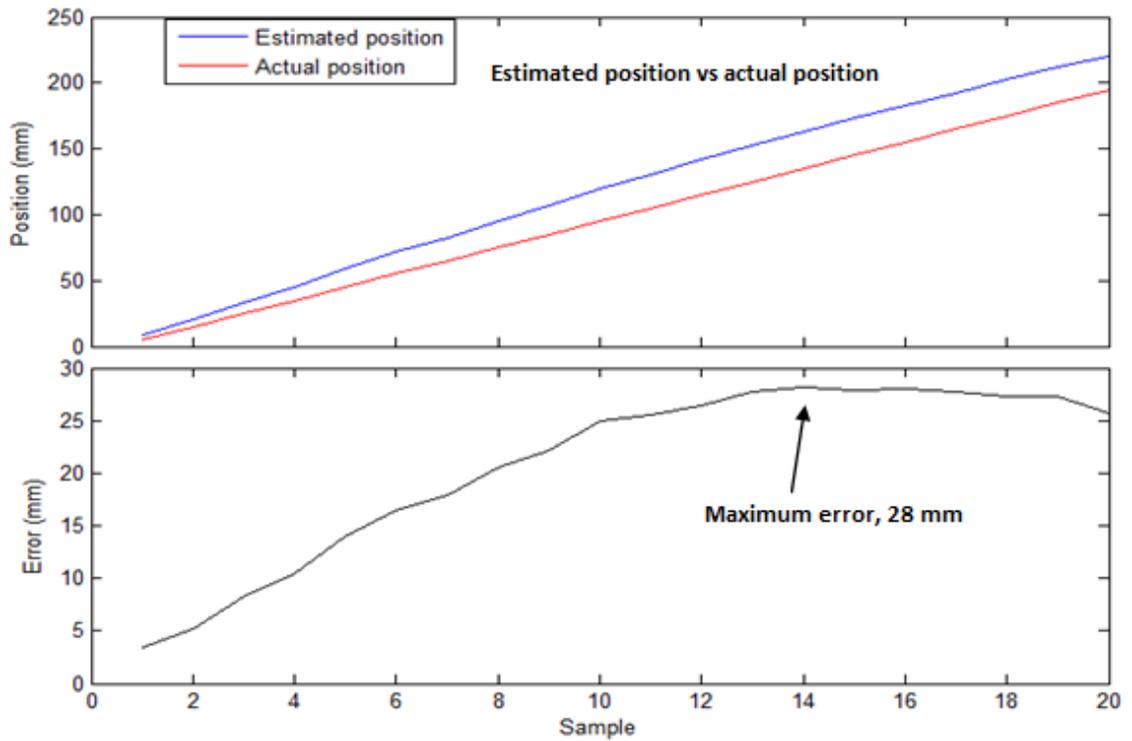
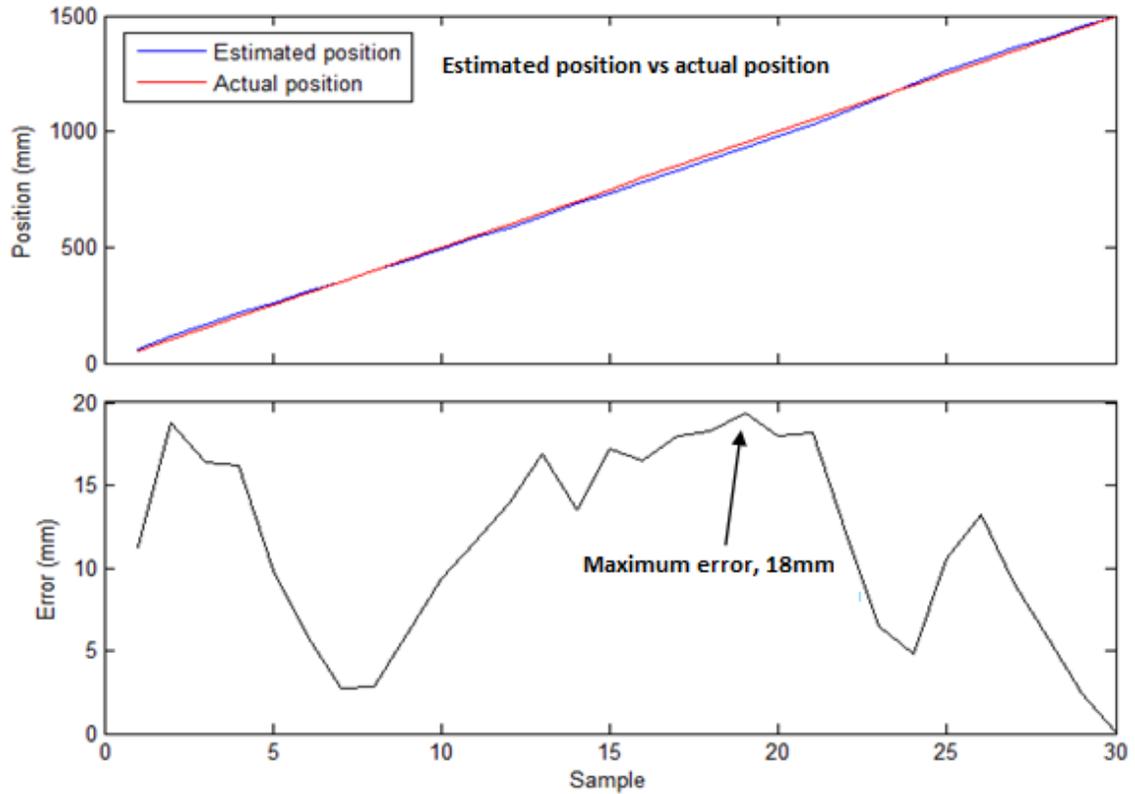


Fig. 4.22 20 samples of position estimation for 5 mm displacement sampling and its error



**Fig. 4.23** 30 samples of position estimation for 50 mm displacement sampling and its error

#### **4.2.2 Test results for displacement measurement that used interpolation method for the webcam**

In order to improve the displacement measurement accuracy of the webcam for the tests carried out in section 4.2.1 above, we present here an interpolation method as proposed in last paragraph of part I in section 3.2.5. Fig. 4.24 to 4.26 illustrates the outcome of the computed position and their corresponding errors for 1 mm displacement sampling, 5 mm displacement sampling and 5 cm displacement sampling, respectively. It can be seen that, as compared to the results in section 4.2.1, there is a 50% improvement in the measurement, which have a maximum error is now around 12 mm. Thus, the static position estimation accuracy using webcam has improved to 12 mm by using the interpolation method.

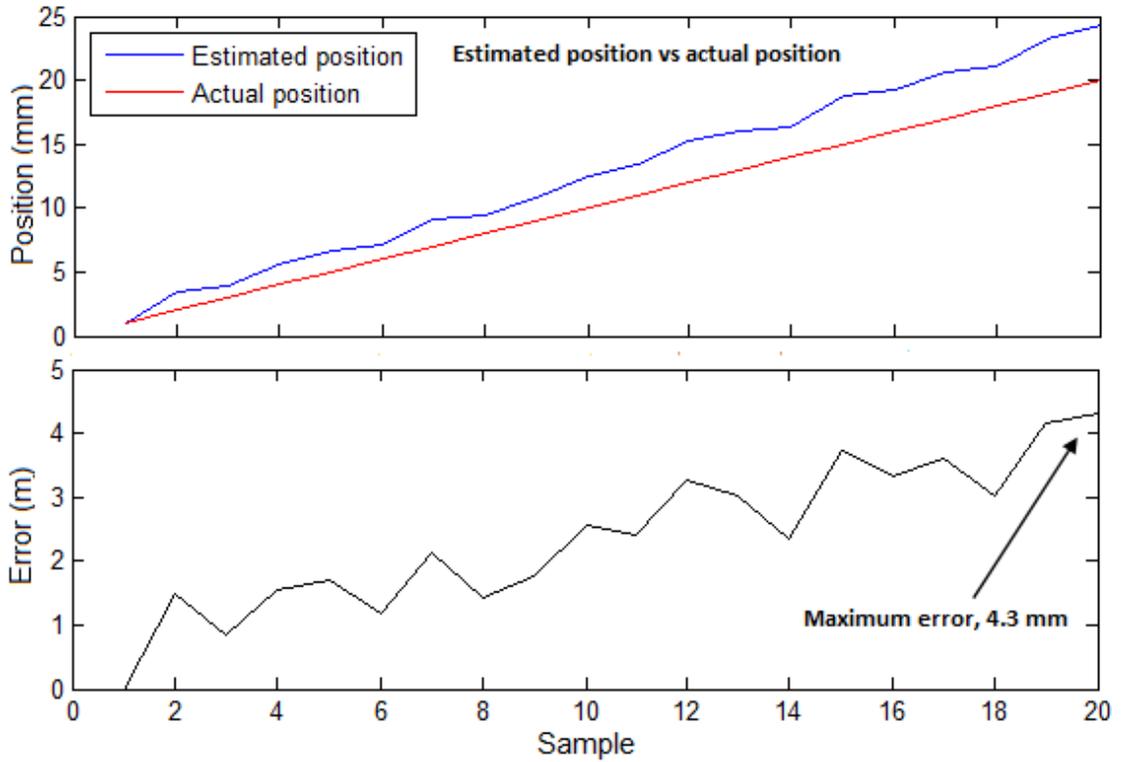


Fig. 4.24 20 samples of position estimation for 1 mm displacement sampling and its error

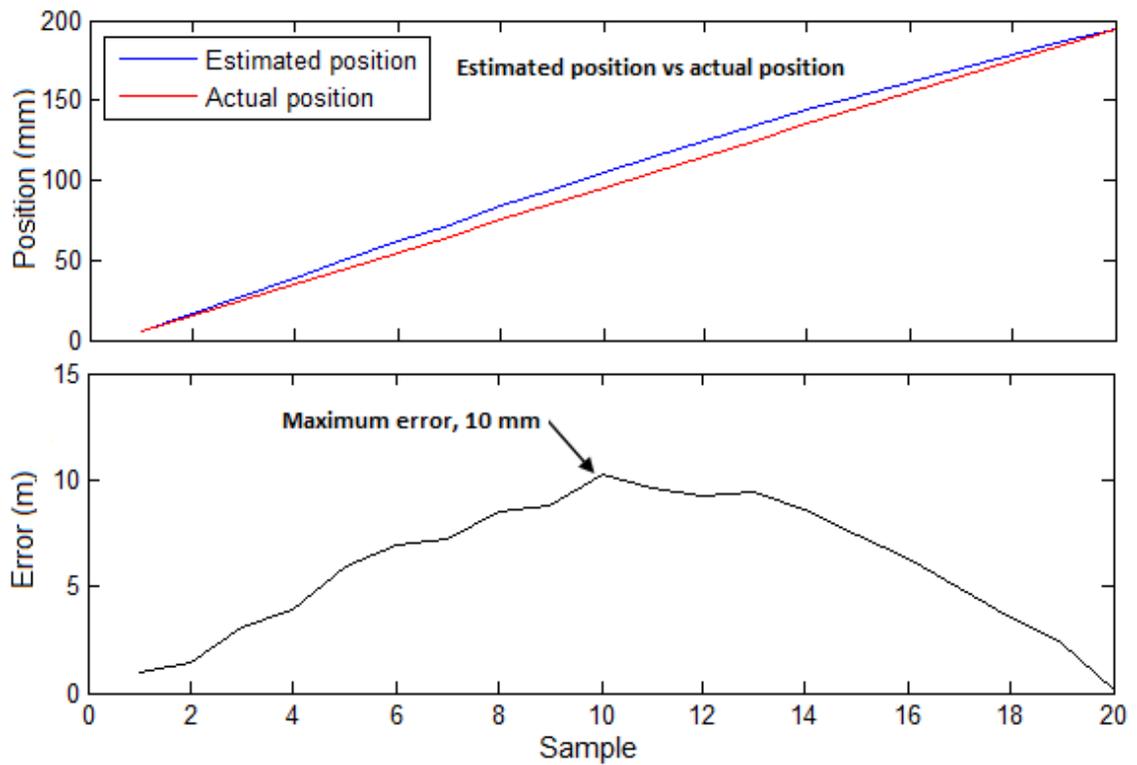
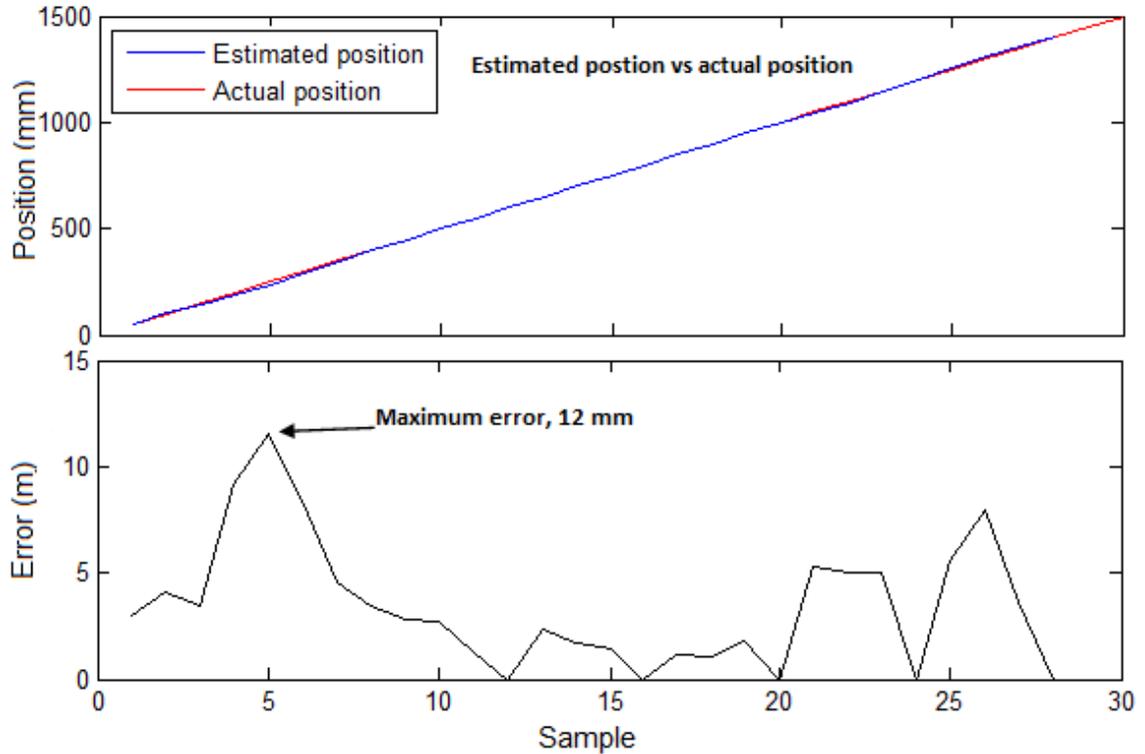


Fig. 4.25 20 samples of position estimation for 5 mm displacement sampling and its error



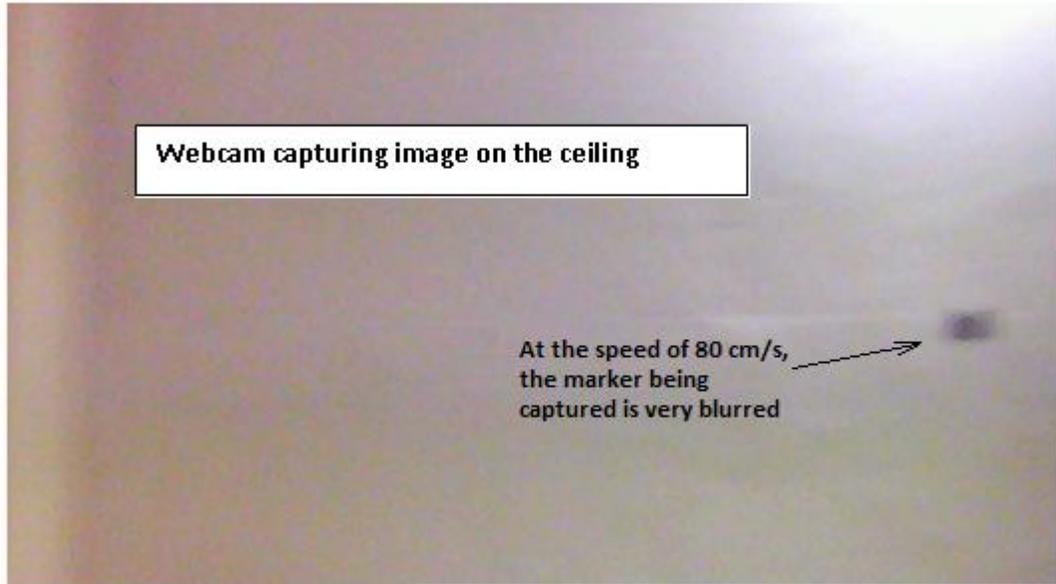
**Fig. 4.26** 30 samples of position estimation for 50 mm displacement sampling and its error

### 4.2.3 Results for webcam dynamic tests for localization of mobile platform

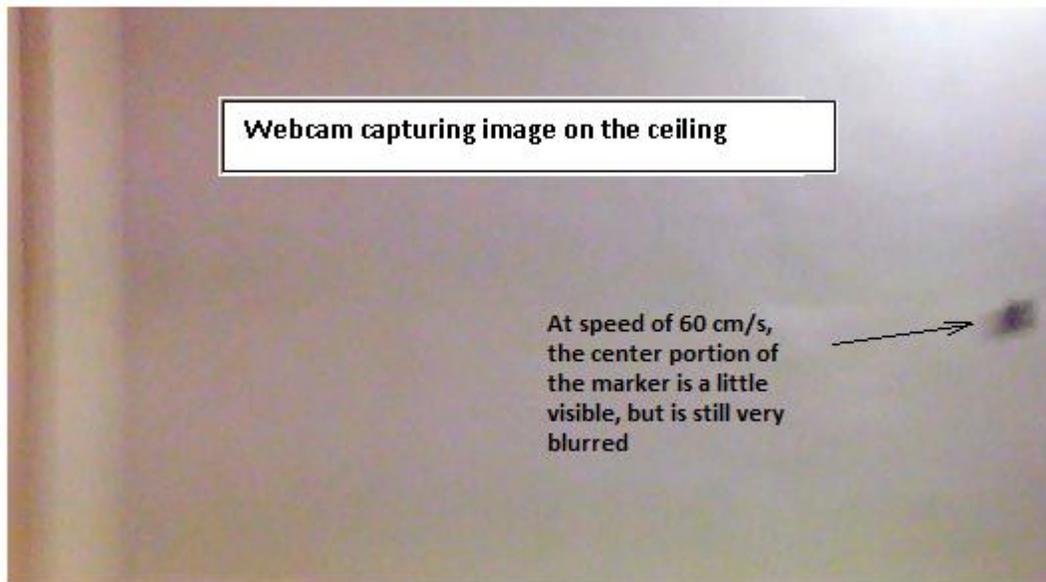
We present here the dynamic tests set in point II in section 3.2.5. The tests as mentioned in part II (in section 3.2.5) were carried out at speeds of 80 cm/s, 60 cm/s, 40 cm/s, 30 cm/s, 20 cm/s and 4 cm/s.

Fig. 4.27 to 4.32 illustrate the images of the marker taken when the mobile platform moves at the set speeds. Fig. 4.33 depicts the average estimation errors of the webcam at various speeds. Due to poor visibility of the images, the marker position estimation for 40 cm/s, 60 cm/s and 80 cm/s speeds were done by manually locating of the marker on the acquired images.

From Fig. 4.27 and 4.28, for the moving speed of between 60 cm/s and 80 cm/s, the marker is hardly unrecognisable. The position estimation errors (refer to Fig. 4.33) were between 2.7 cm and 4.2 cm.

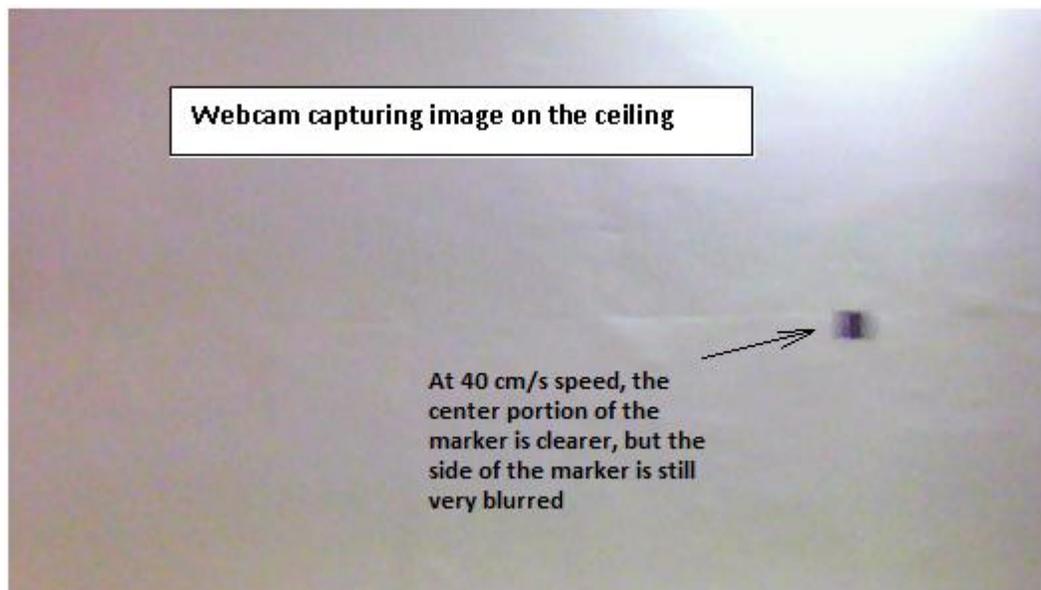


**Fig. 4.27** Platform moving at 80cm/s, the marker image became unrecognisable

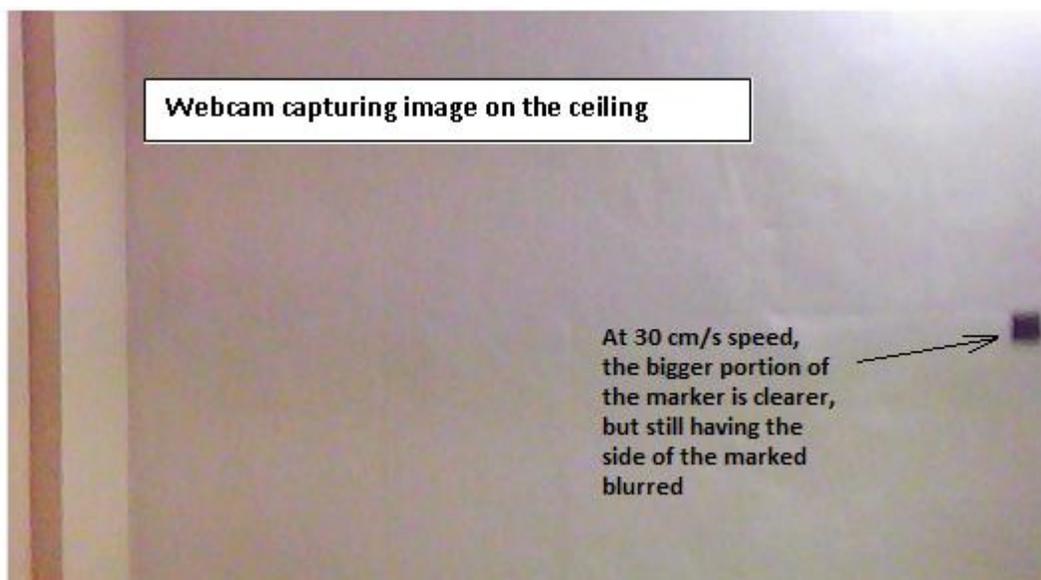


**Fig. 4.28** Platform moving at the speed of 60 cm/s, the marker is blurred

Fig. 4.29 depicts the speed of 40 cm/s. It can be seen that the edges of marker is blurred, but the middle portion of the marker is visible. Referring to Fig. 4.33, the error is 2.2 cm. As for the speed of 30 cm/s and below (see Fig. 4.30 to 4.32), the marker became clearer, and the speed estimation errors were between 0.4 cm/s to 2.3 cm/s. The webcam is thus suitable for position estimation for robot moving at the speed of below 40 cm/s.



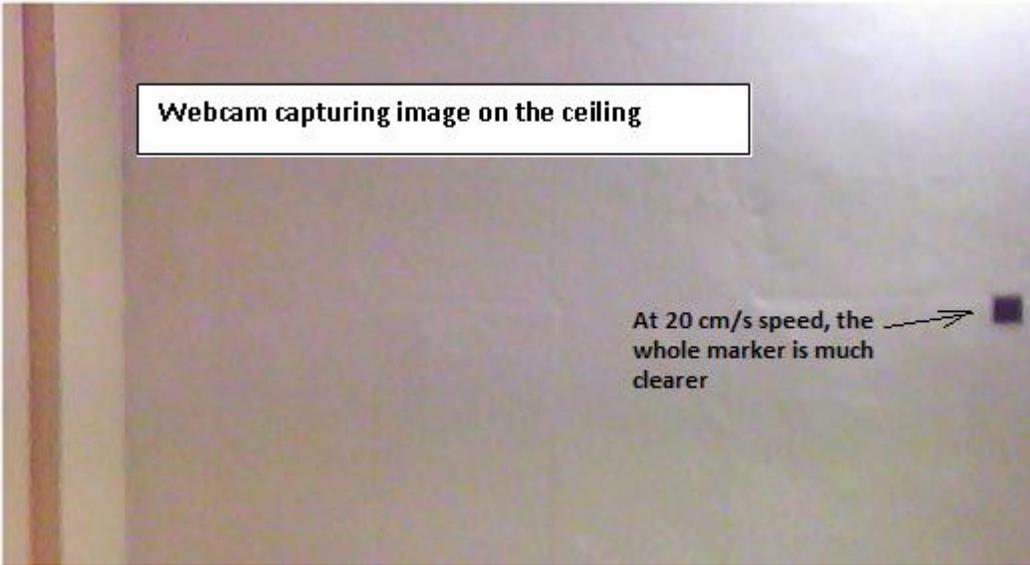
**Fig. 4.29** Platform moving at the speed of 40 cm/s, the marker is slightly blurred



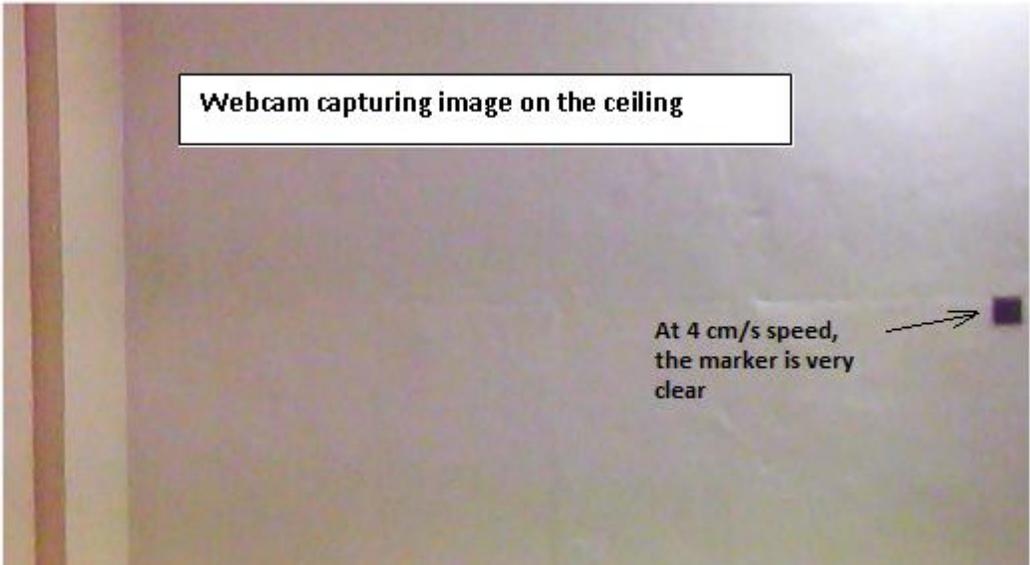
**Fig. 4.30** Platform moving at the speed of 30 cm/s, the marker is more visible

For Fig. 4.30 (30 cm/s speed), we can see that bigger portion of the marker is visible. For 20 cm/s and 4 cm/s (Fig. 4.31 and 4.32 below), it can be seen that the marker is very clear. Therefore, we can say that the webcam is only suitable for capturing images at

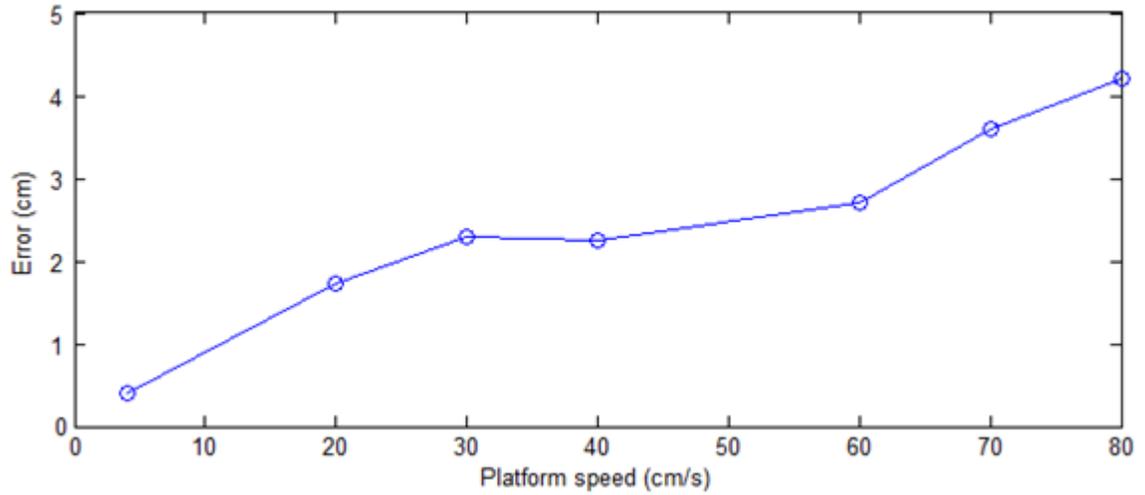
speeds above 30 cm/s, where the marker being captured is visible. It is not recommendable to use webcam for motion estimation at speed of above 30 cm/s.



**Fig. 4.31** Platform moving at the speed of 20 cm/s, the marker is more visible



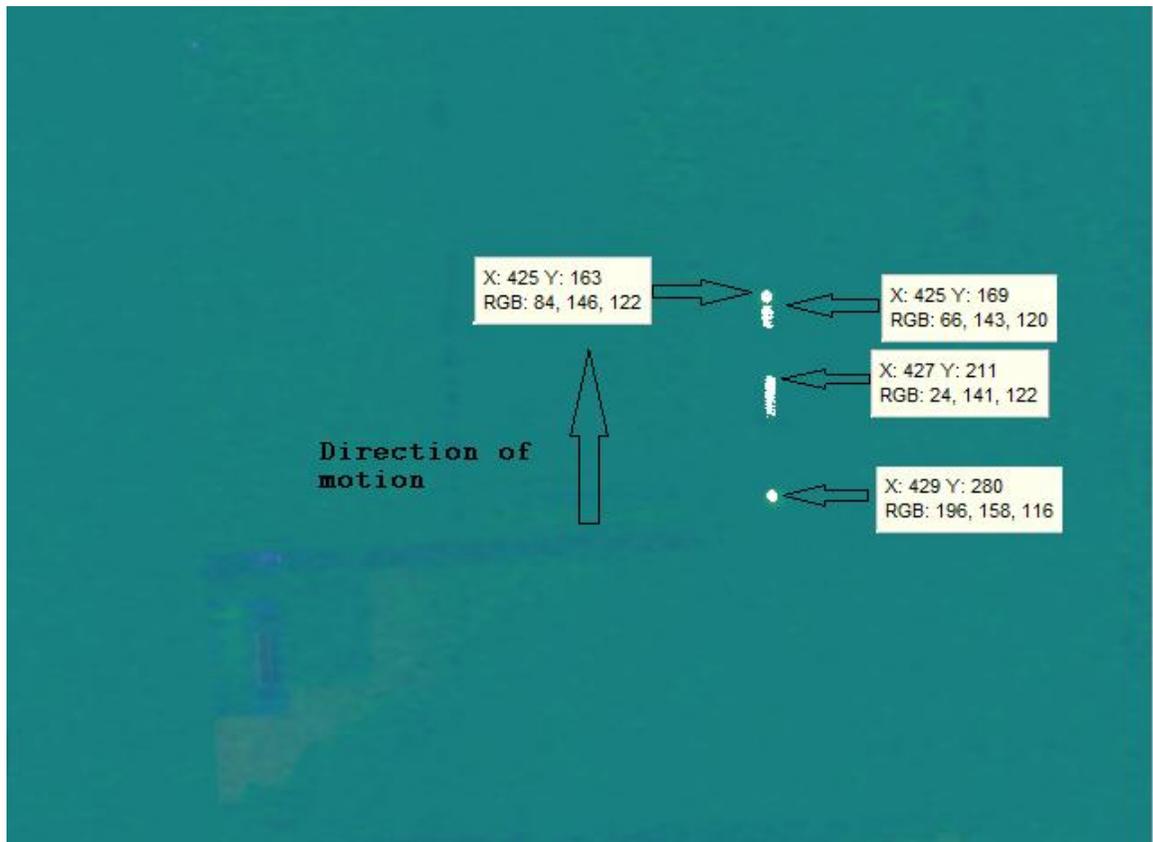
**Fig. 4.32** Platform moving at the speed of 4 cm/s, the marker is very visible



**Fig. 4.33** Position estimation errors for the webcam at set moving speeds

#### 4.2.4 Results for capturing vibration of flexible beam using camera

This section, the vibration test that was set in part I of section 3.2.6 is presented.



**Fig. 4.34** Images of the LED light at the tip of the beam captured by the camera

Fig. 4.34 above illustrates the image capturing the LED light by the high speed camera. It can be seen that when the LED is at the bottom end, the full image of the LED image can be seen. When the tip is at the mid way the LED image is distorted, until it reached the top end where the full image of the LED image can be seen. This revealed that, during when the tip of the beam swing from end to end, the accurate position of the tip is most visible only at the peak ends. Nevertheless, this test draws inference that the camera can be used for vibration estimation of flexible beam.

#### 4.2.5 Results for vibration tracking of flexible beam using camera

We present here the vibration test that was set in part II of section 3.2.6. The image captured has a delay and has low data rate. The exposure time is set to the reciprocal value of the frame-rate. The camera achieves a framerate of around 60 fps, hence the exposure time is about 16 ms. An image is captured somewhere along these 16 ms, which means an accuracy of 16 ms. A further delay is for image processing. In addition, the exposure time of 16 ms caused blurred image to be captured due to high displacement speed of the vibrating beam. Therefore, using the visual estimation with the camera alone inference the time uncertainties that comprised of +/- 7 ms including the uncertainties due to blurred image. A direct cross-correlation computation may not provide sufficient accuracy.

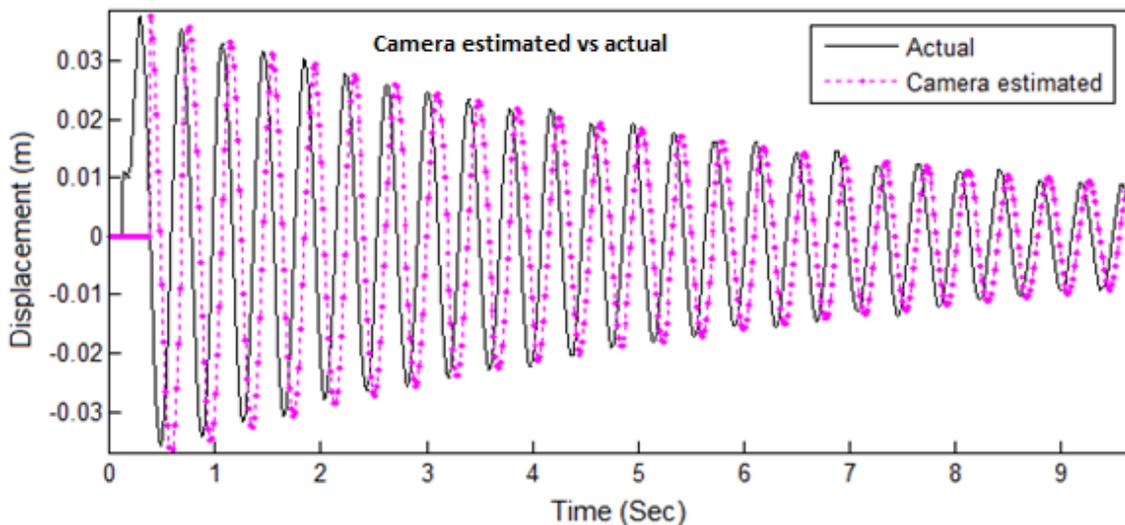
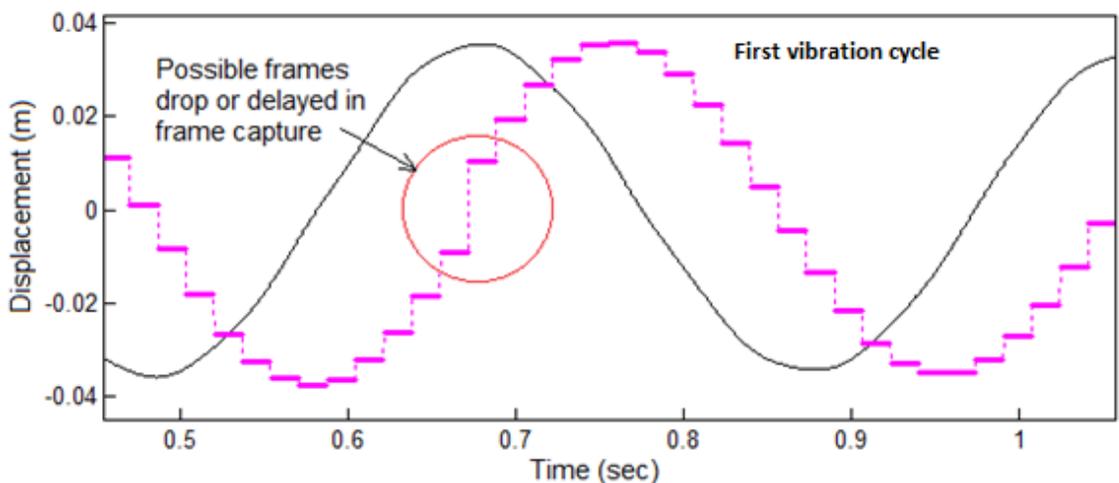


Fig. 4.35 Plot of tip vibration estimation using camera

The processing time of an image takes every 116 ms for each coordinate of the LED on the image plane. The displacement data obtained from camera is thus 116 ms lag. This displacement value is then held until the next image is acquired. Fig. 4.35 depicts the plot of the tip vibration estimation taken from the image. It can be seen that it matched quite accurately to the actual displacement with a lag.

However, the plot in Fig. 4.35 assumed that the time interval between each frames are constant. In actual fact, the plot actually consists of some frames drops and inconsistency frame capture intervals. Fig. 4.36 illustrates a close-up view of the vibration at second cycle of the vibration displacement. As highlighted in red circle, it can be seen that at 0.67 seconds there is an overshoot of the displacement, which could be due to frame dropped or delayed in frame capture. As mentioned above, there is no way of knowing when the image was taken within the 16 ms time period. The frame drop or the time of delay in frame capture is not detectable.



**Fig. 4.36** Close-up view of second cycle of vibration displacement

### 4.3 Results for the encoder tests

In this section, the static and dynamic tests as established in 3.3.5 of Chapter 3 are carried out, and the corresponding test results are illustrated.

### 4.3.1 Results for static tests of the encoder

The static tests that were defined in part I of section 3.3.5 presented. The tests were carried with the input voltages of 0.67V to 6V at increment of 0.67V. Illustrated in Fig. 4.37 above are the results for the angular displacement computed of the wheel at the speeds between 0.86 rad/s to 11.10 rad/s. It can be seen that it has lower error at lower speeds, and at higher speeds the errors are higher.

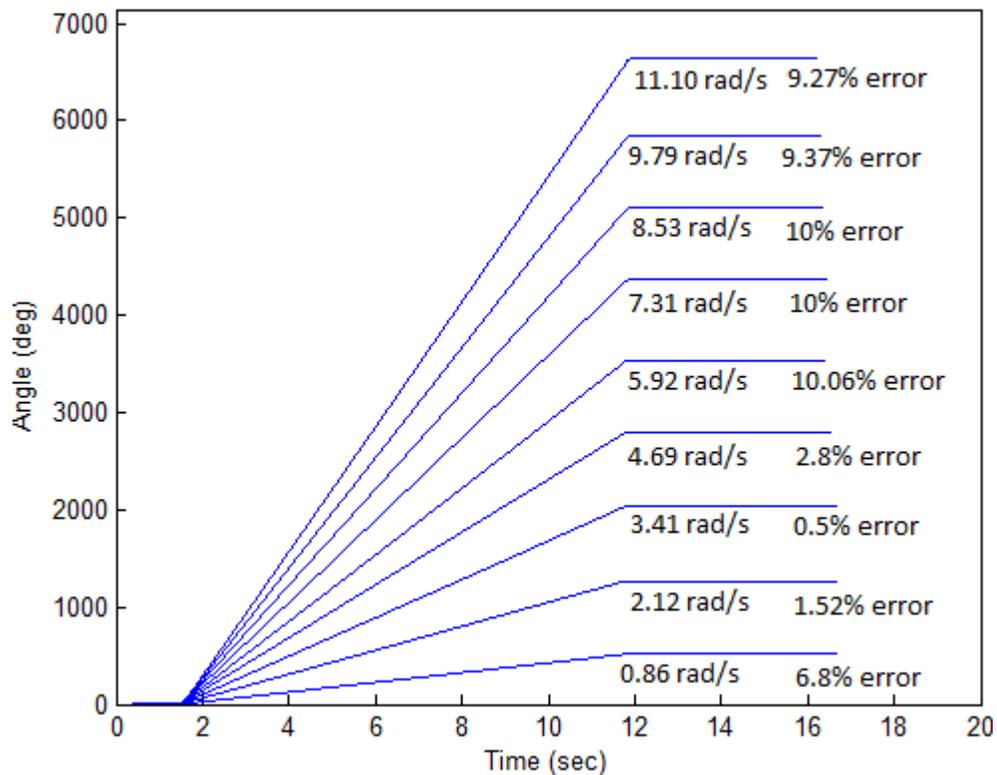
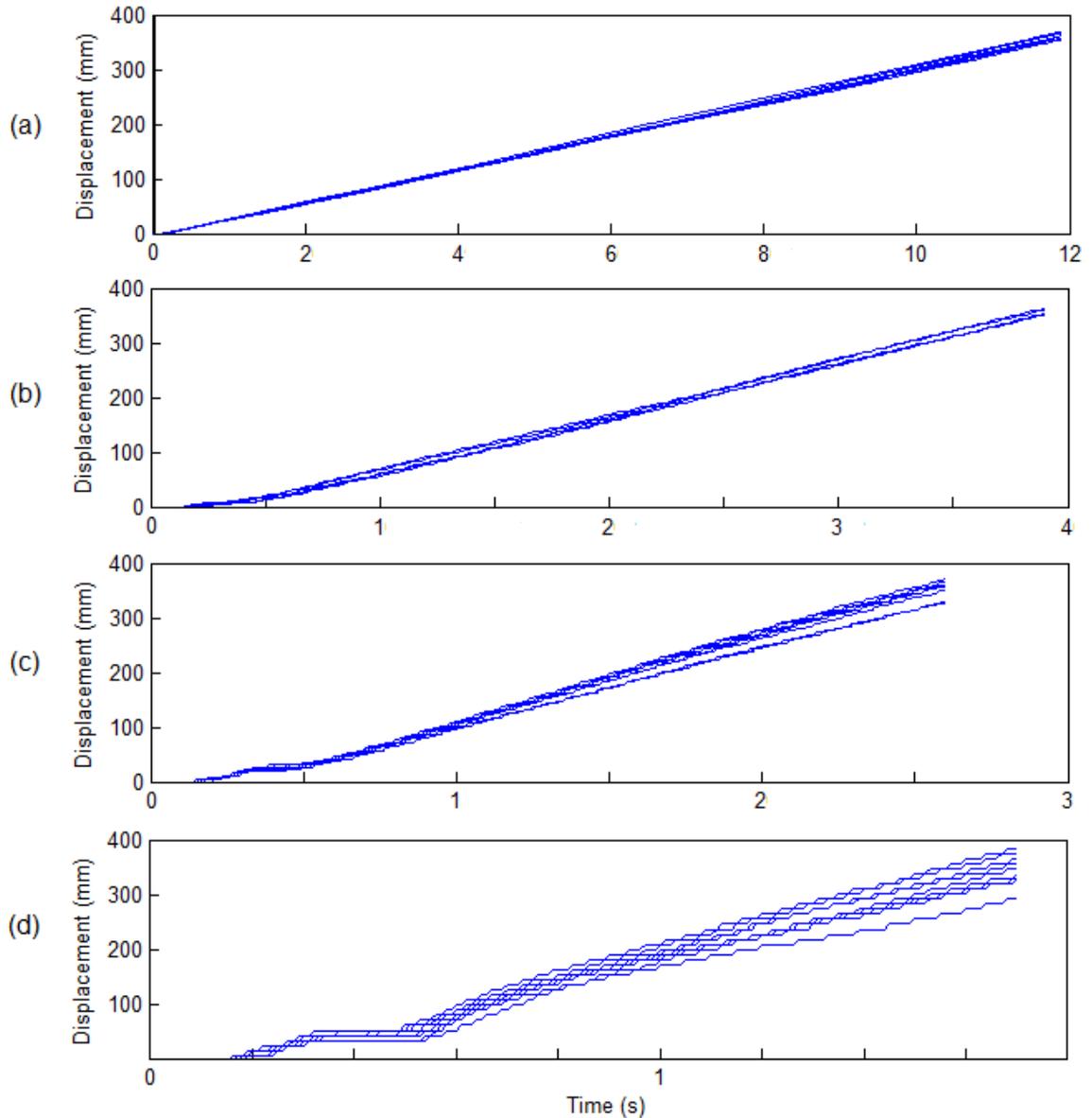


Fig. 4.37 Angular displacement estimation error of the motor wheel.

### 4.3.2 Results for dynamic tests of the encoder

The dynamic test that was set in part II of section 3.3.5 is presented. The encoder was tested with wheel rotating 370 mm linear displacement. The tests were carried out with 32 mm/s, 100 mm/s, 146 mm/s and 266 mm/s wheel's linear speeds.



**Fig. 4.38** Displacement estimation repeatability at various speeds, (a) 32 mm/s, (b) 100 mm/s (c) 146 mm/s, (d) 266 mm/s

8 set of tests for each speed were carried out. Fig. 4.38 above depicts the repeatability of the encoder for the estimations of the displacements. It can be seen that it has good displacement repeatability at 32 mm/s and 100 mm/s speeds, while the speed of 266 mm/s has worse accuracy and repeatability in the displacement estimation. Therefore, the accuracy of the encoder is poorer at higher speed. The contributing factors to this error include:

- Systematic errors

- Calculation for the wheel's circumference may be inaccurate.
- Mis-count of the encoder pulse count.
- Non-systematic errors
  - Backlash occurring between the gear, shaft and wheel.
  - Delay in wheel rotation.

## 4.4 Simulations for the models of motor

This section starts with identify constants for the motor at set in section 3.7, followed by the simulations and comparison to verify the accuracy of the models and the obtained constant values.

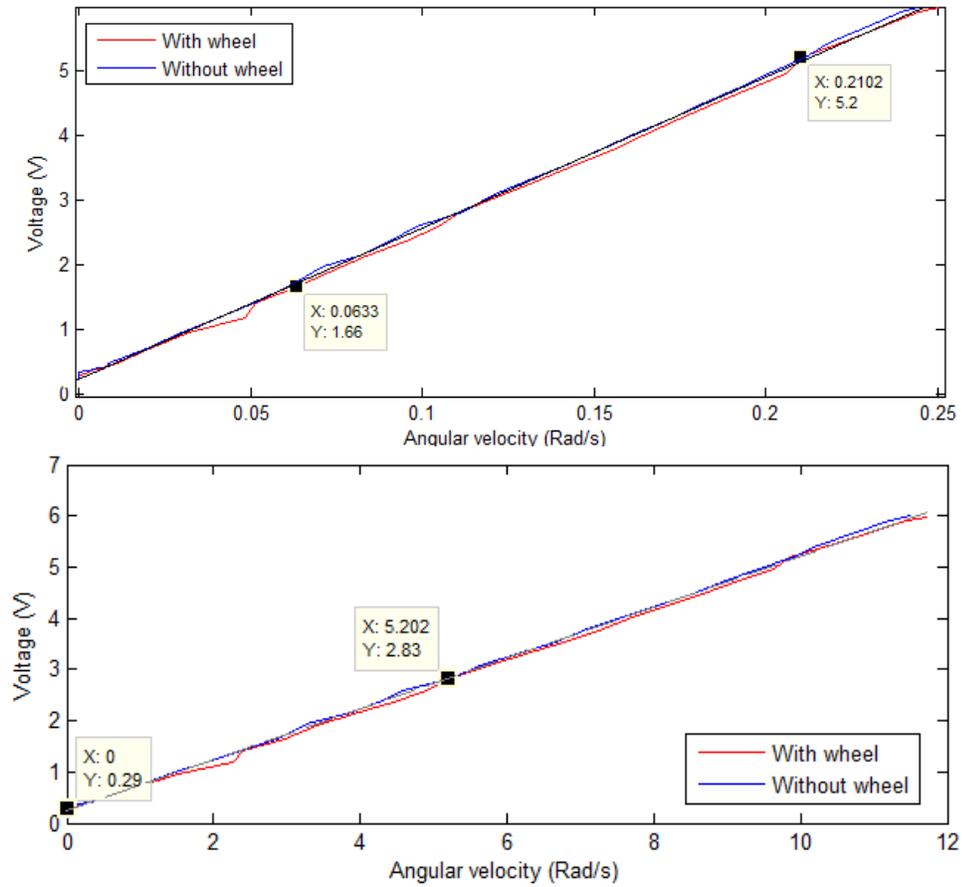
### 4.4.1 Identifying the constants for the motor

In this section, the constants for the motor as defined by the methodology in section 3.7 are presented. The experimental tests results include EMF constant, armature resistance, armature inductance, torque constant and moment of inertia. Experimental test results also include the viscous friction coefficient, coulomb friction and static friction.

#### I. Identifying back EMF constant and armature resistance

The methodology defined in section 3.7.1 is presented. The enhanced 10 Amp DC motor driver (MD10C) was used to control input voltage to the motor, while the MD10C motor driver was controlled via Arduino microcontroller. Fig. 4.39 exhibits the results of the measurements. By drawing a straight line along the curve, we can determine the back EMF constant  $K_e$  as equalling the slope of the line, which yields 0.482 rad/s/V.

By utilising equation (3.132) we can determine the armature resistance at stall, in which  $\omega_m = 0$ . Thus, we have  $V = RI$  which determined the armature resistance as 4.36  $\Omega$ .



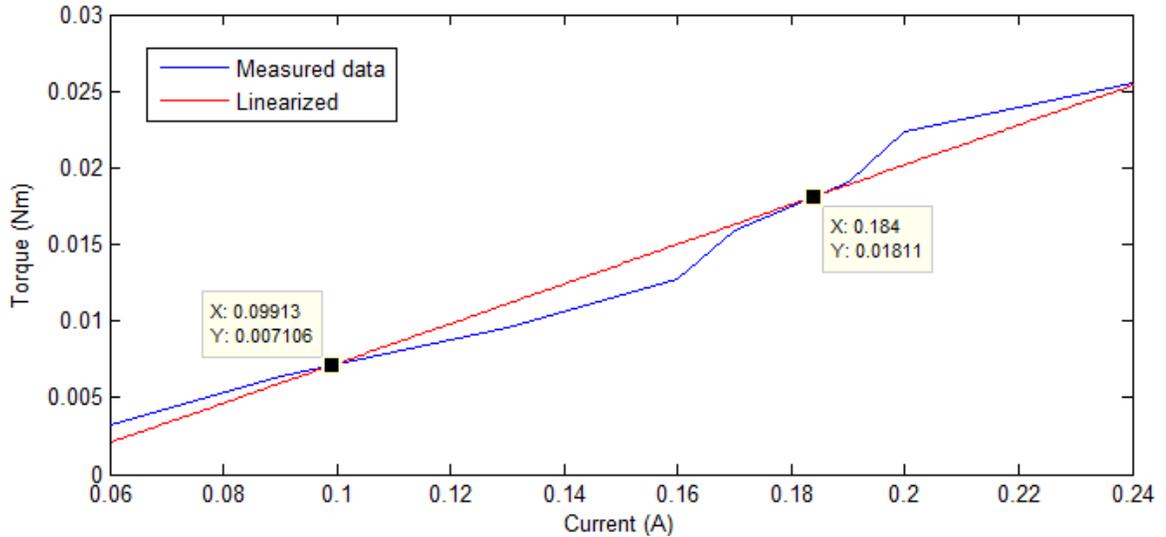
**Fig. 4.39** Steady state result of voltage VS angular velocity of motor with and without wheel

## II. Identifying torque constant

With the test methodology explained in section 3.7.2, we identified the torque constant as follows. Fig. 4.40 exhibits the results for the above tests with load weight of 10 g to 80 g. Each test incremental of 10 g loads. By drawing a straight line along the curve, the equation of the straight line is determined as:

$$y = 0.1297x - 0.00575$$

where  $y$  is torque  $\tau_m$  and  $x$  is current  $I$ . We determined the torque constant  $k_t$  as equalling the slope of the line, which is 0.1297 Nm/A.



**Fig. 4.40** Steady state result of current VS torque

### III. Identifying armature inductance and moment of inertia

With the test methodology explained in section 3.7.3, we obtained the armature inductance and moment of inertia as follows. Within the first 2.6 ms the current went from 0 to 0.034 A with a 1.34 V step input, and applying equation (3.133) leads to  $L = 1.9$  mH.

Upon released of the held rotor, the calculated angular acceleration  $\omega_m/dt$  is 0.9466 rad/s<sup>2</sup> at steady release current of 0.202 A. The friction  $T_{fr}$  and coupling torque  $\tau_{coup}$  is negligible when nothing attached the motor. Applying equation (3.16) resulting in the motor inertia  $J_m$  of 0.08083 kgm<sup>2</sup>.

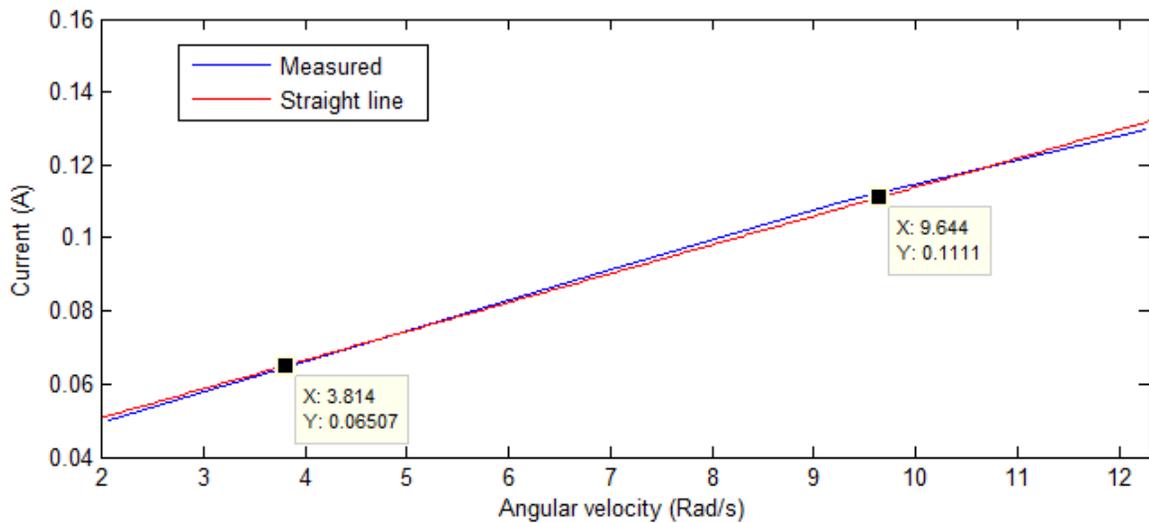
### IV. Identifying viscous friction coefficient and Coulomb friction

With the test methodology explained in section 3.7.4, we identified the viscous friction coefficient and coulomb friction as follows. Fig. 4.41 exhibits the results of the current against steady state angular velocity at no load condition. A straight is drawn is along the curve, and the equation for the straight line calculated as:

$$y = 0.00102x + 0.00452$$

where  $y$  is  $k_i I_a$  and  $x$  is  $\omega_m$ . For motor shaft rotating at steady state, the static friction and stribeak effect is zero. Therefore, according to equation (3.18), we left only the viscous

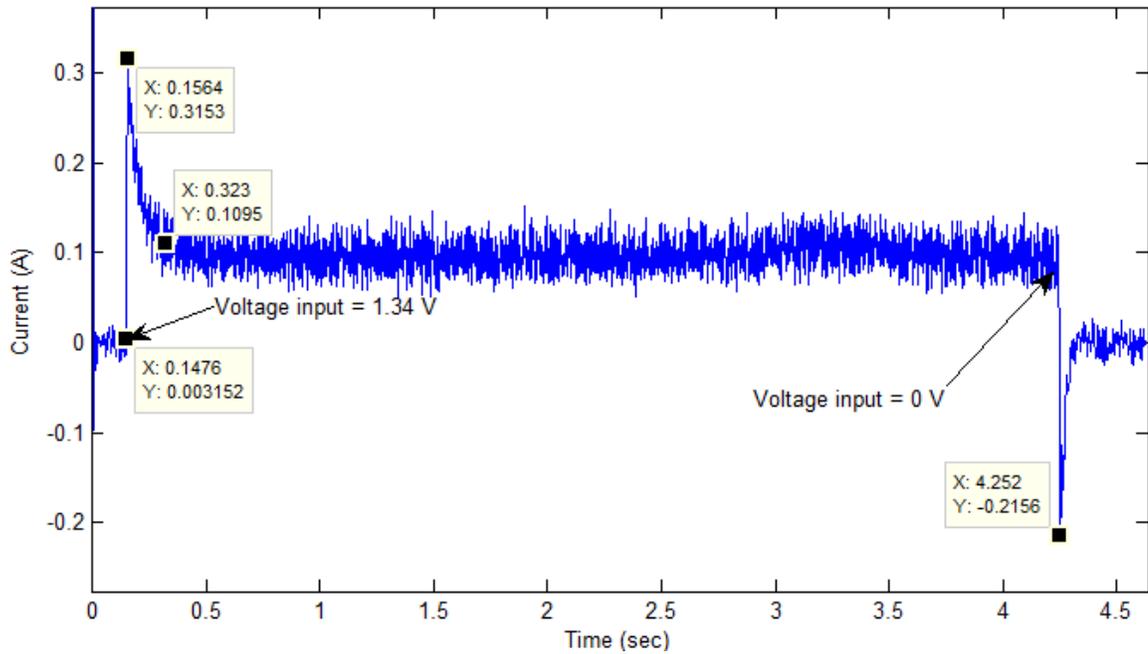
friction and Coulomb friction. From the above equation, since viscous varies with shaft rotating speed the viscous friction coefficient  $\nu$  is determined as 0.00102 Nm/rad/s. The coulomb friction torque determined as 0.00452 Nm since it is not dependent on the shaft speed.



**Fig. 4.41** Steady state result of current VS angular velocity of motor without load

## V. Identifying static friction and stribeak effect

With the test methodology explained in section 3.7.5, we identified the static friction and stribeak effect here. At no load condition the static friction torque is calculated to be 0.00782 Nm. Fig. 4.42 below illustrates the current measurement for the static friction and stribeck effect for the experiment of 20 g load at 1.34 V step input voltage. The peak current to break away the static friction is 0.3153 A. It took 8.8 msec to break away the static friction, and the stribeck effect took 166.6 msec to decay, exponentially.



**Fig. 4.42** Illustration of static friction and stiction effect

## VI. Tabulated motor parameters

Using the derived models in section 3.4.1 and the constants of the DC gear motor obtained in part I to V above, the parameters of the DC gear motor is tabulated in Table 4.4 below.

The simulations for the speed/time and position/time curves were then carried out. These curves can be used to evaluate and validate the model accuracy of the DC motor, comparing with the outputs acquired from the test rig.

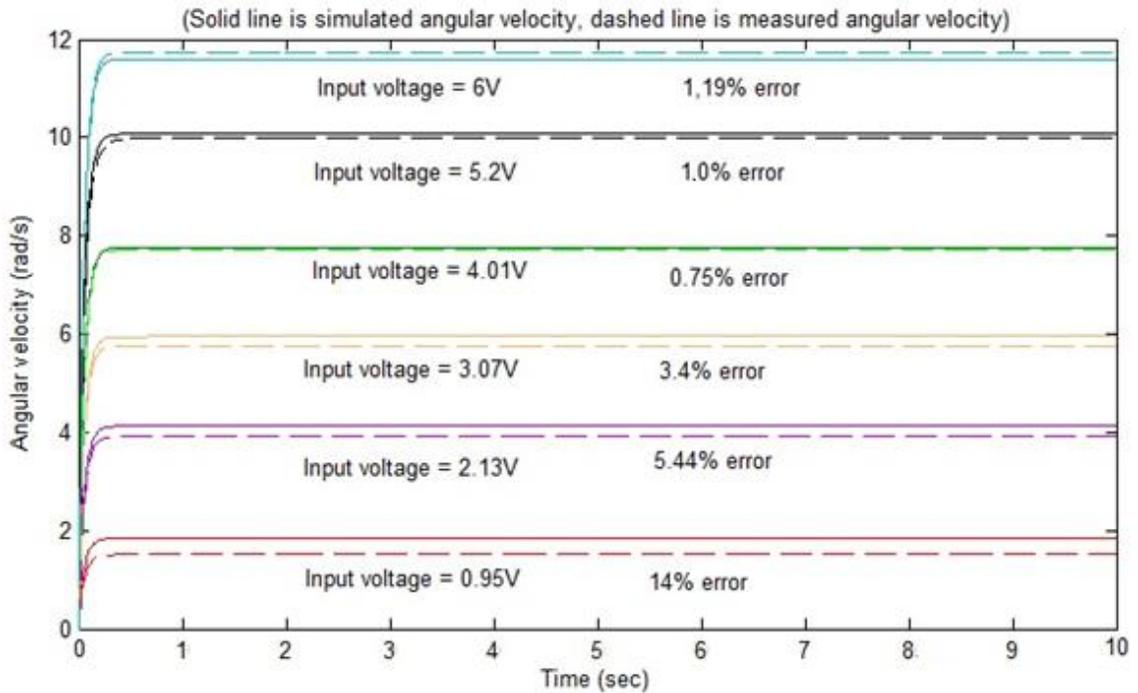
**Table 4.4** Motor parameter

Gear ratio	46.85:1 (47:1)
Free-run speed at 6V	120 rpm
Free-run current at 6V	80 mA
Stall current at 6V	2200 mA
Stall torque at 6V	3.6 kg.cm
EMF constant, $k_e$	0.482 rad/s/V
Armature resistance, $R$	4.36 $\Omega$
Torque constant, $k_i$	0.1297 Nm/A
Armature inductance, $L$	1.9 mH
Motor inertia, $J$	0.08083 kgm <sup>2</sup>
Quadrature encoder counts of motor shaft	48 CPR
Encoder counts of gearbox output shaft	2248.8 counts per revolution
Wheel dimension of driving wheel	65 mm dia. by 26 mm wide
Wheel coefficient of friction	1.1 for marble floor
Viscous friction coefficient, $\nu$	0.00102 Nm/rad/s
Coulomb friction torque, $T_{coul}$	0.00452 Nm
Static friction torque, $T_{static}$	0.00782 Nm
Stibeck effect	166.6 msec

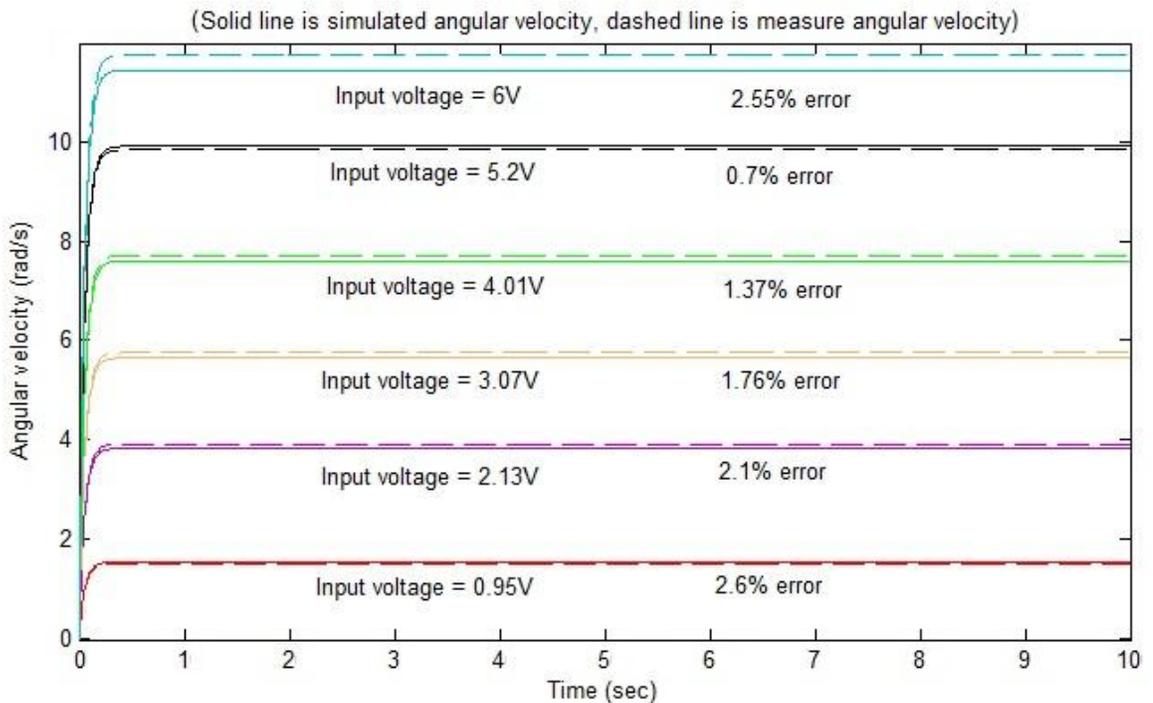
#### 4.4.2 Simulation results for testing the motor model

With the methodology set in section 3.7.6 and considering the parameter of the motor in Table 4.4 above, the purposes of the simulation are carried out and the results illustrated in this section.

For the first simulations, model of equation (3.47) was used, where we assumed no coulomb friction. Fig. 4.43 depicts simulation results and the comparison to the actual tests, and the percentage errors. Higher error is seen at low speed, with highest 14% error at 0.95 V input. This could be due to the effect of frictions that the motor need to overcome during when the shaft rotates. At higher input voltages the error is lower, until 4.01V input where it is only 0.75% error. This could be due to the breakthrough of the friction at the 4.01V. The errors are slightly increases in at 5.2V and 6V by 1% and 1.19%, respectively. This could be due to the inaccuracy in the identified motor constants in the preceding sub-sections, but the errors are quite close.



**Fig. 4.43** Angular velocity estimation comparing simulated output with actual output (assumed no coulomb friction)



**Fig. 4.44** Angular velocity estimation comparing simulated output with actual output (Coulomb friction considered)

In the second simulations, model of equation (3.48) was used, where we include frictions into the simulations. Fig. 4.44 illustrates the results for the various input voltages. It can be seen that the errors have improved for the input voltages of between 0.95V and 5.2V range. Similar to Fig. 4.43, at 6V input the error increases slightly. Likewise, this could be due to the inaccuracy in the identified motor constants, and the selection of gains. However, the amount of error is acceptable. It thus showed that the methods of identifying the frictions are quite accurate, and by including the friction the model become more accurate. Therefore, the model equation (3.48) that considers coulomb friction is suggested for use in this research.

## 4.5 Chapter Conclusion

The tests on the accelerometer affirm accelerometer can deliver realtime motion tracking to the subject that we intended to measure. With the recommended signal frequency of 11 KHz (see Appendix B.1, pg. 243) we are able to signals with no aliasing problem. As illustrated in the test results, the acquired acceleration signals are filled with noises and contain errors. In order to filter out the noisy signal, low-pass filter has been selected due to its acceptable shape preserving capability of the signal and its low processing time.

Another error to deal with for accelerometer's signal is the walking bias (or drifting) error. A method that uses standard deviation of less than 0.1 is used to illuminate this error. The results showed 100% improvement for signals with zero acceleration and 90% improvement for signal with constant acceleration.

The dynamic tests on the accelerometer furnished evidences that the traversing speeds of less than 15 cm/s has very poor signal-to-noise ratio. It is good in tracking displacements at 25 cm/s and above speeds. This makes accelerometer good for direct vibration estimation for flexible beam, but it is only good for short duration accuracy of only a few 10 seconds.

The cross-axis sensitivity of the sensor contributed to errors at higher motion speeds. This cross-axis error cannot be illuminated with direct algorithm on the signal. It would require perfect alignment of the sensor towards its axis of measurement, which is impossible. Thus, other methods, such as sensor fusion with camera using Kalman filter

model is proposed. The accelerometer provides the high frequency measurement at real-time with errors; while camera provides accurate low rate of measurement to correct the errors from the accelerometer at a delay. The fusion of these two sensors can thus improve the measurement accuracy.

Oppose to accelerometer, camera is good for slow speed displacement measurement. The tests on the webcam demonstrated that the speeds of above 30 cm/s have very poor capability for motion tracking. There are frames drop and the image of the marker became very much distorted and blurred. Thus, it is not recommendable to use webcam for motion detection of mobile platform at speed above 30 cm/s.

The vibration test for the CMOS camera proved to be able to track vibration of flexible beam, and was quite accurate. However, there present time lag which make camera not suitable for real-time tracking of flexible vibrations. Fusion with accelerometer would be recommended to promote the real-time vibration tracking capability.

The encoder is well known for measuring displacement for mobile robots, but is notorious for bring unreliable. The systematic and non-systematic errors such as mechanical variation errors, encoder mis-count and error in wheel rotation resulted cumulative error. The experiments demonstrated that higher accuracies are at speeds of up to 100 mm/s. Higher speeds, such as 266 mm/s, yielded higher error in the displacement measurements. However, a fusion with webcam would enhance the displacement tracking accuracy.

In validating the accuracy of the motor model developed in chapter 3, simulations of the motor revealed that the simulation and actual output are quite comparable when motor model with coulomb friction is considered. This proved that the simple methods that were used for identifying the parameters and friction of the motor's model are quite accurate.

## **5 Mobile flexible link manipulator and its predictive sensor system**

The uncertainties associate with the mobile flexible manipulator caused accurate prediction difficult. Therefore, it is required to integrate the signal from the sensors to achieve optimum estimation accuracies for vibration estimation of the manipulator.

This chapter covers research objectives points 5, 6 and 7 that are set in section 1.2. It is separated into three parts. First part presents the introduction and analysis of the fusion and prediction algorithms, which covers sections 5.1 to 5.3.

Second part presents the building and simulation of the flexible link manipulator and mobile flexible link manipulator (MFLM), which covers sections 5.4 to 5.6.

The third part presents the experimental tests for the MFLM with sensors and with fusion algorithms, covering sections 5.7 to 5.9.

### **5.1 Kalman filter and cross correlation algorithms**

To attain an optimum prediction accuracy of the vibration at the tip of the flexible manipulator, the important requirements for employing sensor fusion algorithm are:

1. Reliability: It should be reliable in providing vibration tracking. If one sensor fails or data drops, it should continue to provide sensory output.
2. Robustness: It should robust in terms of vibration tracking at wide range of variations where the system can operate.
3. System dynamics: The dynamic of the mobile flexible manipulator considered in this research is non-linear. It needs to be able to estimate motions of non-linear system.
4. Prediction: It needs to be able to predict ahead in time the motion trajectories of the manipulator.

With the requirements as stated above, Kalman filter is most appropriate for the sensor fusion for the accelerometer and camera used due to its recursive nature. The prediction stage in the Kalman filter made it readily available for implementing prediction of the future vibration, which is the objective of this research. Thus, only the algorithms for the Kalman filter and Extended Kalman filter are presented here.

### 5.1.1 Kalman filter

Kalman filter (KF) is the most widely known Bayesian filtering method [104]. KF equations consist of two sub-groups [51]: the time update group and the measurement update group. The time update group of equations are accountable for predicting forward using the current state and error covariance estimates for obtaining priori estimates for the next time step. The measurement update equations are accountable for putting together a new measurement from the sensors into the a priori estimate to obtain an improved a posterior estimate.

Consider the tracking end-point displacement of the flexible link manipulator using  $N$  number of sensors. Assuming that the sensors' are having identical sampling rates, we can write the signal model as [51, 104]:

$$x(k+1) = Gx(k) + e(k) \quad (5.1)$$

where,

$k$  = discrete-time index,

$x(k)$  = state vector,

$G$  = state transition matrix,

$e(k)$  = Gaussian white noise assuming known covariance matrix  $Q(k)$

$Q(k) = E[n(k)n(k)^T]$ .

The measurements corresponding to the sensor is [51, 104]:

$$z_i(k) = H_i x(k) + e_i(k), i = 1 \dots N \quad (5.2)$$

where,

$z_i$  = sensor  $i$  measurement vector,

$E_i$  = sensor  $i$  white Gaussian observation noise having zero mean and an assumed given covariance matrix  $R_i(k)$

$R_i(k) = E[e_i(k)e_i(k)^T]$ ,

$H_i$  = measurement matrix associated to the sensor  $i$  to  $N$  is the number of sensors.

With the model described by equations (5.1) and (5.2), the multisensory Kalman filter (KF) can be computed as [51, 104]:

- The estimation stage

$$\hat{x}(k|k) = \hat{x}(k|k-1) + \sum_{i=1}^N K_i(k)[z_i(k) - H_i \hat{x}(k|k-1)] \quad (5.3)$$

$$K_i(k) = P(k|k)H_i^T R_i^{-1}(k)H_i \quad (5.4)$$

$$P^{-1}(k|k) = P^{-1}(k|k-1) + \sum_{i=1}^N H_i^T R_i^{-1}(k)H_i \quad (5.5)$$

- The prediction stage

$$\hat{x}(k+1|k) = G\hat{x}(k|k) + Bu(k|k) \quad (5.6)$$

$$P(k+1|k) = GP(k|k)G^T + Q(k) \quad (5.7)$$

where,

$P$  = matrix provides the estimation uncertainty

$K$  = the sensor  $i$  Kalman gain for the data fusion

$u$  = control input.

$[z_i(k) - H_i \hat{x}(k|k-1)] = v_i(k)$  is the innovation associated to the observation for the sensor  $i$ . Kalman filter is limited to application on linear models with additive Gaussian noises. With system which is non-linear, as in the case of our system used in this research, thus Kalman filter is not optimal solution. The choice will be Extended Kalman filter.

### 5.1.2 Extended Kalman filter

Extended Kalman filter (EKF) generally involves a linearization of the problem [105]. In EKF, the model is linearized around the previous estimate [50]. Again consider the displacement at the tip of the flexible manipulator tracked using  $N$  number of sensors. Assuming that the sensors' signal have identical sampling rates; we can write the signal model as [50]:

$$x_{k+1} = f(x_k, u_{k+1}, n_k) \quad (5.8)$$

where,

$k$  = discrete-time index,

$x(k)$  = state vector,

$f(\cdot)$  = generic non-linear functions relating the past state and current input,

$n_k$  = system's Gaussian white noise of assumed known covariance matrix  $Q_k$   
 $Q_k = E[n_k n_k^T]$ .

The measurements corresponding to the sensor is [50]:

$$z_{ik} = h_i(x_k, b_{ik}), \quad i = 1 \dots N \quad (5.9)$$

where

$z_i$  = sensor  $i$  measurement vector,

$b_i$  = sensor  $i$  white Gaussian observation noise with zero mean and with assumed known covariance matrix  $R_{ik} = E[b_{ik} b_{ik}^T]$ ,

$h_i$  = sensor  $i$  measurement function associated

$N$  = number of sensors.

Let  $F(k)$  and  $H(k)$  be the Jacobian matrices of  $f(\cdot)$  and  $h(\cdot)$ ,

where

$$F(k) = \nabla f_k | \hat{x}(k|k)$$

$$H(k+1) = \nabla h | \hat{x}(k|k)$$

With the model described by equations (5.8) and (5.9), the EKF for multisensor fusion can be written as [50, 105]:

- The estimation stage

$$\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + \sum_{i=1}^N K_i(k+1) [z_{i(k+1)} - h_{i(k+1)} \hat{x}(k+1|k)] \quad (5.10)$$

$$K_i(k+1) = P(k+1|k) H_i^T(k+1) [H(k+1) P(k+1|k) H^T(k+1) + R(k+1)]^{-1} \quad (5.11)$$

$$P(k+1|k+1) = [I - K(k+1) \sum_{i=1}^N H_i^T(k+1)] P(k+1|k) \quad (5.12)$$

- The prediction stage

$$\hat{x}(k+1|k) = f(\hat{x}(k|k), u(k|k)) \quad (5.13)$$

$$P(k+1|k) = F(k) P(k|k) F(k)^T + Q(k) \quad (5.14)$$

where

$P$  = matrix for uncertainty on the estimate

$K$  = sensor  $i$  Kalman gain for the data fusion,

$u$  = control input.

The innovation [105] associated to the observation for the sensor  $i$  is given by  $[z_i(k) - Hixkk-1= vik$ .

### 5.1.3 Cross-correlation

Consider two periodic signals  $x_a(n)$  and  $x_b(n)$  subjected to the same period of  $N$  number of samples, the cross-correlation function having  $N$  samples can be defined as [106]:

$$C_{xy}(m) = \frac{1}{N} \sum_{n=1}^N x_a(n)x_b(n-m) = \frac{1}{N} \sum_{n=1}^N x_a(n+m)x_b(n) \quad (5.15)$$

Let accelerometer signal be  $z(n)$ , with noise  $n(n)$  derives as  $x_a(n)$ , and the visual signal (assuming that the visual data has same sampling rate) is  $z(n)$  delayed by  $n_o$  samples as  $x_b(n)$  [106]:

$$x_a(n) = z(n) + n(n) \quad (5.16)$$

$$x_b(n) = z(n - n_o) \quad (5.17)$$

The cross-correlation between  $x_b(n)$  and  $x_a(n)$  during  $M$  samples is [106]:

$$C_{x_a x_b}(m) = \frac{1}{M} \sum_{n=1}^M x_b(n)x_a(n-m) \quad (5.18)$$

Substituting (5.16) and (5.17) into (5.18), and performing mathematical manipulation, yields [106]:

$$C_{x_b x_a}(m) = \frac{1}{M} \sum_{n=1}^M z(n - n_o + m)z(n) + \frac{1}{M} \sum_{n=1}^M z(n - n_o + m)n(n) \quad (5.19)$$

$$C_{x_b x_a}(m) = C_{zz}(m - n_o) + C_{zn}(m - n_o) \quad (5.20)$$

where  $C_{zz}(m - n_o)$  is the auto-correlation of the time lagged periodic signal lag in time, and  $C_{zn}(m - n_o)$  is the cross-correlation between the time lagged periodic signal and noise.

A fixed-size sliding window cover at least one period of the cross-correlation for signals coming from the inertial data and visual data is employed to estimate the capture delay of the camera images. The capture delay for each visual data is not constant, while the cross-correlation is computed over at least one period of the oscillating signals, thus it is required to take into account of the last measured value  $P$ .

Consequently, equation (5.19) becomes [106]:

$$\begin{aligned}
C_{x_b x_a}(m) &= \frac{1}{M} \sum_{n=1}^{M-P} x_b(n) x_a(n-m) + \frac{1}{M} \sum_{n=M-P+1}^M x_b(n) x_a(n-m) \quad . \\
&= \frac{M-P}{M} C_{x_b x_a}^{old}(m) + \frac{P}{M} (C_{x_b x_a}^{new}(m) - C_{x_b x_a}^{old}(m)) \quad . \quad . \\
&= C_{x_b x_a}^{old}(m) + \frac{P}{M} (C_{x_b x_a}^{new}(m) - C_{x_b x_a}^{old}(m)) \quad (5.21)
\end{aligned}$$

where  $C_{x_b x_a}^{old}(m)$  computes the cross-correlation for the values before the last reception of visual data, and  $C_{x_b x_a}^{new}(m)$  computes the cross-correlation for the latest data. In order to put more weightage on recent measured values, a  $k$  term can be added, equation (5.20) becomes [106]:

$$C_{x_b x_a}(m) = C_{x_b x_a}^{old}(m) + \frac{kP}{(1-k)(M-P)+kP} (C_{x_b x_a}^{new}(m) - C_{x_b x_a}^{old}(m)), 0 \leq k \leq 1 \quad (5.22)$$

The capture delay  $n_o$  is  $C_{x_b x_a}(n_o) = \max [C_{x_b x_a}(m)]_{0 \leq m \leq N}$ .

## 5.2 Development of sensor fusion algorithm

The Kalman filter algorithm as reviewed in section 5.2 is used to develop the sensor fusion algorithm. As the noise is non-Gaussian and the system is non-linear, it is proposed to use Extended Kalman filter (EKF) to fuse the camera and accelerometer data. Cross-correlation combines with the EKF to align the camera estimates with the accelerometer estimates. The algorithm works such that the camera output readjusted the drifted displacement output from accelerometer the through Extended Kalman filtering, while the delayed visual data is reconstructed by the accelerometer signal using cross-correlation.

### 5.2.1 Sensor fusion algorithm

The basic research methodology used in this thesis is iterative design and experimentation. The data rate for the camera is low with a delay, while the data rate for the accelerometer is fast and realtime but with drifting outputs. The sensor fusion thus work such that when both accelerometer's and camera's data are available, EKF performs data fusion from the input of both sensors. When the camera data is not available, the EKF take either: 1) only accelerometer data; 2) combines accelerometer data with previous camera data; 3) combines accelerometer data with extrapolation of

previous camera data; or 4) combining modelled data and extrapolation of previous camera data. In addition, the input to the EKF is either with modelled data input or with acceleration input.

With these considerations in the course of the design iteration, five types of EKF algorithms were established. For the duration of absent of camera's data, the five types of EKF fusion algorithms are as follows:

EKF Type 1: No camera interpolation, but with modelled input:

During the absent of the latest camera data, the algorithm combine the previous camera data with the latest accelerometer output, and integrate with the simulated input from the MFLM model to feed to the sensor update stage of the EKF.

EKF Type 2: Combine camera data extrapolation and model, and with modelled input:

During the absent of the latest camera data, the previous camera data are extrapolated to predict the latest camera data and combine with simulated output from the MFLM model. The output is then combines with the latest accelerometer data, and integrate with the simulated input from the MFLM model to feed to the sensor update stage of the EKF.

EKF Type 3: With camera data extrapolation, with no modelled input:

During the absent of the latest camera data, the previous camera data are extrapolated to predict the latest camera data and combines with the latest accelerometer data to feed to the sensor update stage of the EKF.

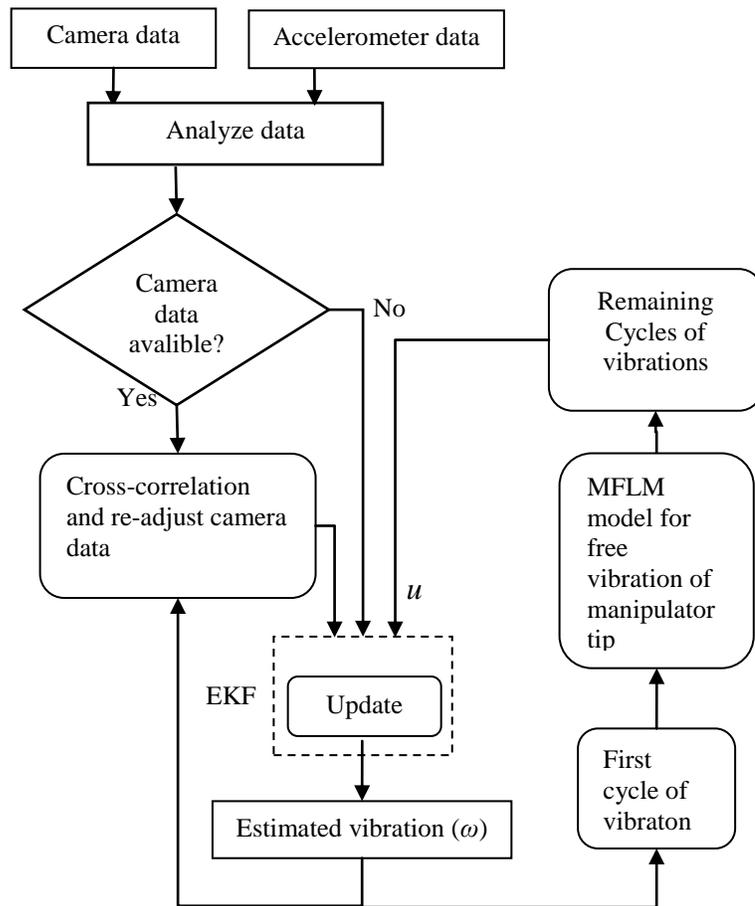
EKF Type 4: With modelled data in place of camera data, and with modelled input:

During the absent of the latest camera data, the simulated output from the MFLM model is combine with the latest accelerometer output, and integrate with the simulated input from the MFLM to feed to the sensor update stage of the EKF.

EKF Type 5: No camera data, and no modelled input.

During the absent of the latest camera data, only the latest accelerometer data is fed to the sensor update stage of the EKF. No input is integrated to feed to the update stage of the EKF.

The description of the EKF is in section 5.1. Fig. 5.1 describes the block diagram for the proposed sensor fusion algorithm. The algorithm works such that the input  $u$  to the EKF is the output waveform computed by the combination of the equations (3.48) and (3.130) for modeling the vibration of flexible beam reacted under the motion of the mobile platform are input voltages to the driving motor. It was realized that the natural frequency  $\omega_n$  and amplitude  $y_o$  of the vibration can be calculated using the first cycle of the waveform for the vibration when the beam was being excited. With this information, the payload at the tip of the beam can also be determined using reverse dynamics. The modelled vibrational waveform of the response from the mobile flexible beam can then be computed, and feed to the EKF as input  $u$ .



**Fig. 5.1** Block diagram of displacement estimation algorithm for flexible beam of augmented Extended Kalman filter

When the data from camera becomes unavailable, the data update to EKF is based on accelerometer's signal. When the data from camera becomes available, cross-correlation is first compute the delayed frame and the resulting vibrational data readjusted and feed to the EKF as sensor update.

## 5.2.2 Applying the extended Kalman filter

With the vibration at the tip of the MFLM defined as  $w$ , the state model of equation (5.8) can be defined using of current estimate  $w_k$  plus the incremental change in state  $\Delta w_k$ , which linearized yields:

$$f_x = w_{k+1} = w_k + \Delta w_k \quad (5.23)$$

The incremental change in the deflection  $\Delta w_k$ , can be obtained utilizing equations (3.130) and (3.131), giving:

$$\dot{X} = \begin{bmatrix} 0_{2 \times 2} & I_{2 \times 2} \\ -M(q)^{-1}R(q) & 0 \end{bmatrix} X + \begin{bmatrix} 0_{2 \times 1} \\ M(q)^{-1} \end{bmatrix} \tau \quad (5.24)$$

$$Y = [I_{1 \times 2} \ 0_{1 \times 2}] X \quad (5.25)$$

where

$$X = X = [w \ d \ \dot{w} \ \dot{d}]^T,$$

$w$  = deflection of the flexible beam,

$d$  = the displacement of the mobile platform.

The change in deflection  $\Delta w$ :

$$\Delta w_k = w_{k-1} - w_k$$

For matrix linearization, we obtain the transition matrix  $F_k$  and  $H(k)$  from the Jacobian matrix, which yields:

$$F_k = \begin{bmatrix} \frac{\partial f_w}{\partial w_k} & \frac{\partial f_w}{\partial \dot{w}_k} & \frac{\partial f_w}{\partial d_k} & \frac{\partial f_w}{\partial \dot{d}_k} \\ \frac{\partial f_{\dot{w}}}{\partial w_k} & \frac{\partial f_{\dot{w}}}{\partial \dot{w}_k} & \frac{\partial f_{\dot{w}}}{\partial d_k} & \frac{\partial f_{\dot{w}}}{\partial \dot{d}_k} \\ \frac{\partial f_d}{\partial w_k} & \frac{\partial f_d}{\partial \dot{w}_k} & \frac{\partial f_d}{\partial d_k} & \frac{\partial f_d}{\partial \dot{d}_k} \\ \frac{\partial f_{\dot{d}}}{\partial w_k} & \frac{\partial f_{\dot{d}}}{\partial \dot{w}_k} & \frac{\partial f_{\dot{d}}}{\partial d_k} & \frac{\partial f_{\dot{d}}}{\partial \dot{d}_k} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$H_k = 1$ , because there no unit transformation as the displacement input from the mobile platform is converted directly into displacement from equation (3.48).

$z_i$  is the measurement computed from the sensor. The sensors input are converted directly into displacement.

### 5.3 Development of predictive algorithm

With the discussions in part iii of section 2.5, predictive method has been proposed in this research. The state-space model is used to predict the future state, at  $j$  number of steps ahead, using previous and the current states. Future state variables are calculated sequentially using the set of future control parameters using the augmented state-space model. The linearized state-space model is for a single step ahead state/output prediction is [22, 72]:

$$\begin{cases} x(k+1) = \mathcal{A}x(k) + \mathcal{B}u(k) \\ y(k+1) = \mathcal{C}x(k+1) \end{cases} \quad (5.23)$$

where

$x \in \mathbb{R}^n$  is state vector

$u \in \mathbb{R}$  is system input

$y \in \mathbb{R}$  is system output.

$\mathcal{A} \in \mathbb{R}^{n \times n}$ ,  $\mathcal{B} \in \mathbb{R}^n$  and  $\mathcal{C} \in \mathbb{R}^n$  are transition matrices.

To predict the system's future response, we need to determine the change in the future input trajectory. The changes in future inputs are [74]:

$$\Delta u(k), \Delta u(k+1), \dots, \Delta u(k+N-1) \quad (5.24)$$

where  $N$  is the future input horizon in time. The predicted future state variables can be denoted as [74]:

$$x(k+1), x(k+2), \dots, x(k+N) \quad (5.25)$$

For the one-state ahead future state prediction,  $k+1$  is written as [74]:

$$x(k+1) = \mathcal{A}x(k) + \mathcal{B}\Delta u(k) \quad (5.26)$$

For the two-states ahead future state prediction,  $k+2$  is written as [74]:

$$\begin{aligned}
x(k+2) &= \mathcal{A}x(k+1) + \mathcal{B}\Delta u(k+1) \\
&= \mathcal{A}^2 x(k) + \mathcal{A}\mathcal{B}\Delta u(k) + \mathcal{B}\Delta u(k+1)
\end{aligned} \tag{5.27}$$

It follows that, for the  $j$  step-states ahead future predictions, it can be written as [74]:

$$x(k+j) = \mathcal{A}^j x(k) + \sum_{n=1}^j \mathcal{A}^{j-n} \mathcal{B} \Delta u(k+n) \tag{5.28}$$

The modelled MFLM input is fed to the change in input  $\Delta u(k+j-n)$ . The prediction algorithm works such that the future control input, which is the motor's voltage input. This yields the velocity output of the entire mobile robot. Thus, utilizing equation (3.48) to compute the future velocity of the mobile robot, we get:

$$v_{el} = V(k+j-n) \left[ \frac{k_i}{(Jm_s + v_c)(R+Ls) + K_e k_i} \quad \frac{-(R+Ls)}{(Jm_s + v_c)(R+Ls) + K_e k_i} \right] \tag{5.29}$$

where

$V$  = the motor's input voltage.

Then, the change of input  $\Delta u$  of equation (5.28) above is the acceleration of the entire mobile robot, which can be obtained by differentiating  $v_{el}$ :

$$\Delta u = \frac{\partial v_{el}}{\partial t}$$

Depending on the receding horizon of prediction,  $j$ , based on the state-space model of equation (3.130) and (3.131), the elements in the  $\mathcal{A}$  and  $\mathcal{B}$  matrixes are:

$$\mathcal{A} = \begin{bmatrix} AL \left( \frac{-16+6\pi}{4\pi} \right) \rho + m_p & 0 \\ 0 & m_B \end{bmatrix}$$

$$\mathcal{B} = \begin{bmatrix} \frac{EI\pi^4 w_0}{32L^3} \\ 0 \end{bmatrix}$$

The block diagram of the Kalman filter based predictive algorithm for the MFLM is illustrated in Fig. 5.2 below. The flow diagram of the MFLM block is illustrated in Fig. 5.3.

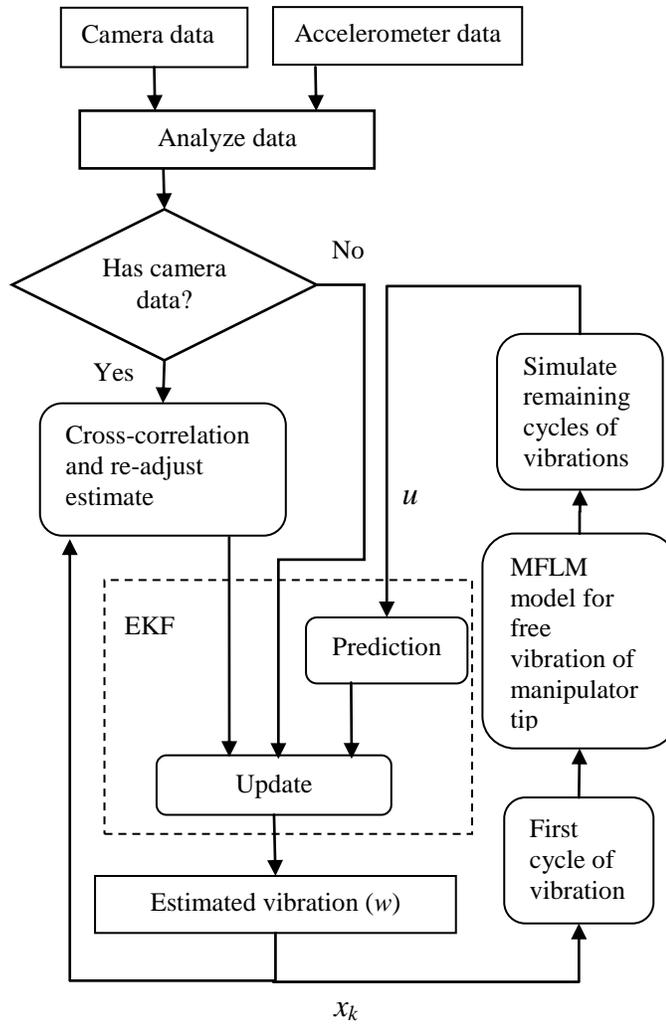


Fig. 5.2 Block diagram of Kalman filter based vibration prediction algorithm for MFLM

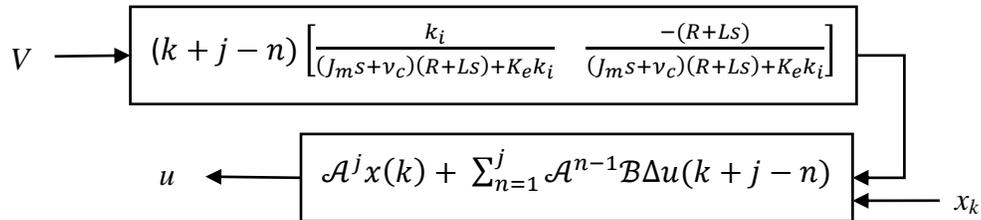
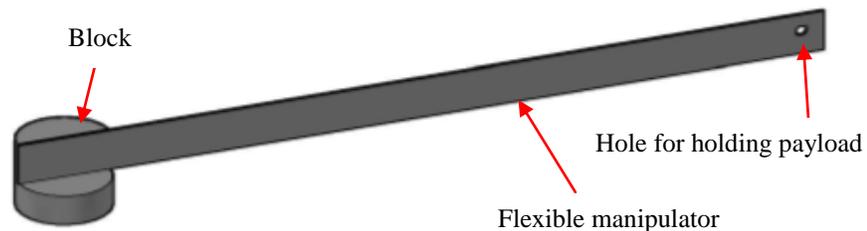


Fig. 5.3 Block diagram of the MFLM block

## 5.4 Fabrication and analysis of flexible manipulator

To design the flexible manipulator we consider the common domestic door passage, where the robot should pass through, for instance, to travel from living room to bed room. The allowance door is around 78 cm wide, so for the mobile robot to pass through the door without obstruction we consider the maximum allowable width of the robot to be 70 cm. If we consider the platform of the robot to be 40 cm wide so that it will not topple, then length of the manipulator should be around 50 cm if we mount the manipulator at the centre point of the mobile platform. Therefore, the flexible manipulator is fabricated with thin steel beam of 530 mm long, 28.8 mm width, and thickness of 0.96 mm.

One end of the beam is affixed to a block so that it can be mounted to a fixed platform. The flexible manipulator is fabricated in the Mechanical workshop at the University of Nottingham, Malaysia campus. Fig. 5.4 below depicts the isometric drawing of the flexible manipulator and the block that was being fabricated. The flexible manipulator is affixed to the round block.



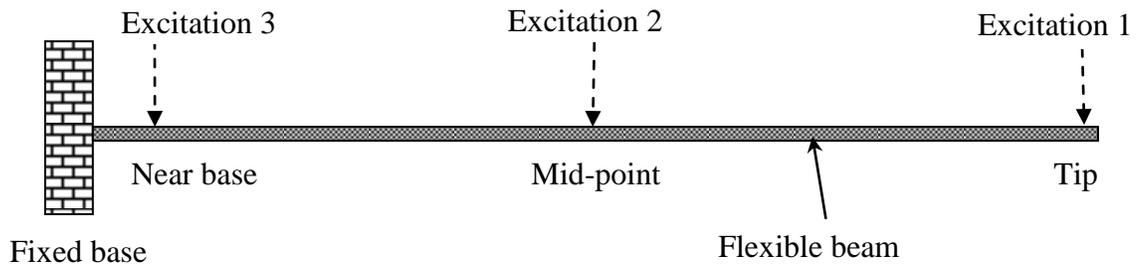
**Fig. 5.4** Flexible manipulator affixed to the block

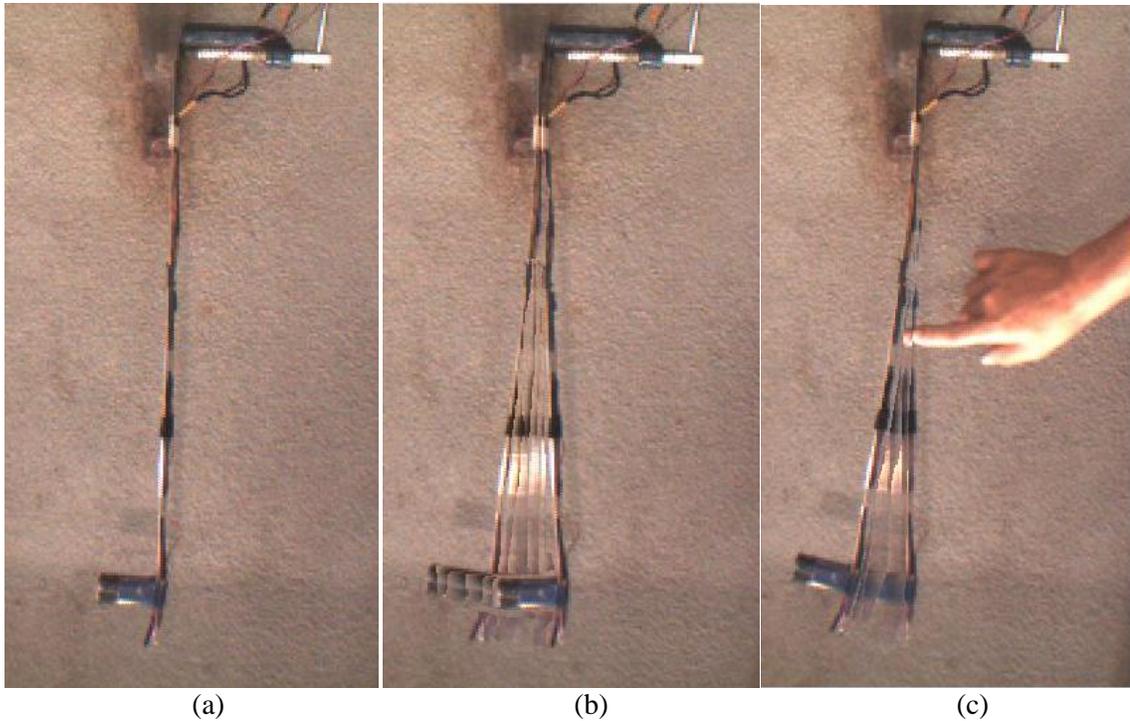
The physical parameters for the flexible manipulator are tabulated in Table 5.1. The beam's vibration can be analyzed as its frequency of vibration, and it is contributed by the mode shape of vibration. Higher mode shape would result in higher vibration frequencies for the given material of the beam.

**Table 5.1** Flexible beam parameters

Parameter	Name	Value
Young's Modulus	$E$	190 GPa
Poisson's Ratio	$\gamma$	0.27 – 0.3
Area moment of Inertia	$I$	$2.123 \times 10^{-12} \text{ kgm}^2$
Cross-sectional area	$A$	$0.96 \times 28.8 \text{ mm}^2$
Length	$L$	53 cm
Density	$\rho$	7308.864 kg/m <sup>3</sup>
Mass of beam	$m_b$	107.1 g
Stiffness	$k_{stiff}$	9.5376
Damping coefficient	$D$	0.024

To determine the modes of vibration for the manipulator used in this research, we investigate the natural modes of the fixed-free flexible beam as illustrated in Fig. 5.5. Theoretically, a flexible beam has infinite number of modes. But, the number of excited modes is finite due to finite energy along the beam and the rectangular cross-sectional dimension of the beam. For the fixed-free flexible beam, most researches assumed mode 1 [26] vibrations and some considered mode 2 [107] vibrations.

**Fig. 5.5** Flexible beam (with impulse excitation)



**Fig. 5.6** Flexible beam and vibration modes. (a) Beam at rest, (b) Vibration with impulse input at tip, (c) Vibration with impulse input at mid-point.

A high speed digital camera was used to record vibration of the beam when the tests were carried out. Fig. 5.6a shows the picture of the beam being tested, at rest position). Impulse inputs were excited at the tip, mid-point and near base of the beam to observe the transient responses. The beam was manually excited at the tip which was found to exhibit mode 1 response throughout the duration, as shown in Fig. 5.6b. For the excitation at mid-point, it exhibited a mode 2 deflection when the load was acting on the beam, and continued by mode 1 vibration when the load released, as shown in Fig. 5.6c. When excited near fixed end, mode 1 vibration was observed. Similar tests were performed with a 50 grams payload added to the tip of the beam, similar phenomenon was observed. Therefore, only the first mode vibration (or fundamental vibration frequency) has the most significant and observable contribution to the behaviour of the flexible beam.

By employing finite element analysis, the first bending mode shape of the cantilever beam is computed as illustrated in Fig. 5.7. It can be seen that the deflection matches that of the test investigation shown in Fig. 5.6. The maximum stress is at the fixed position of the beam.

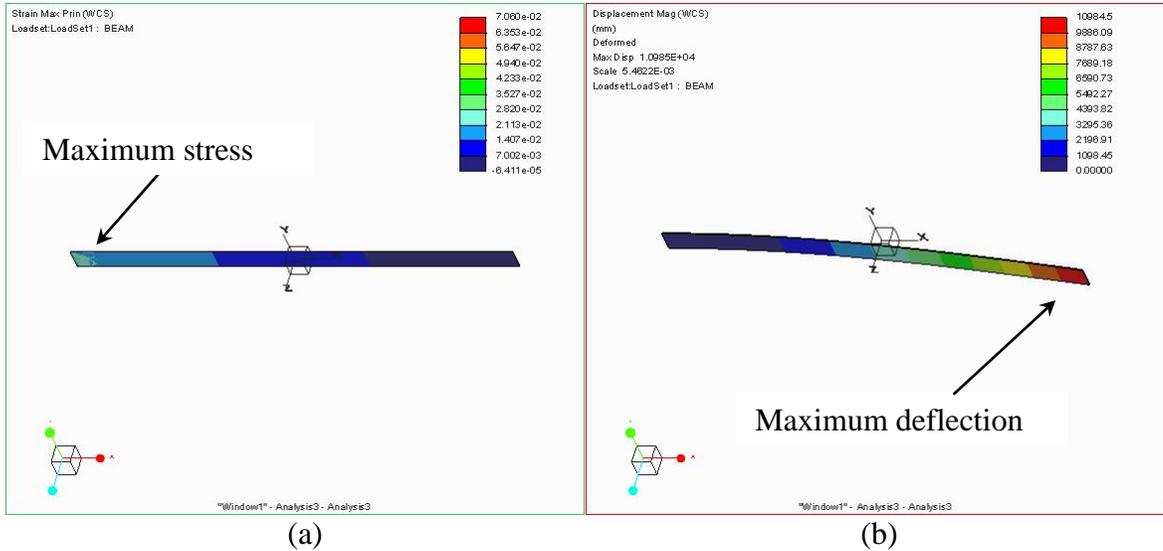


Fig. 5.7 FEA for cantilever beam: (a) stress analysis and (b) mode shape

Table 5.2 Force Vs vibration amplitude

Force (N)	Displacement (m)	Velocity (m/s)	acceleration ( $m/s^2$ )
50	0.00633	0.04526	0.3333
100	0.01263	0.09	0.6661
150	0.01887	0.1327	0.9965
200	0.02504	0.176	1.326
250	0.03109	0.2262	1.662
300	0.03737	0.2645	1.997
350	0.04425	0.3082	2.323
400	0.05	0.3621	2.652
450	0.05697	0.4009	2.984
500	0.06318	0.4532	3.297
1000	0.1241	0.9076	6.447
2000	0.235	1.726	11.17
3000	0.2862	2.229	13.4
3100	0.2912	2.273	13.51
3200	0.2923	2.282	13.6
3300	0.2925	2.283	13.63
3400	0.2923	2.284	13.65
3500	0.2916	2.278	13.67
3600	0.2906	2.258	13.65
3700	0.2826	2.251	13.62
3800	0.287	2.229	13.55
3900	0.2849	2.209	13.48
4000	0.2839	2.194	13.41

Table 5.2 illustrates the maximum vibration amplitude, velocity and acceleration for the respective impulse input forces with no load at the tip. Here, we discovered that the

maximum vibration amplitude at the beam's tip after being released from its deflected displacement. It can be seen that the maximum vibration amplitude saturated at around 3100 N actuating force. The highest actuating input force for highest vibration amplitude is around 3300 N, and the corresponding displacement, velocity and acceleration are 0.2925 m, 2.283 m/s and 13.63 m/s<sup>2</sup>, respectively. 13.63 m/s<sup>2</sup> acceleration equates to 1.37 g gravitational force. This vibrational acceleration is within the sensing range of the selected accelerometer, MMA7260Q (refer to section 3.1.3). The sensing range of  $\pm 2$  g has thus been selected.

## **5.5 Methodology for comparing simulated vibration to actual vibration of the flexible manipulator**

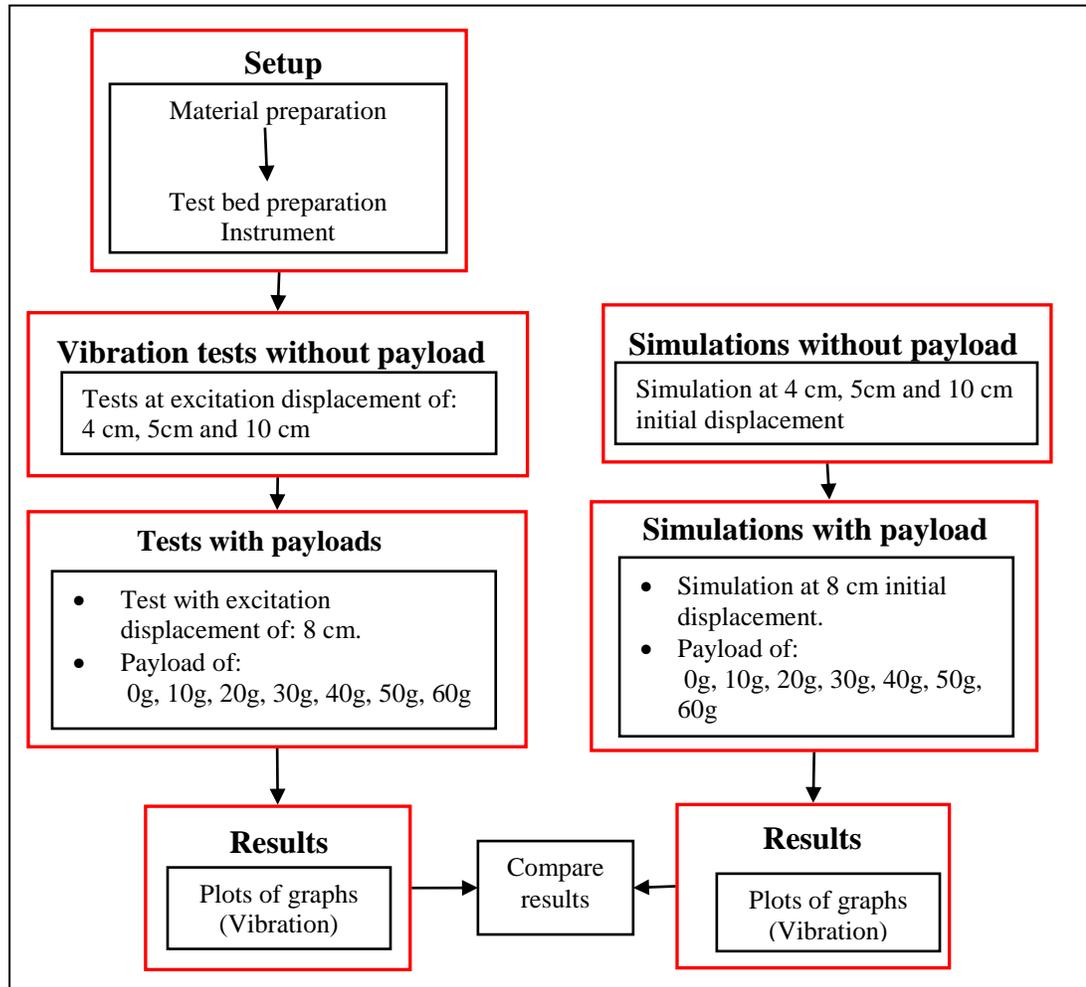
With the model of the flexible link manipulator being developed in section 3.5, we would like to investigate the accuracy of the model, done by comparing the free vibration at the tip of the flexible manipulator with the actual manipulator fabricated as described in section 5.4 above. To validate the mathematical model in equation (3.82), tests will be carried out at difference excitation displacement of the tip of the manipulator, and with difference payload attached to the tip. The methodology for the test is as follows:

- I. Setup: For the test setup, the flexible manipulator to be tested is mounted with one end to a fixed platform. The short range IR distance sensor (GP2Y0A21YK) is fixed at the free end of the manipulator to sense the vibration at the free end, and vibration signal acquired via NI 9201 module. Refer to Fig. 5.9 and 5.10 for the setup.
- II. Vibration tests:
  - a. Tests with no payload at the tip of the manipulator, but with excitation displacements of 4 cm, 5 cm, 8 cm and 10 cm.
  - b. Tests with various payloads of 0g to 60g with increment of 10g load fixed at the tip of the manipulator. Excitation displacement of 8 cm.
- III. Simulate vibrations:
  - a. Simulation of vibration of the flexible manipulator with no payload, but with initial simulation displacements of 4 cm, 5 cm, 8 cm and 10 cm.

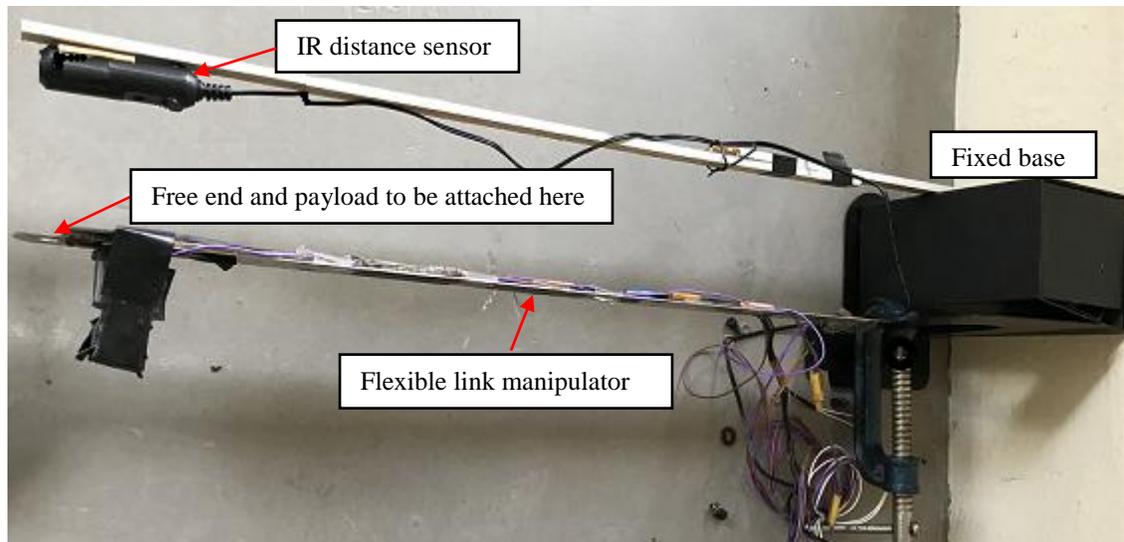
b. Simulation of vibration of the flexible manipulator with various payloads of 0g to 60g with increment of 10g load, with initial simulation displacement of 8 cm.

IV. Comparison: The test results are then compared with the simulation outputs to determine the accuracy of the mathematical model of the flexible manipulator.

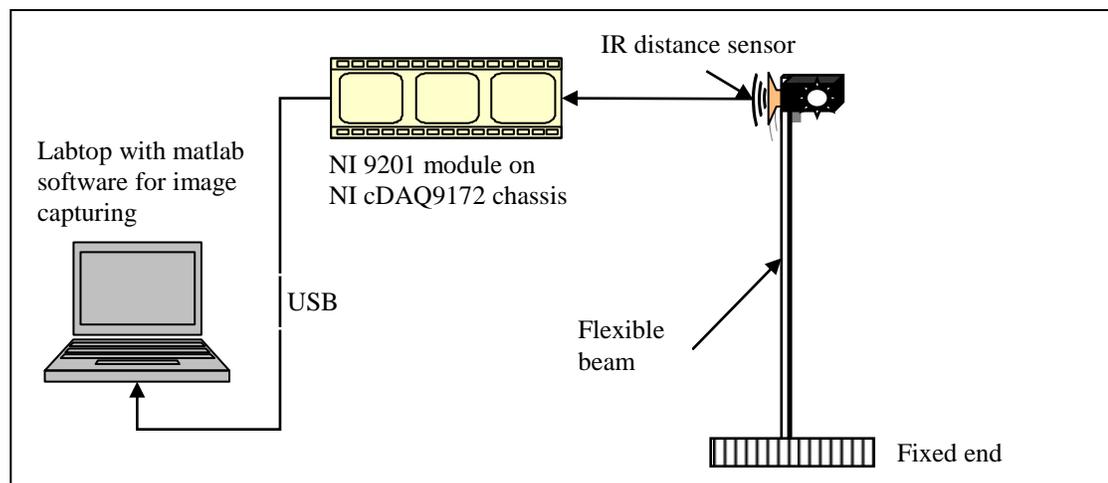
Fig 5.8 illustrates the flow chart of methodology illustrating the steps in carrying out tests on the flexible manipulator. The test preparation for the test is shown in Fig 5.9. Fig. 5.10 illustrates the schematics for the test setup for testing the flexible link beam. The test results and comparisons are depicted in section 6.1.



**Fig. 5.8** Methodology of simulation and testing of the flexible link manipulator for comparison



**Fig. 5.9** Setup for testing flexible link manipulator



**Fig. 5.10** Schematics of the test setup for vibration of flexible link manipulator

## 5.6 Methodology for comparing simulated vibration with actual vibration for the mobile flexible link manipulator

In order to test the accuracy of the model of mobile flexible link manipulator (MFLM), we set the methodology for comparing the simulated vibration at the tip of the MFLM with the actual vibration taken from the tests on MFLM rig. The tests will be carried out by setting various input voltage to the motor, and the results to be acquired are the platform velocity and the vibration at the tip of the flexible link manipulator. The methodology for carrying the test is as follows:

I. Setup: For the test setup, the flexible manipulator is mounted onto the mobile platform. A range sensor is fixed at the free end of the manipulator to sense the vibration at the free end. Refer to section 5.6.1 for detailed descriptions of the setup.

II. Vibration test on the MFLM:

1. Set no payload at the tip of the manipulator.
2. Set the direction of travel: Straight line forward path.
3. Set the time of recorded travel: Record only for 10 seconds.
4. Set repeating with the input voltages to the driving motor: 0.67V, 1.34V, 2.01V, 2.68V and 3.35V. The ranges of input voltages are due to the output of the programmed microcontroller. Refer part I of section 5.6.2 for the explanation for input voltages and the tests.

III. Vibration simulation of the MFLM: With the setting of zero for payload, the MFLM was tested with input voltages of 0.67V, 1.34V, 2.01V, 2.68V and 3.35V. (Refer to part II of section 5.6.2 for the explanation of the simulations).

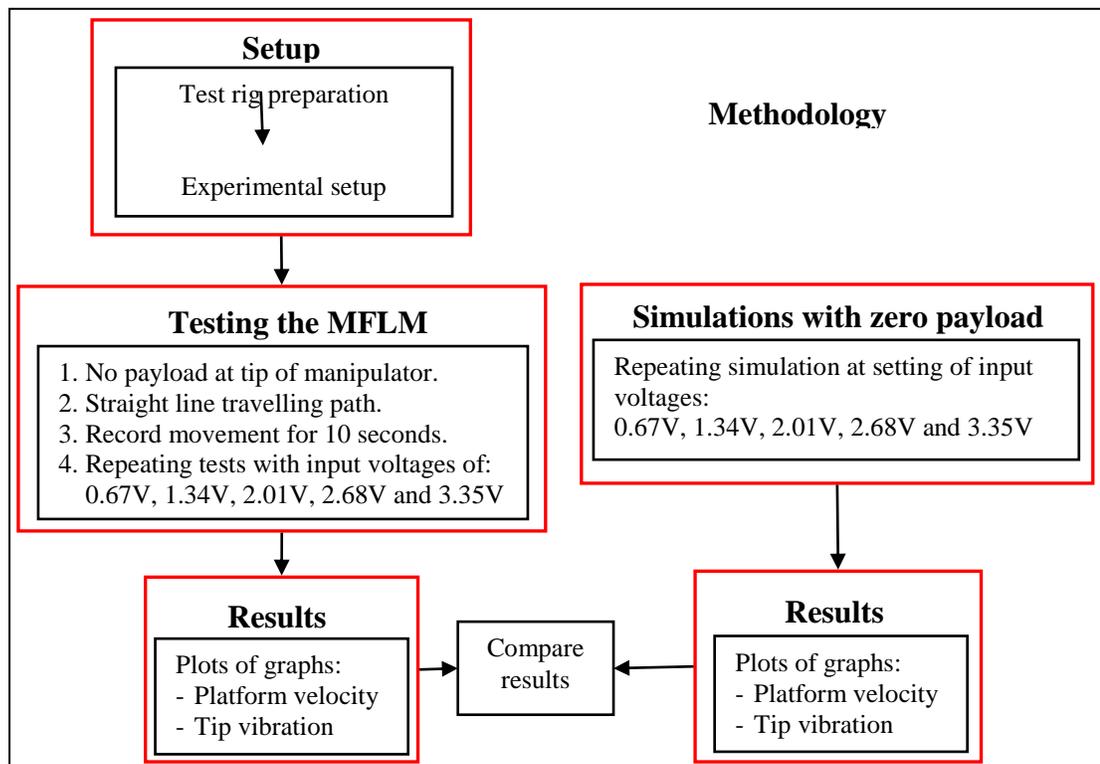
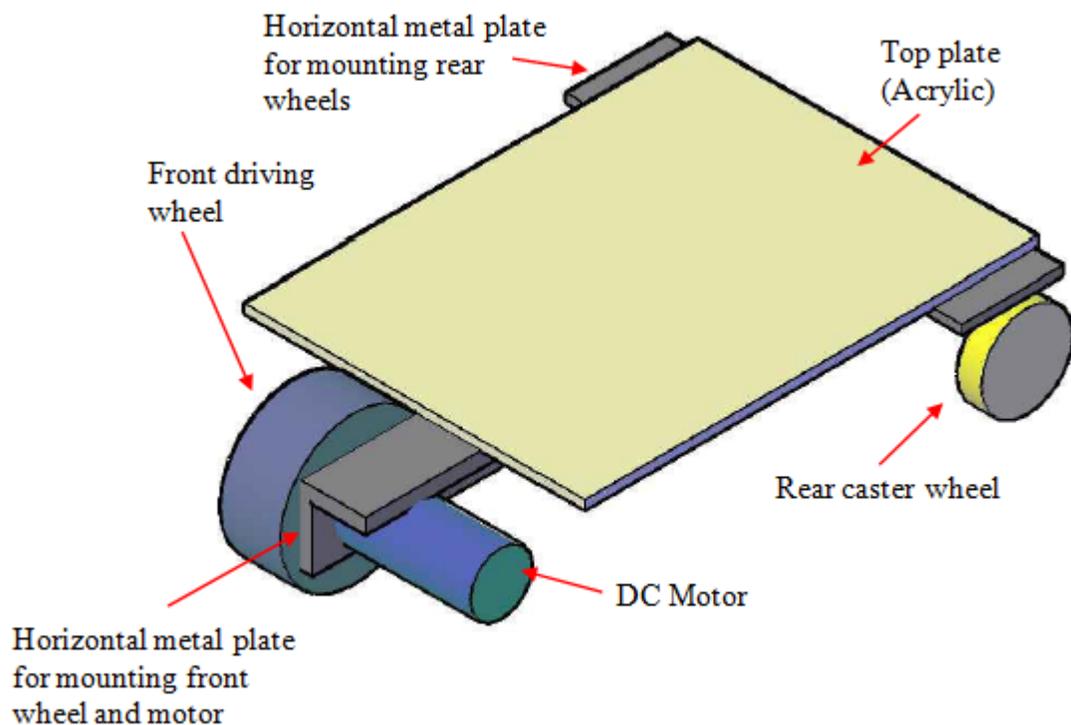


Fig. 5.11 Methodology of simulation and testing vibration of the MFLM

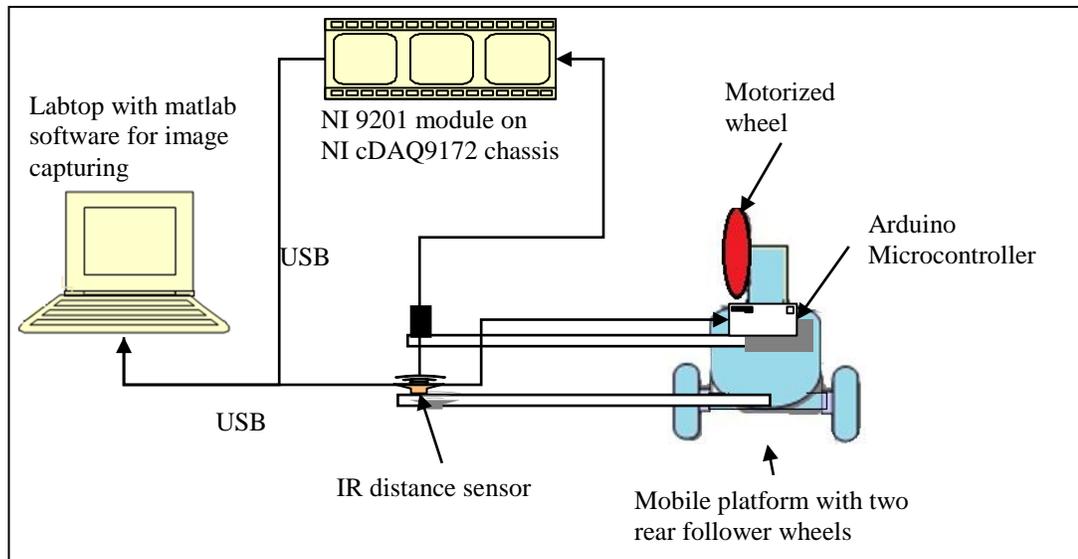
### 5.6.1 Test rig setup for the mobile flexible link manipulator

The mobile base consists of a rectangular piece fabricated using acrylic material, seated on three wheels. Two horizontal metal plates (one for mounting rear wheels, and one for mounting front wheel and motor) are mounted to the top acrylic plate. Two rear free rolling wheels, and a front wheel driven by a motor. The isometric drawing for the mobile base is depicted in Fig. 5.12 below. The base was fabricated in the mechanical workshop in the University of Nottingham (Malaysia campus). The flexible manipulator exhibited in Fig. 5.4 is then mounted onto the mobile base.

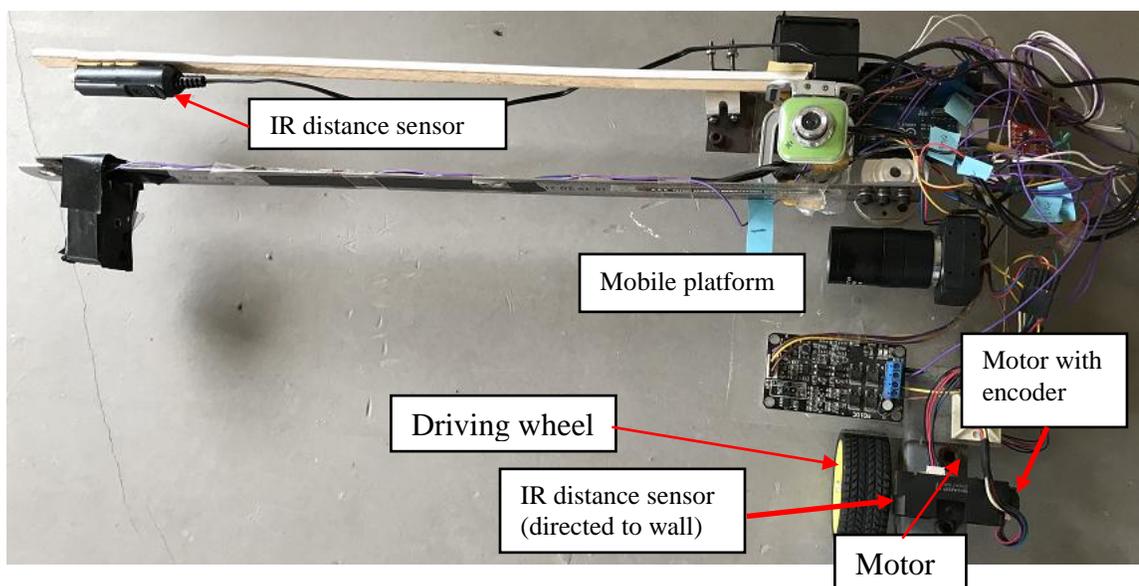
The schematics of the test setup of the flexible link manipulator on mobile platform is illustrated in Fig. 5.13. The physical setup of the complete test rig is shown in Fig. 5.14. The short range IR distance sensor (GP2Y0A21YK), fixed at the proximity to the tip of the flexible manipulator, and vibration signal acquired via NI 9201 module.



**Fig. 5.12** Isometric view of the mobile platform



**Fig. 5.13** Schematics of the test setup for vibration of mobile flexible link manipulator



**Fig. 5.14** Mobile flexible manipulator setup

## 5.6.2 Tests and simulations on the mobile flexible link manipulator

### I. The tests on the mobile flexible link manipulator

Tests were carried out with various step voltage inputs to the motor of the mobile platform to study the response of the flexible manipulator. The tests carried out in this section only show preliminary results for the responses of the mobile platform and the

flexible manipulator when subjected to input voltage supplied to the motor of the driving wheel.

The voltage input to the motor of the driving wheel is via the motor amplifier board (enhanced 10 Amp DC motor driver, MD10C), the voltage supply to MD10C is 6.03 V. An Arduino microcontroller sends the control signal to the MD10C, which is programmed for the output selection of 0 to 9, where 0 is 0 V and 9 is 6.03 V. (Refer to Appendix E.1 for the Arduino code).

All tests were carried out with 10 seconds of step input voltage. Due to the space constraint, the distance limitation of the range sensor and the limited length of the cable, the tests can only be carried until 3.35 V. This voltage input resulted in around 10 m/min speed (0.17 m/s). It is unsafe to the user for the domestic robots to travel above 10 m/min speed. It is thus not necessary to perform tests above 10 m/min (0.17 m/s) speed. Therefore, the MFLM of Fig. 5.14 will to be tested with the input voltages of 0.67 V, 1.34 V, 2.01 V, 2.68 V and 3.35V.

## **II. The simulations on the mobile flexible link manipulator**

Before developing the simulation, additional information about the MFLM is required as tabulated in Table 5.3 below. The motor's model of equation (3.48) was used for the simulations using the input voltage of 0.67V, 1.34V, 2.01V, 2.68V and 3.35V, respectively. To calculate the initial deflection of the beam during when the mobile platform accelerates from 0 to constant velocity, we assumed that the acceleration resulted as in force acting at the end of the beam. Thus, we can use equation (3.102) and (3.130) to compute the beam's deflection. The simulations ignored the disturbances contributed by the umbilical cable and possible backlashes of the geared motor that driving the front driving wheel.

The comparisons of the test and simulation results are presented in section 6.2 in the result chapter.

**Table 5.3** Parameters of the mobile flexible link manipulator

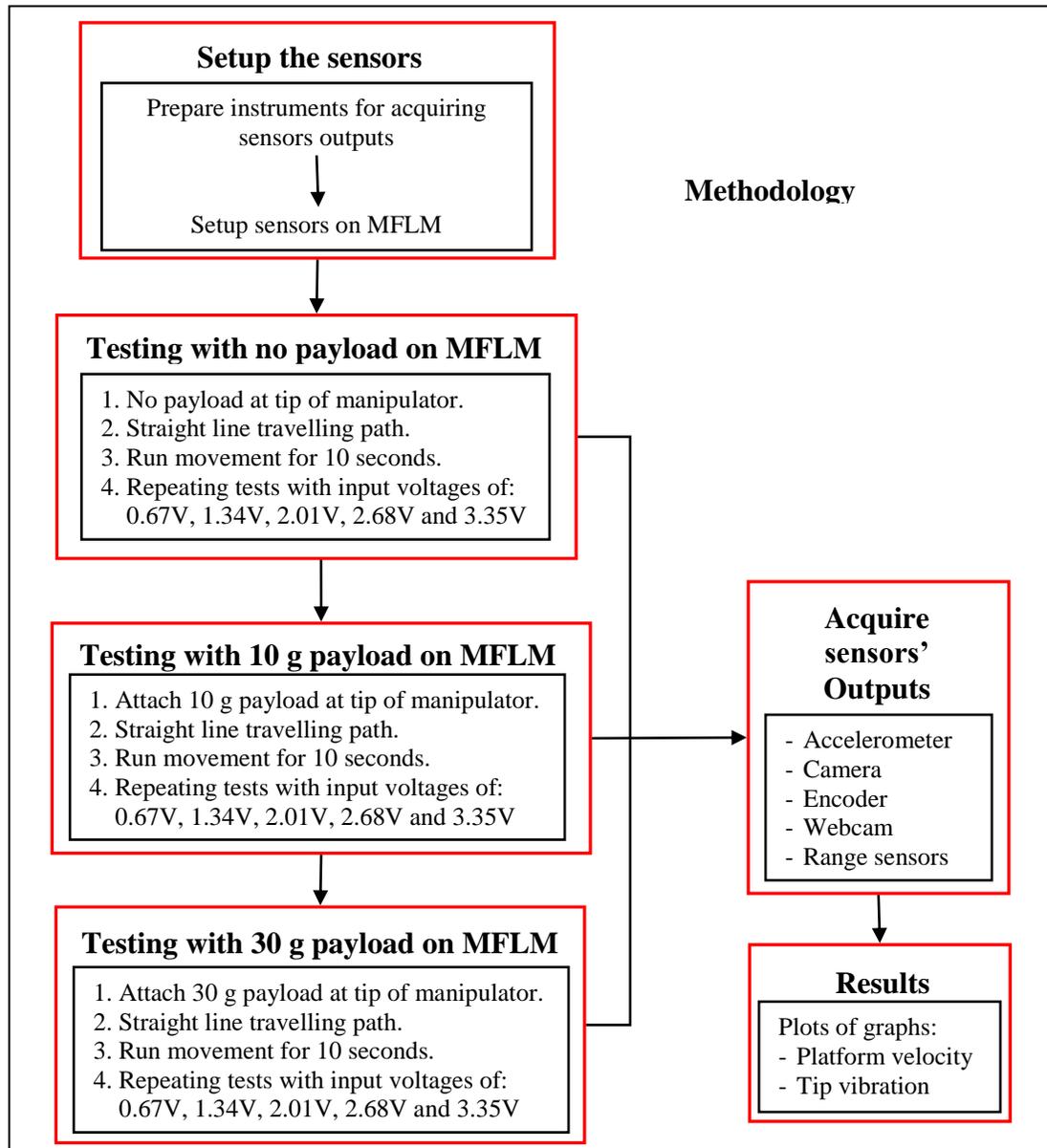
Parameter	Value
Front wheel's diameter, $D_f$	65 mm
Caster wheel's diameter, $D_w$	20 mm
Caster wheel's axle diameter, $D_a$	4 mm
Coefficient of friction for driving wheel, $\mu_d$	0.086
Coefficient of friction for caster wheel, $\mu_a$	0.03
Total mass of mobile platform	1.3072 kg
Mass of manipulator	107.133 g
Length between front and rear wheels, $l$	0.23 m
Length from centre of gravity to rear wheel, $x_1$	0.09 m
Length from centre of gravity to front wheel, $x_2$	0.14 m
Height from ground to centre of gravity, $h$	0.045 m

## 5.7 Methodology for tests on the mobile flexible link manipulator with sensors

In this section, the tests on the mobile flexible link manipulator (MFLM) are set. Test results will be obtained from output of the sensors that are mounted on the MFLM. The purpose is to investigate the capability of the individual sensors in tracking the vibration of the MFLM. The methodology for carrying out the test is as follows:

- I. Setup: Prepare the accelerometer, camera, webcam, encoder and IR distance sensors to mount onto the MFLM. Refer to section 5.7.1 on the details of setting up the sensors on the MFLM.
- II. Vibration test on the MFLM with no payload:
  1. Set no payload at the tip of the manipulator.
  2. Set the direction of travel: Straight forward path of travel.
  3. Set the time of recorded travel: Run only for 10 seconds.
  4. Set repeating with the input voltages to the driving motor: 0.67V, 1.34V, 2.01V, 2.68V and 3.35V. (The ranges of input voltages are due to the output of the programmed microcontroller. Refer to second paragraph of section 5.6.2 for the explanation for input voltages and the tests).
- III. Repeat the test in step II using 10 g payload.
- IV. Repeat the test in step II using 30 g payload.

Refer to 5.7.2 for the detailed descriptions on the tests carried out. The methodology for carrying out the test is depicted in Fig. 5.15 below.

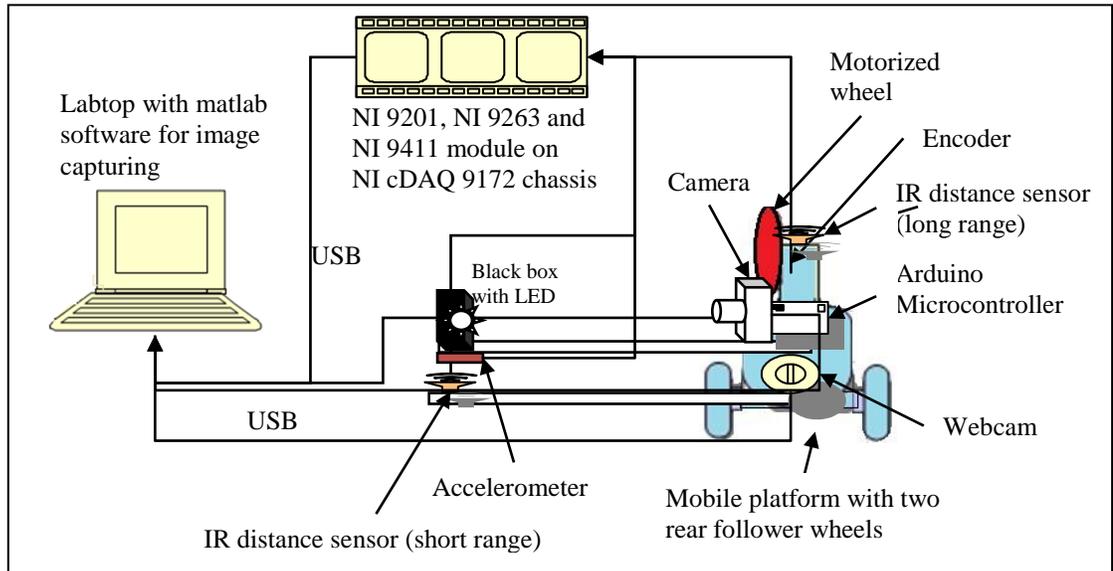


**Fig. 5.15** Methodology for acquiring sensor output for vibration estimation of the MFLM

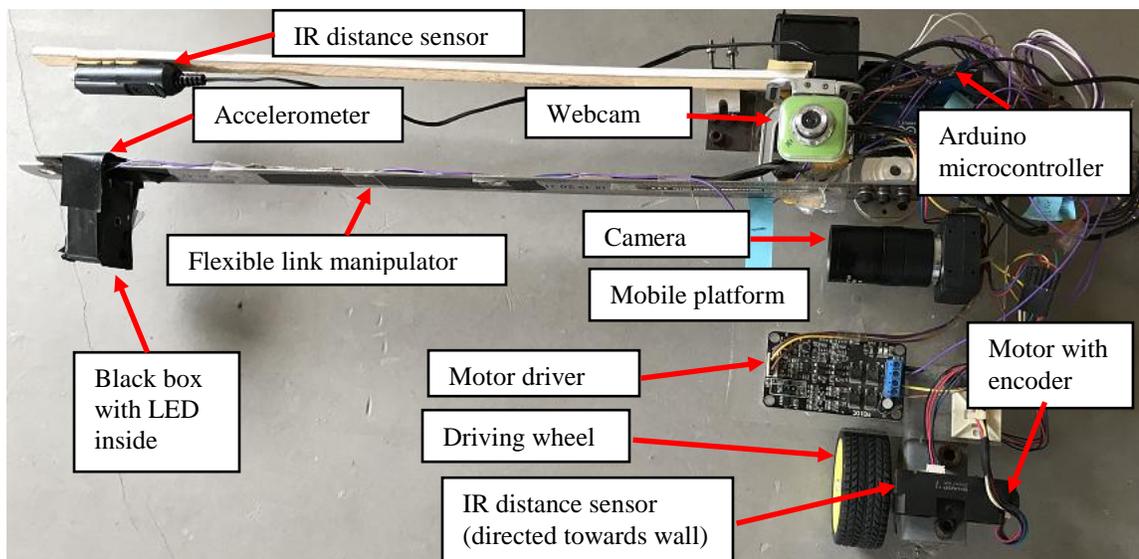
### 5.7.1 Setup for mobile flexible link manipulator with sensors

The configuration of the mobile flexible link manipulator considered in this work is based on a single link flexible beam mounted on mobile platform, as shown in the

schematics of Fig. 5.16 below. This entire system together composes superposition of nonlinearities, with high degree of vibration of the flexible beam when the platform moves.



**Fig. 5.16** Schematics of the test setup for vibration of mobile flexible link manipulator with all sensors (CMOS camera, webcam, accelerometer, encoder and IR distance sensors) attached



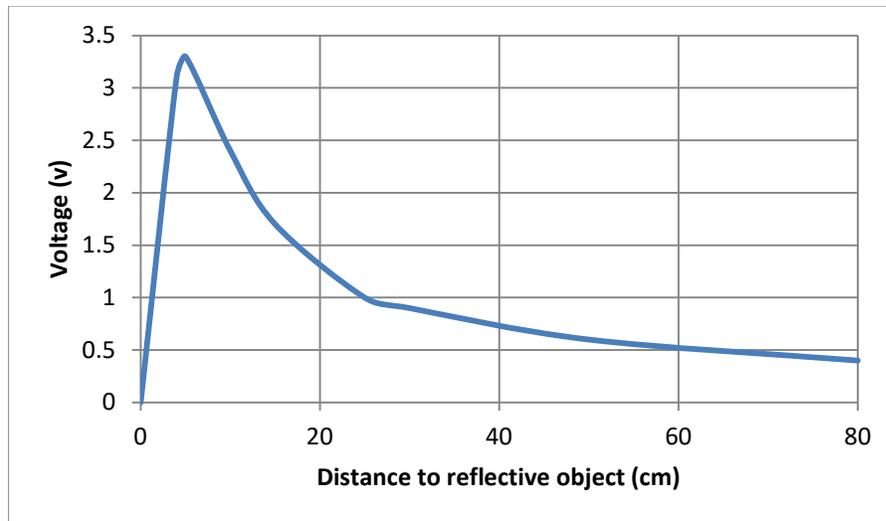
**Fig. 5.17** Setup of mobile flexible link manipulator with sensors

As shown in Fig. 5.17, the sensors consists of: 1) accelerometer mounted at the backside of the black box; 2) CMOS camera to capture the image of the LED inside black box; 3) webcam for capturing the black marking at the ceiling; 4) encoder that comes with the motor; and IR distance sensor to sense the tip of the flexible beam. The accelerometer and camera are used to capture the vibration of the flexible link manipulator, while webcam and encoder are used to the estimate the velocity of the mobile platform.

The accelerometer is MMA7260Q (refer to section 3.1.3 for detailed description), and the signals is acquired through National Instruments NI 9201 and powered by NI 9263 on NI eDAQ9172 chassis. The encoder is a rotary encoder (refer to section 3.3.3 for detailed description) with 2248.8 pulses per revolution, and the signal is acquired through NI 9411 on NI eDAQ 9172 chassis. The data acquisition rate was sampled at 11 KHz so as to obtain the best resolution of the signals and no aliasing.

The Webcam is Ultra-HD PC video webcam and the camera is a Firefly series FMVU-03MTC-CS USB CMOS camera (refer to section 3.2.3 for detailed descriptions). CMOS camera has a frame rate of 57 – 60 fps (frame per second), that frames a single LED fixed at 50 cm away from the CMOS camera. Calibration discovered that it has 1 mm accuracy (an averaged scale factor of 1 pixel/mm) measuring at 50 cm distance from the LED. In order to achieve the aforementioned frame rate, the acquired image has been reduced from  $640 \times 480$  to  $430 \times 64$  pixels, corresponding to the arc that the LED describes on the image plane. Timestamp exchange is used to predict the delayed visual data from the camera.

For image processing operation, a blob detection operation is used to evaluated the coordinates of the LED on the image plane. An algorithm has been implemented to predict the next LED position, so that only a  $100 \times 20$  rectangular portion of the image is analyzed each cycle, which is the distance from the current LED position to the next LED position for highest speed. This saves computational burden and image processing time. The acquisition rate for webcam is set at its maximum of 30 fps and the camera acquisition rate is set at its maximum of 60 fps.



**Fig. 5.18** Output voltage VS distance to reflective object for infrared sensor

A high accuracy short range infrared distance measuring sensors is used to measure the horizontal displacements at the tip of the vibrating flexible beam, while a long range infrared distance measuring sensor was used to measure the distance travelled by the mobile platform. Short distance infrared measuring sensor (Sharp GP2Y0A21YK) is used to measure the vibration at the tip of this manipulator. This infrared sensor can measure range between 0 cm and 80 cm. However, its linear characteristic is only up to the maximum of 4 cm. Above 4 cm it exhibits an opposite and non-linear displacement to voltage output. Fig. 5.18 shows the distance vs voltage output plot for this sensor. So for a linear accuracy, the sensor is best mounted within 4 cm to the manipulator. The data acquisition instrument is NI 9219 on NI DAQ 9172 chassis, and the data acquisition rate was set at 25 KHz.

A long distance infrared distance sensor (Sharp GP2Y0A710K) was used to measure the horizontal motion of the mobile platform. This long distance infrared sensor can measure distances between 1 m to 5.5 m. It was mounted at the heading direction of the mobile platform and pointed horizontally towards the wall.

### **5.7.2 Tests on the mobile flexible link manipulator**

The MFLM could have the following systematic and non-systematic uncertainties:

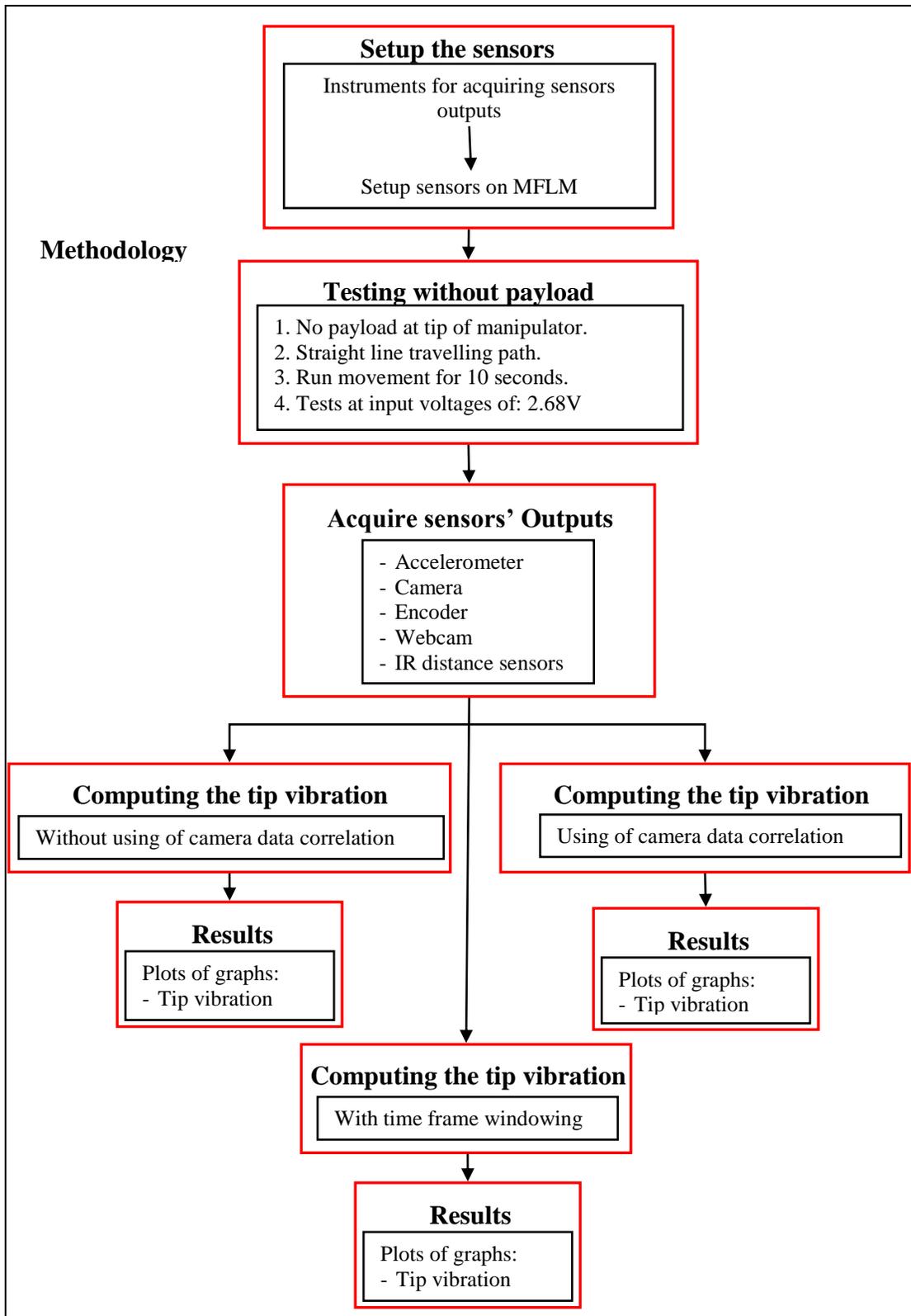
1. Variable deformations at the links due to flexible beam.
2. Backlashes at the gears of the motor.
3. Jerking motor when driving the entire weight of the MFLM.
4. External forces (the umbilical cables).
5. Friction from follower wheels.
6. Overall weight of the whole platform and manipulator.
7. Encoder mis-count.

The tests carried out were for a longitudinal path of the mobile platform for 10 seconds and stop. For the same reason discussed in section 5.6.2 that the input voltage is programmed by the microcontroller for the selection of 0 to 9, the experiments were carried out with 0.67V, 1.34V, 2.01V, 2.68V and 3.35V only due to space constraint. All experiments were carried out with 10 seconds of step input. Then, the same tests were carried out on the MFLM with payload of 10 g and then with payload of 30 g attached at the tip of the manipulator.

The results for the validation tests as described above are presented in section 6.3 in the results chapter.

## **5.8 Validation tests with sensor fusion algorithm**

In this section, the five types of EKF data fusion algorithms as established in the section 5.2 are compared for vibration tracking at the tip of the flexible manipulator. The validation tests are run under various input voltages to the driving motor of mobile platform and with various loads at the tip of the flexible manipulator. The objective of the tests is to command the MFLM to move along a straight line and to validate the accuracy of the system in computing the longitudinal vibration at the tip of the mobile flexible link manipulator.



**Fig. 5.19** Methodology for acquiring and computing the vibration of the MFLM

The methodology for the tests is set out as follows:

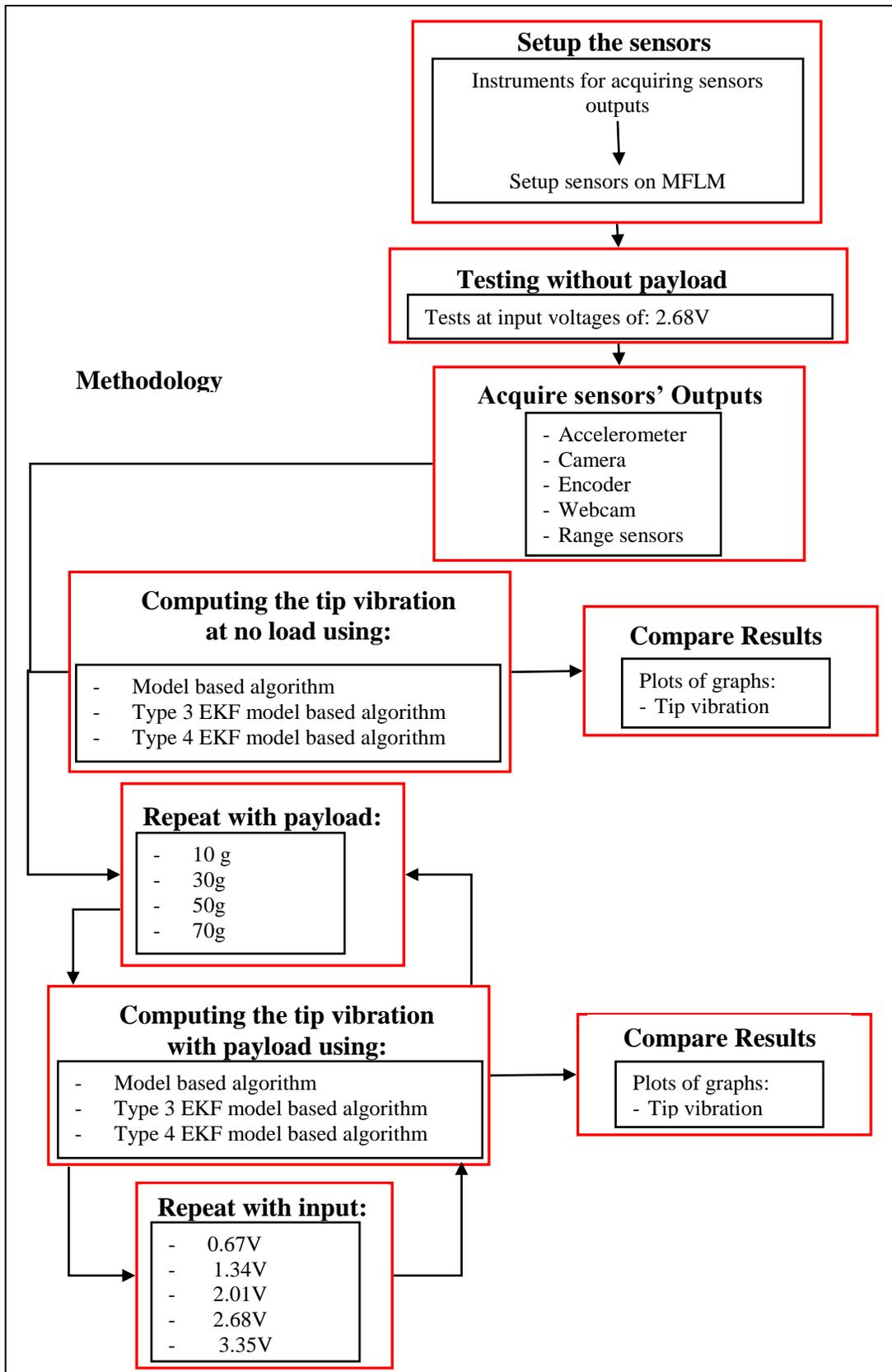
- I. Setup: the setup in section 5.7.1 above is used for carrying out the tests.
- II. Vibration test on the MFLM with no payload:
  1. Set no payload at the tip of the manipulator.
  2. Set the direction of travel: Straight line forward path.
  3. Set the time of recorded travel: Run only for 10 seconds.
  4. Set repeating with the input voltages to the driving motor: 0.67V, 1.34V, 2.01V, 2.68V and 3.35V. (Note that only the result for 2.68V is presented as the results are significant enough to represent results of other input voltages).
- III. Testing with date correlation:
  1. Tests without camera data correlation.
  2. Tests with camera data correlation.
- IV. Testing with time frame windowing

The flow chart for the methodology is shown in Fig. 5.19 above. The MFLM setup with the sensors setup as in section 5.7.1 is used to perform the tests, but only tested using 2.68V step input. The tests results are presented in section 6.4.

## **5.9 Methodology for validating predictive algorithm on the MFLM**

Experimental validations are carried out to compare three different predictive algorithms. The three predictive algorithms are model based algorithm, type 3 EKF model based algorithm and type 4 EKF model based algorithm. Type 3 EKF sensor fusion does not utilize modelled input, while type 4 EKF sensor fusion consists of modelled data in place of camera data and has modelled input. The predict-ahead point was set at 0.15 seconds ahead.

The validation tests are first carried out with no load at 2.68V input, and the three types of algorithms are compared. Then, the tests are carried out with the payload of 10g, 30g and 50g. The tests repeated at 0.67V, 1.34V, 2.01V, 2.68V and 3.35V.



**Fig. 5.20** Methodology for comparing the three types of predictive algorithms

The methodology for the validation tests are as follows:

- I. Setup: the setup as described in section 5.1.7 is used in carrying out the tests.
- II. Carry out the vibration tests on the MFLM with no payload:
  1. Set no payload at the tip of the manipulator.
  2. Set the direction of travel: Straight line forward path.
  3. Set the time of recorded travel: Run only for 10 seconds.
  4. Set input voltages to the driving motor: 2.68V.
- III. Compute the tip vibration using:
  1. Model based algorithm.
  2. Type 3 EKF model based algorithm
  3. Type 4 EKF model based algorithm
- IV. Repeat step II and III using the payload: 10g, 30g, 50g and 70g.
- V. Repeat step IV with input voltage: 0.67V, 1.34V, 2.01V, 2.68V and 3.35V.

The flow chart for the methodology to carry out the tests is shown in Fig. 5.20 above.

The validation test results are then depicted at section 6.5 in the results chapter.

## **5.10 Summary**

This chapter is separated into three parts. In part one, it started with the introduction to the Kalman filter, Extended Kalman filter and the cross correlation algorithms. It then followed by the development of the fusion algorithm, where five EKF types of fusion algorithms are established. The algorithms work such that when the camera's signal is available, the fusion takes the combination of accelerometer and camera signals. When the camera's signal is not available, the algorithms make use of accelerometer's signal to perform fusion activities.

The last portion of part one is the development of the predictive algorithm. The algorithm makes use of the previous states to predict the future states of the flexible manipulator vibration. MFLM modelled is fed as input for the future prediction.

Part two started with the building the flexible manipulator and its analysis. It was found that mode 1 vibration mode is sufficient for use in the assumed mode method of flexible

manipulator modelling. It then followed by defining a methodology for performing the tests for comparing simulated vibration with actual vibration of the flexible link manipulator (FLM) to certify the accuracy of the FLM model. Lastly, a methodology was defined for performing the tests for comparing simulated vibration with actual vibration of the MFLM. The simulation results are to be presented the next chapter (the result chapter). The rig of the entire MFLM has been fabricated.

Part three of this chapter present the methodology for performing the validation tests on the mobile flexible link manipulator with sensors. Methodology for testing the MFLM with sensors was first defined where the purpose is to present the tracking performance of the sensors. Next, the methodology for testing the sensor fusion algorithm was being defined using the same setup. The final part was defining the methodology for testing the predictive algorithm on the MFLM.

## **6 Performance of the predictive sensor system - results and discussions**

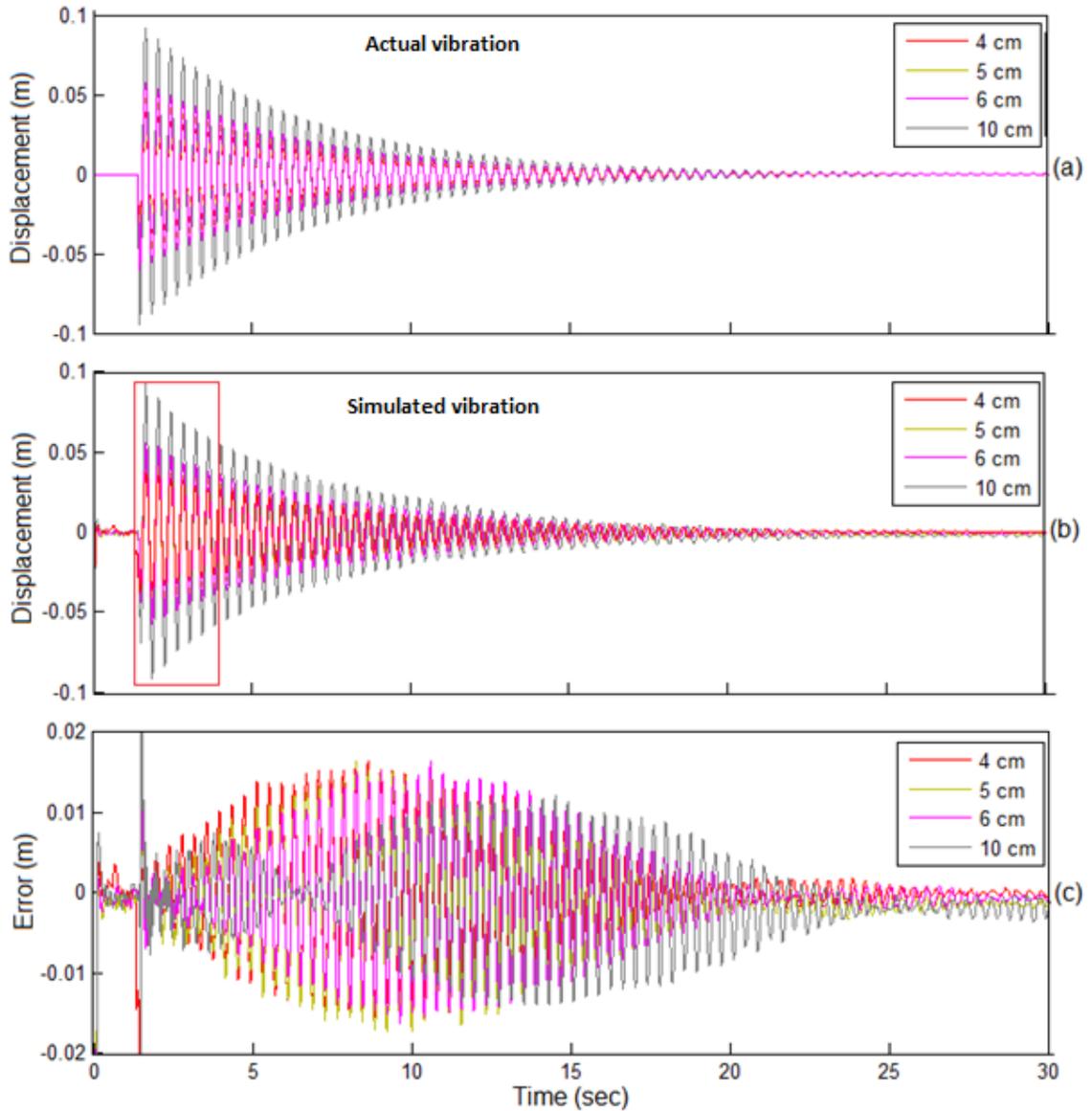
In this chapter, the data from the camera and accelerometer are put together for real-time prediction of the manipulator's vibrations. This chapter answers the research objective point 7 in section 1.2 of chapter 1. The purpose is to discover how the signals from each sensor can be used to improve the vibration tracking of the flexible manipulator when the sensors are put together? What algorithm should be used so that the system can adequately use the information and complement the errors from each sensor? The unmodelled uncertainties are accommodated by having a sensor system to provide accurate feedback.

The methodologies defined in chapter 5 are carried out in this chapter with experimental tests on the sensor fusion system being developed. There are two parts of the validation tests results presented here. First part presents the simulation results for the flexible manipulator and the mobile flexible link manipulator (MFLM), which are covered in sections 6.1 and 6.2.

The second part presents validation tests results for the mobile flexible link manipulator. The tests results on the sensors for tracking the vibration of the MFLM and the testing of the sensor fusion and prediction algorithms are presented, which are covered in sections 6.3 to 6.5.

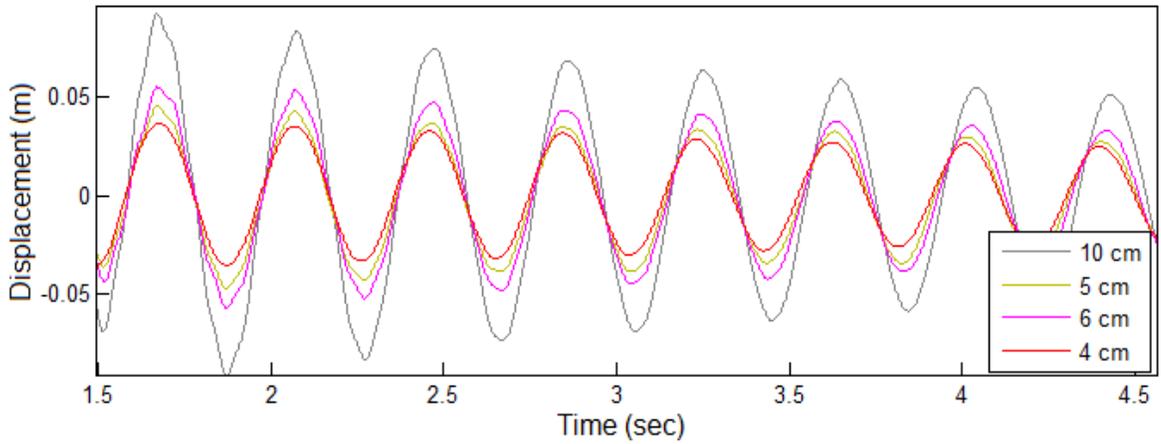
### **6.1 Test results for comparing simulated vibration to actual vibration of the flexible manipulator**

The results for the tests methodology set in section 5.5 are presented in this section. Fig. 6.1 exhibits the comparison of the modelled output with the actual beam's free vibration released at different initial displacements. Observations of Fig. 6.1c depicts that the error computed by the model increases after a few oscillations. The error is due to the gaining vibration frequency when the vibration amplitude diminishes overtime. This can be further explained in closed up view of Fig. 6.2 below.



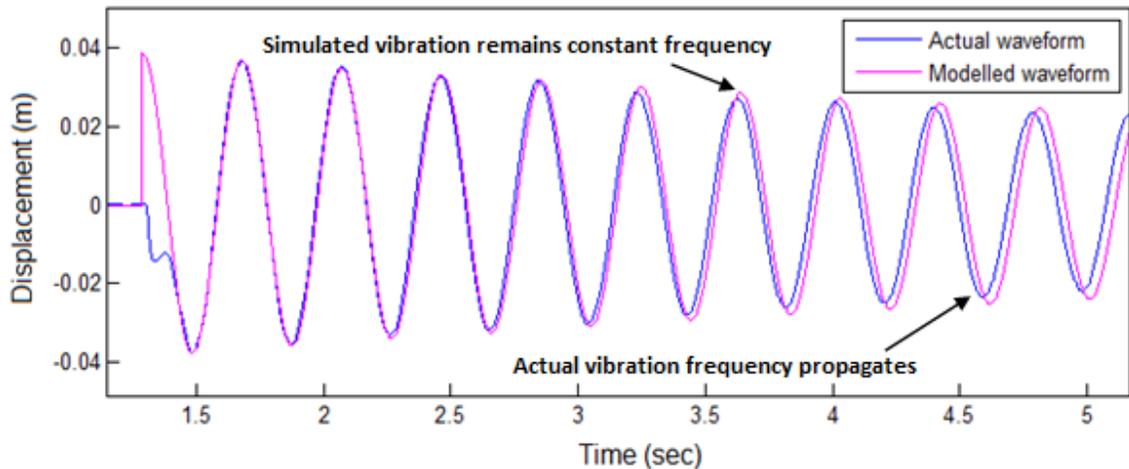
**Fig. 6.1** Comparison of actual displacement at 4 cm, 5 cm, 6 cm and 10 cm: (a) actual displacement, (b) model computed displacement, (c) error

Furthermore, a higher vibration frequency was observed when the beam was excited at lower deflection  $w$ . As it can be seen in Fig. 6.2, the 10 cm displacement initiation depicts a shift to the right as compared to the other displacement initiation. This is because 10 cm is more significant as compared to 4 cm, 5 cm and 6 cm, discovering the fact that at lower vibration amplitude the vibration frequency is higher.



**Fig. 6.2** Closed up view for Fig. 6.1b, showing higher frequency when the beam at lower amplitudes

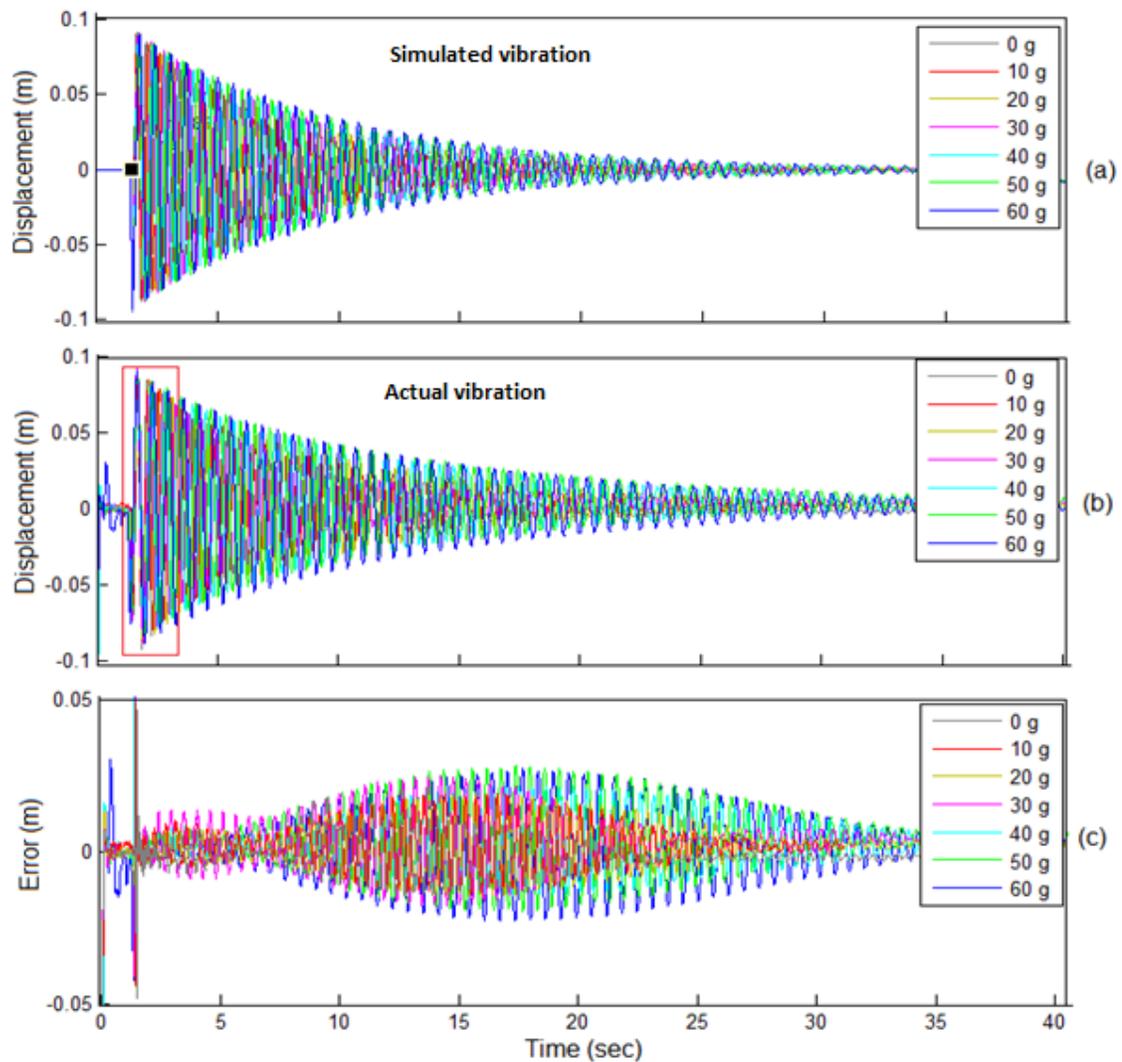
As illustrated in Fig. 6.3, a good match for the first five cycles, but then shows mismatching of the phase during the subsequent cycles. This is due to the fact that as the actual vibration amplitude diminishes the vibration frequency propagates and vibrates faster at small amplitude, whereas our simulated vibration frequency remains constant for the duration of the vibration simulation as the vibration diminishes.



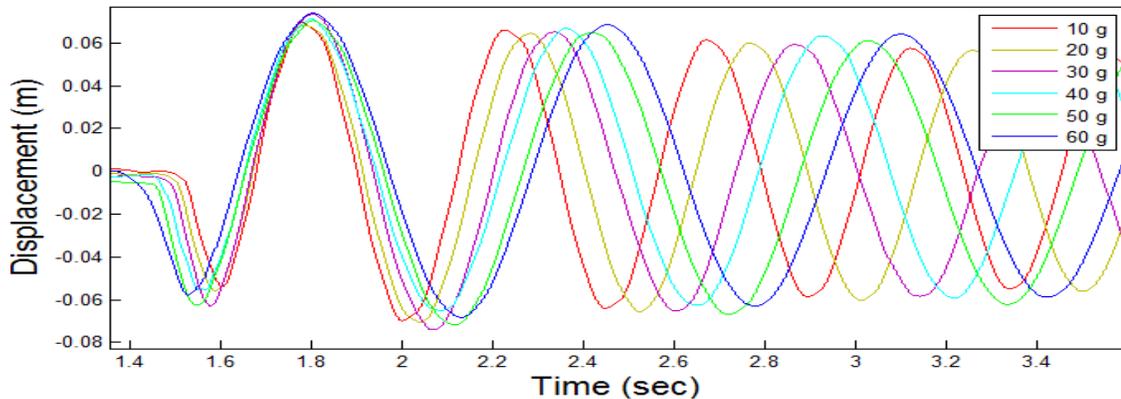
**Fig. 6.3** Comparison of response between actual displacement and modelled displacement

Validation tests were then carried out with the payload attached to the tip of the beam, using the payloads of 10 g, 20 g, 30 g, 40 g, 50 g, 60 g and no load, excited with

deflection  $w$  of 8 cm. It can be seen that the error increases up to 0.025m amidst the vibration duration due to the increasing vibrating frequency, as depicted in Fig. 6.4. Fig. 6.5 shows that higher payloads result in lower vibration frequencies. The validation analysis thus showed fairly good match between the model output and actual output, except for the small vibration amplitude. The reason could be due to the varying beam's damped and undamped natural frequencies,  $\omega_d$  and  $\omega_n$  terms which are assumed constant in our model. Nevertheless, the accuracy of this model is sufficient for use in the model predictive estimation of the beam vibration for 5 seconds prediction.



**Fig. 6.4** Comparison of actual displacement at 0 g, 10 g, 20 g, 30 g, 40 g, 50 g and 60 g payloads:(a) modelled displacement, (b) actual displacement, (c) error.



**Fig. 6.5** Enlarged section for Fig. 6.4b, showing lower vibration frequencies at higher payloads

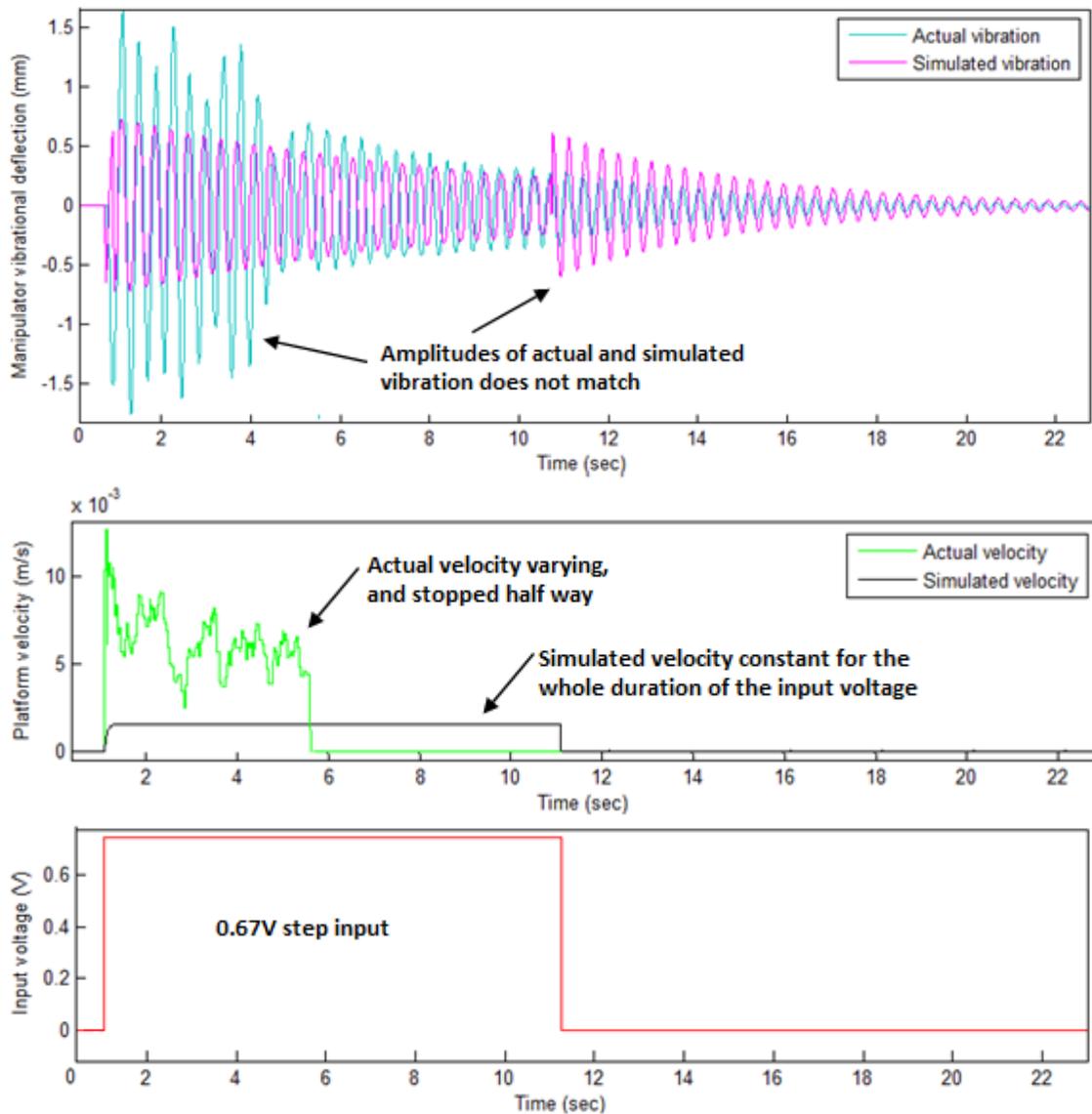
## 6.2 Results for comparing simulated vibration to actual vibration of the mobile flexible link manipulator

The comparison results for the test methodology set in section 5.6 are presented in this section. Fig. 6.6 to 6.10 depicts the comparisons of the outputs of the MFLM with the simulation results. It can be seen that the simulation results of the vibration of the flexible manipulator and platform speed does not match accurately with the actual results. The frequency of the vibration by the simulation does match with the actual, whereas the amplitude of vibration does not match. At the set input voltage, it can be seen that the actual velocity of the platform is varying as it moves. These are due to systematic and non-systematic uncertainties such as: 1) umbilical cable pulling the platform, 2) Backlashes at the gears, 3) unbalanced weight of the whole platform and manipulator; 4) Jerking motor when driving the entire weight of the MFLM; 5) Internal forces from the rear free wheels; and 6) platform accelerations; 7) encoder mis-count. The experimental tests with the various input voltages are as follows:

### 6.2.1 Tests with 0.67 V step input voltage

Referring to Fig 6.6, at 0.67 V step input, the simulated velocity for the robot is 1.5 mm/s. But, but the actual velocity spiked to 10 mm/s and then maintained an inconstant velocity at around 5 mm/s, and then stopped at 5.6 seconds. The vibration at the manipulator's tip held around 1 to 1.5 mm for 4.4 seconds and then diminishes to zero.

The simulated manipulator's vibration responded according to the simulated velocity, thus does not match to the actual vibration of the manipulator.



**Fig. 6.6** Flexible manipulator output and platform velocity for 0.67 V input voltage

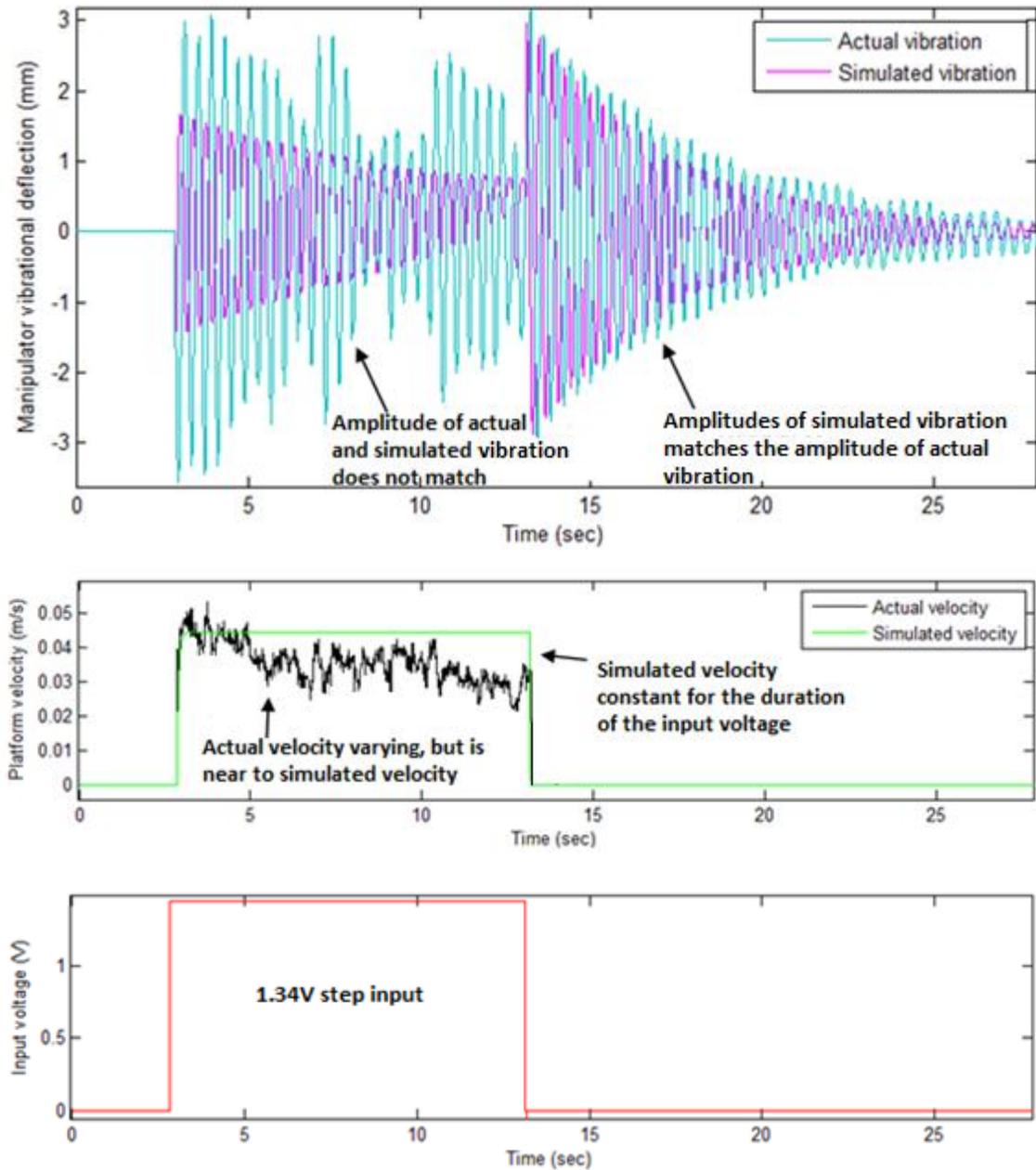
The reaction of the mobile base towards the 0.67 V is due to the weight of the whole robot. The input of the 0.67 V to the driving motor can only drive the front wheel to move for 5.6 seconds, and then came a complete halt. At 4.4 seconds, the motion input from the mobile platform to the base of the flexible manipulator has stopped, even though continued for another 1.2 seconds before came a complete halt. The vibration of

the flexible manipulator thus started to diminish to zero from 4.4 seconds onwards. Our mathematical model failed to account for this phenomenon, hence the simulated velocity of the mobile platform and vibration of the flexible manipulator does not match the actual behaviours of the mobile manipulator.

### **6.2.2 Tests with 1.34 V step input voltage**

Referring to Fig 6.7, at 1.34 V step input, the simulated velocity is 0.043 m/s. It can be seen that actual velocity of the robot varies around 0.043 m/s for ~2 seconds, and then varies around 0.035 m/s for the remaining 1.34 V input until it stopped. This phenomenon of the velocity could be attributed to the umbilical cable pulling the platform; the entire weight of the robot; and jerkiness due to insufficient driving input voltage.

The actual vibration at the tip of the manipulator reacted according to the motion of the entire platform. This vibration behaviour does not match the simulated vibration, which simulated to react according the constant velocity of the platform. The actual and the simulated vibration of the platform matched when the robot stopped, which happened after the 13<sup>th</sup> seconds. This means that our mathematical equations accurately modelled the vibration of the manipulator when the robot stopped. The failure to accurately model the velocity of the platform attributed to the failure to model the actual vibration at the tip of the manipulator.

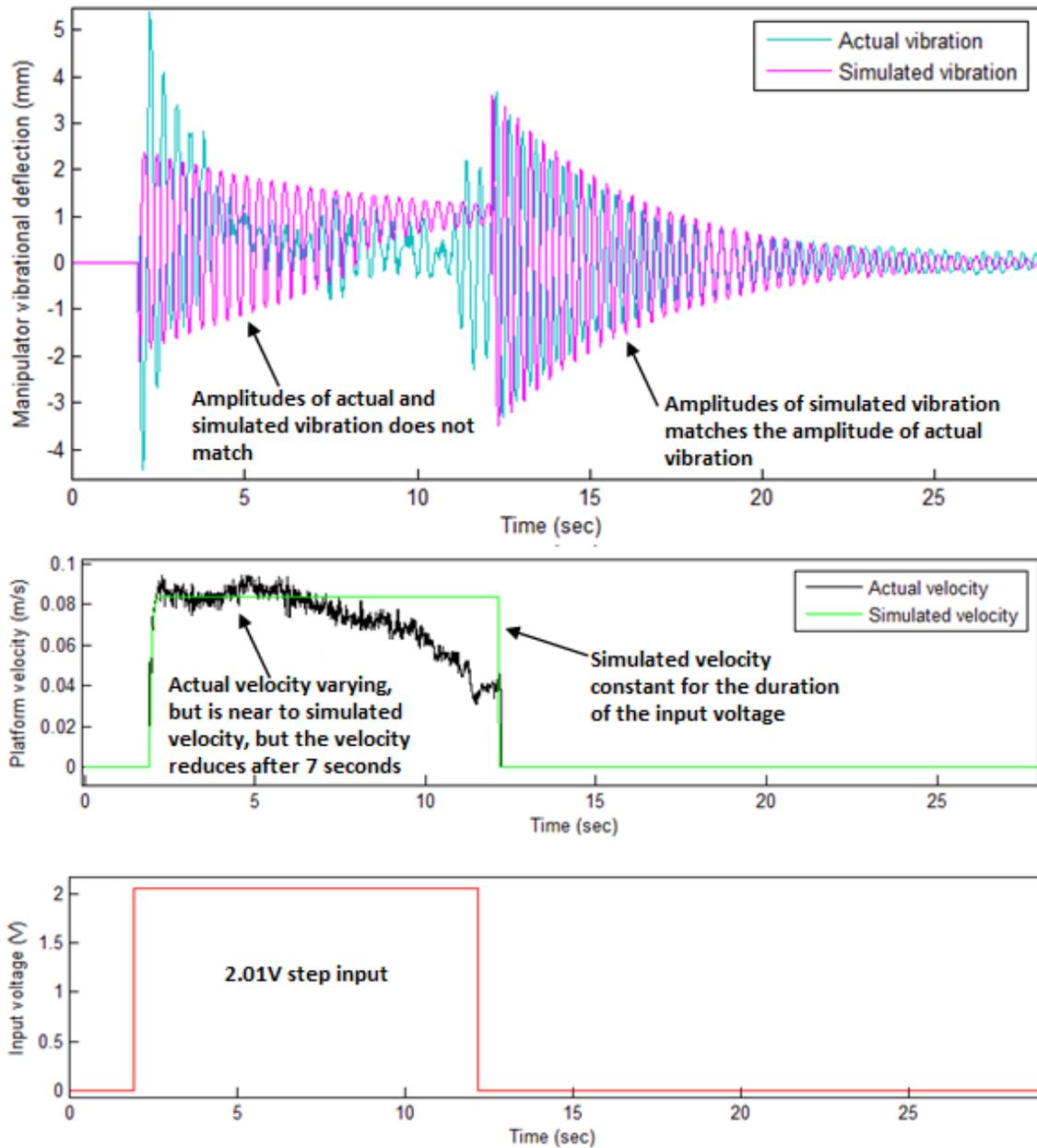


**Fig. 6.7** Flexible manipulator output and platform velocity for 1.34 V input voltage

### 6.2.3 Tess with 2.01V step input voltage

Referring to Fig. 6.8, at 2.01 V step input, the simulated velocity is 0.08 m/s. The actual velocity fluctuating around 0.08 m/s around 5 seconds and then gradually dropped to 0.04 m/s before it stopped. Similarly, it can be explained here that the behaviour of the velocity dropping is attributed by the umbilical cable pulling the platform and the weight of the robot that caused gradual slowing down of the mobile platform.

The actual and simulated vibration of the manipulator's tip does not match during the movement of the mobile robot. The simulated and the actual vibration of the manipulator match when the platform stopped, which indicates our mathematical model for the flexible manipulator is accurate.



**Fig. 6.8** Flexible manipulator output and platform velocity for 2.01 V input voltage

## 6.2.4 Tests with 2.68 V step input voltage (this test included unexpected halt disturbance)

Referring to Fig. 6.9, at 2.68 V step input, the simulated velocity is 0.13 m/s. In this test, a halt disturbance was created to show the reaction of the system. The actual velocity, due to the step input of 2.68 V, resulted in a sudden acceleration to 0.15 m/s. It then slows down gradually to 0.09 m/s due to the umbilical cable pulling the platform and the entire weight of the robot.

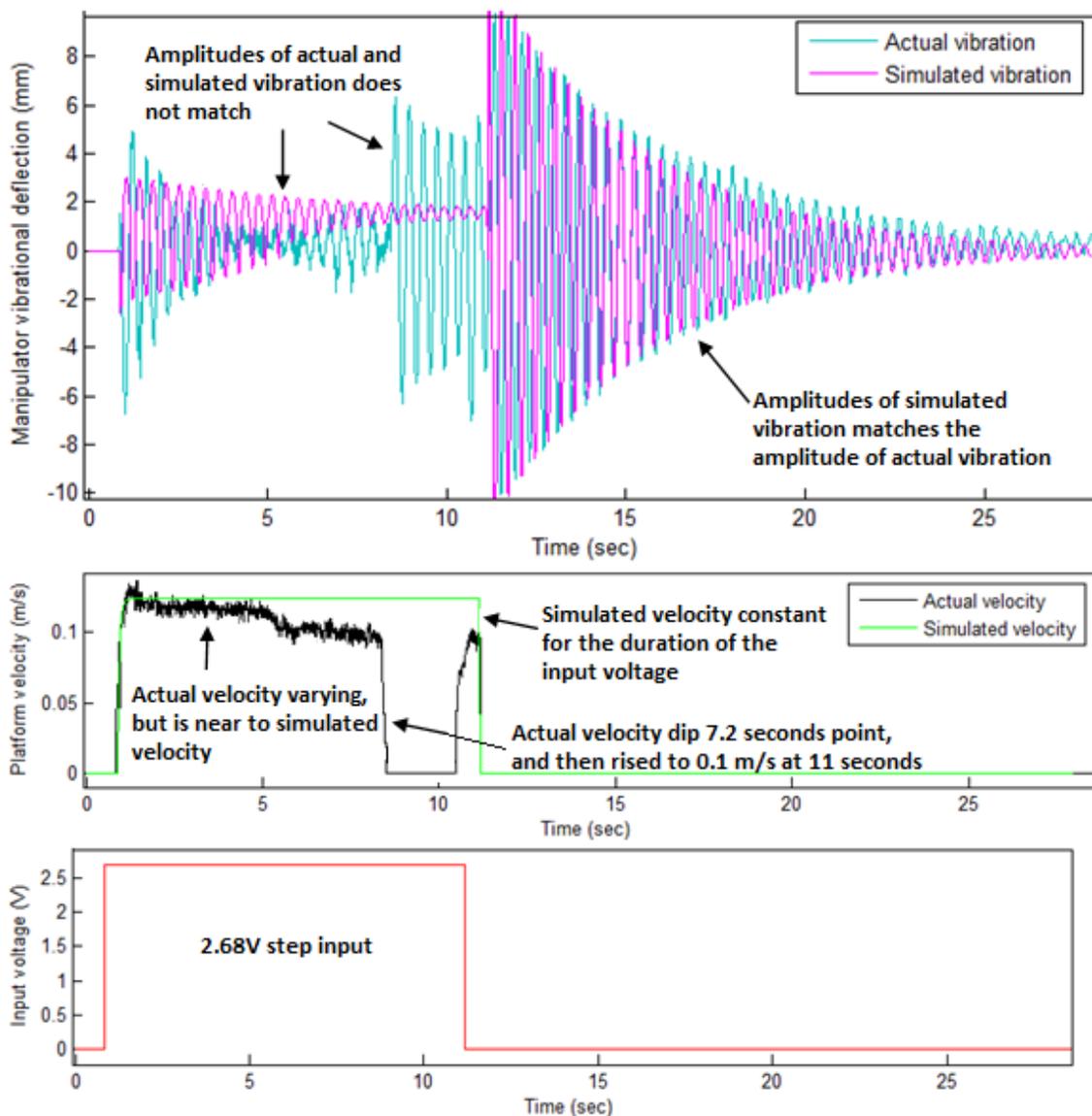


Fig. 6.9 Flexible manipulator output and platform velocity for 2.68 V input voltage

An unexpected disturbance caused the robot to standstill for 4 seconds at the 8<sup>th</sup> second time. This disturbance was not being detected by the model, thus the simulation failed to simulate the halt. The actual velocity then surged to 0.09 m/s then the disturbance was released, and stopped due to end of input voltage.

The vibration simulation of the flexible manipulator in the duration of the moving platform did not match the actual vibration due to the unstable velocity of the platform, which attributed to the unpredictable vibration. When the platform stopped, the simulated the actual tip vibration of the manipulator matches quite, showing a good accuracy of the model.

### **6.2.5 Test with 3.35 V step input voltage (this included unexpected disturbances)**

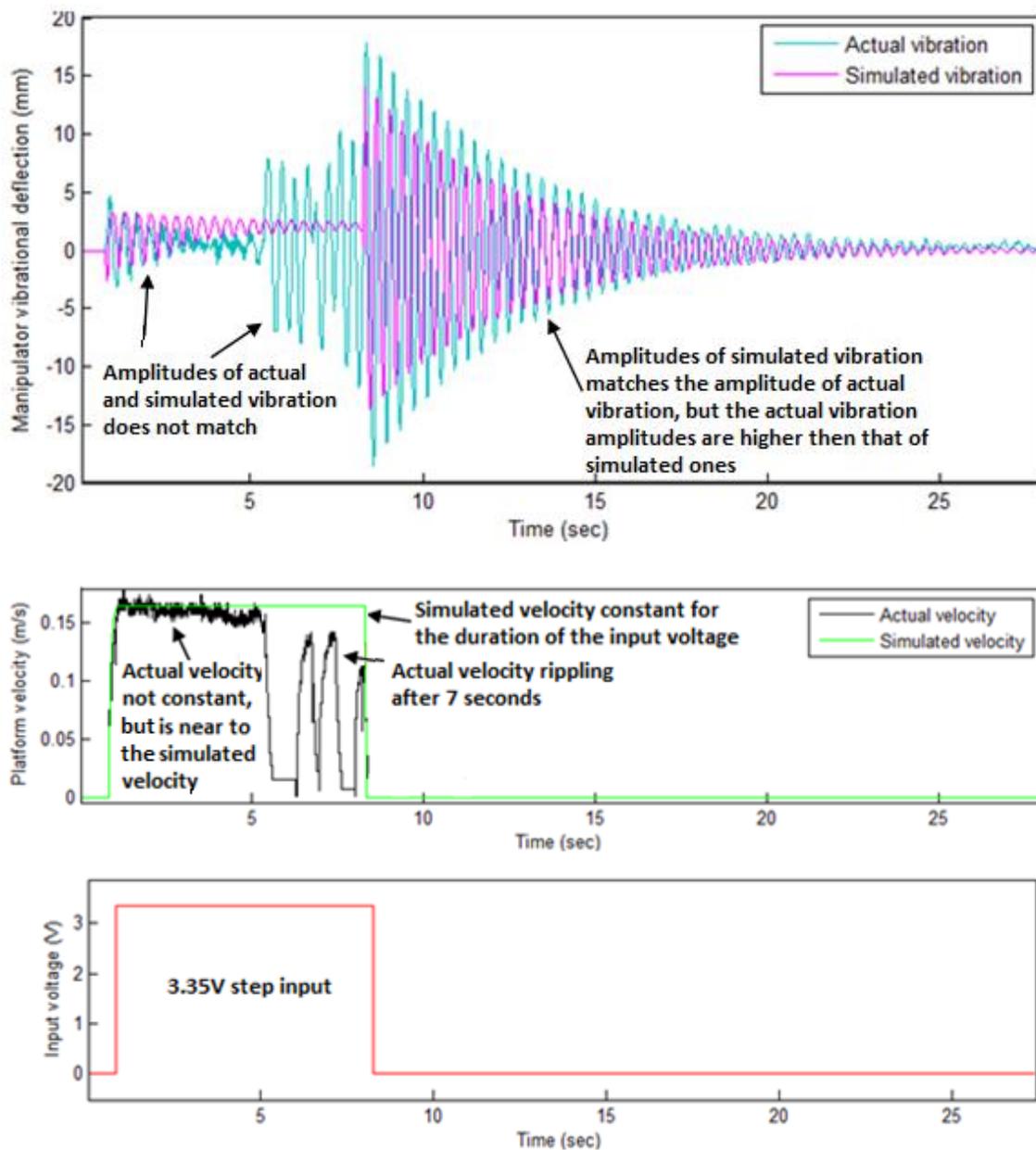
Referring to Fig. 6.10, 3.35 V step input, the simulated velocity is 3.2 m/s. In this test, the disturbances were created after the mid-point of travel. The actual velocity matched quite closely to the simulated velocity for around 4.5 seconds, until disturbances were introduced causing the platform to stop and jerk three times before the end of the input voltage. The disturbances were not being detected by the model, thus the simulated velocity remained constant during the disturbance was being stimulated.

It can be seen that the simulated vibration of the flexible manipulator matched quite well with the actual vibration for the first 4.5 seconds, proved that model is accurate when the actual and simulated velocity matched well. The drawback is inability of the model to detect disturbances.

In the above tests, due to the unmodelled inaccuracies such as disturbances and instable velocity, using our models cannot accurately predict the motion of the platform and the vibration of the flexible manipulator. It illustrated that if the simulated velocity of the matches the actual velocity, it was able to simulate the vibration of the flexible manipulator quite accurately. This means that if there is a feedback to the model, we can harmonize the simulation output for the platform velocity, which would bring to a accurate simulation of the vibration of the flexible manipulator. Sensor feedback and

sensor fusion is required to improve the accuracy of prediction for the MFLM, as will be presented in the next sections.

The contribution in this work is the model of the MFLM that can give a quite accurate prediction of the velocity and vibration of the MFLM. The main drawback is the inability to predict the motions when disturbance is introduced, as well as when insufficient input voltage is introduced to drive the system.



**Fig. 6.10** Flexible manipulator output and platform velocity for 3.35 V input voltage

### 6.3 Experimental test results of the mobile flexible link manipulator with sensors

The results for the test methodology set in section 5.7 are presented in this section, where the acquired signal from accelerometer and camera for estimating the manipulator's vibration were illustrated. The outputs from the sensors are compared with the actual outputs. The validation tests were carried out with no payload, 10 g payload and 30 g payload, respectively at the tip of the flexible manipulator. The results are illustrated as follows:

#### 6.3.1 No payload, tested with various input voltages

Fig. 6.11 to 6.15 illustrate the outputs for the step input voltages of 0.67V, 1.34V, 2.01V, 2.68V and 3.35V, respectively. From Fig. 6.11, it can be seen that the vibration amplitude is low, only the first two cycles of the accelerometer's output accurately matched the actual vibrations. The output then drifted as much as 10 mm after 4.9 seconds. On the other hand, it can be seen that there is no phase shift in the accelerometer's output. This exemplified that the accelerometer provides realtime outputs but with an output drift that cumulates in time. There is no drift in the camera's output, but there is a phase shift of 0.097 seconds, attributed by the low acquisition and processing rate from the camera.

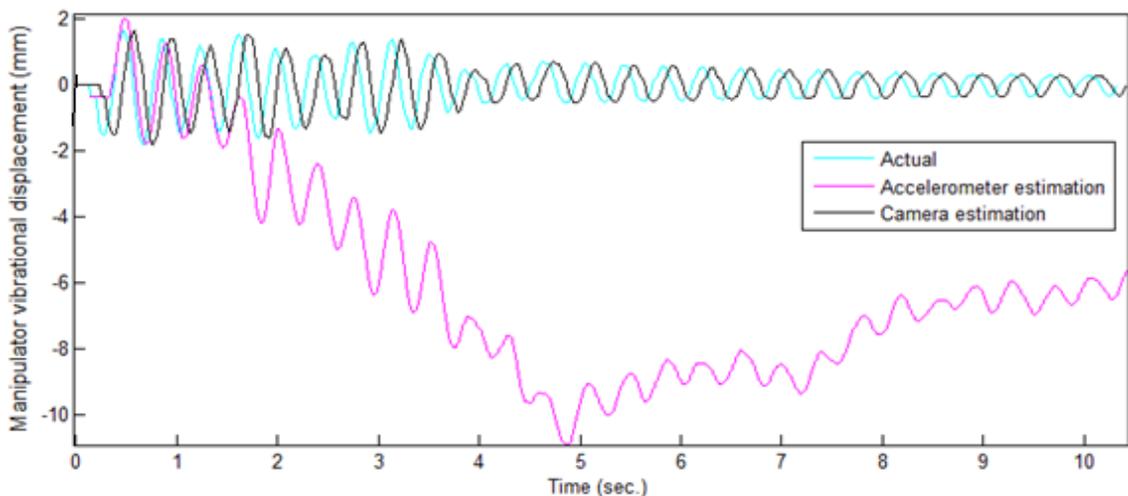
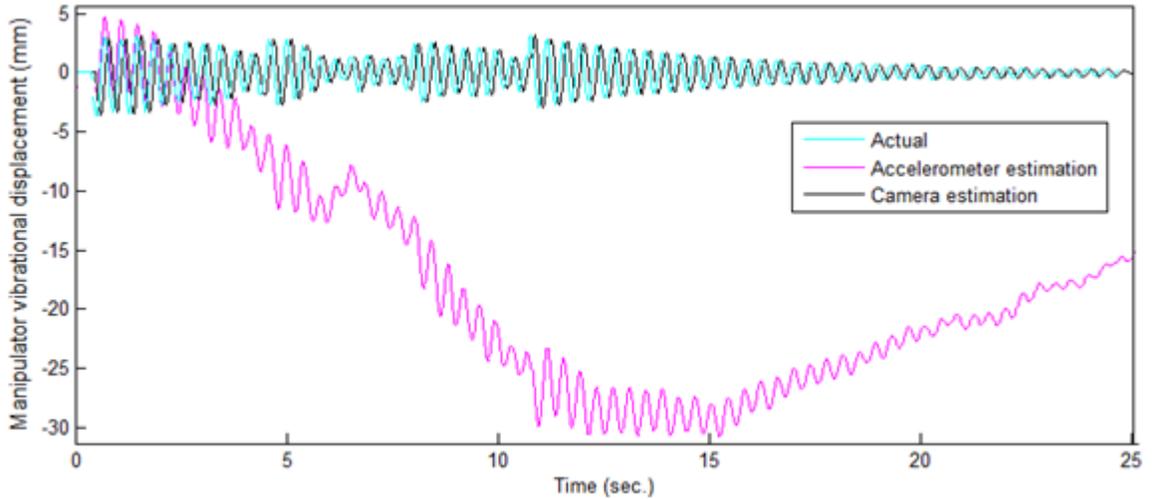
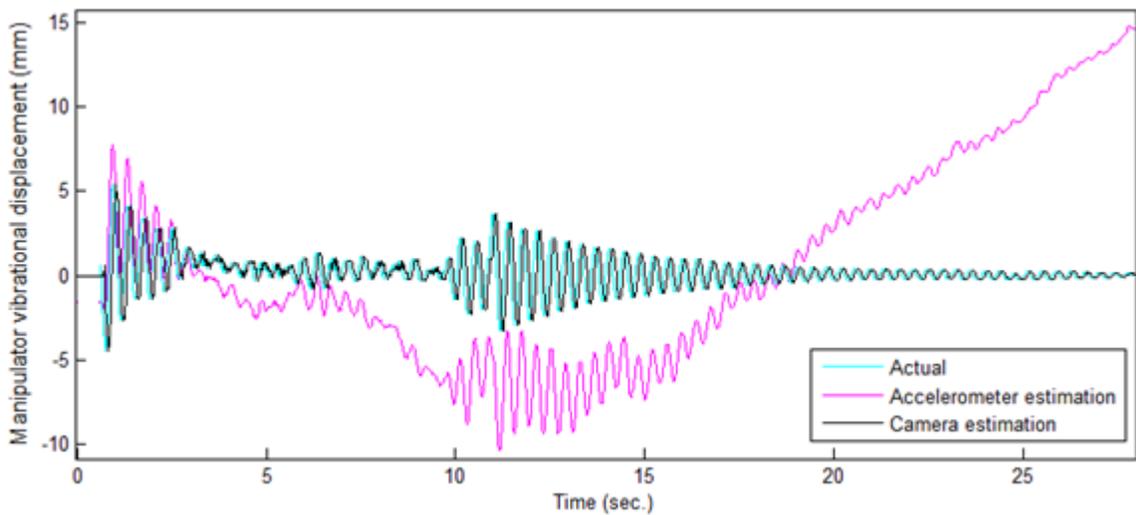


Fig. 6.11 Comparison of accelerometer's estimation and camera's estimation for 0.67 V input voltage



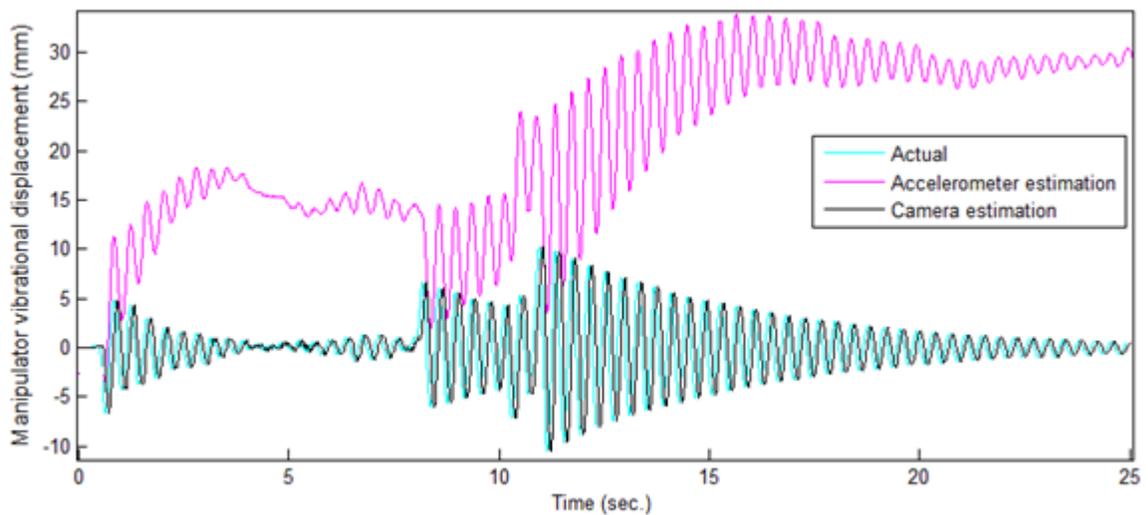
**Fig. 6.12** Comparison of accelerometer's estimation and camera's estimation for 1.34 V input voltage

Fig. 6.12 illustrated the accelerometer's output drift of up to 30 mm, while only the first four cycles of vibration are accurately matched with the actual vibration. The drifting errors from the accelerometer are also attributed by the movement of the mobile platform. The camera still shows accurate but delayed outputs.



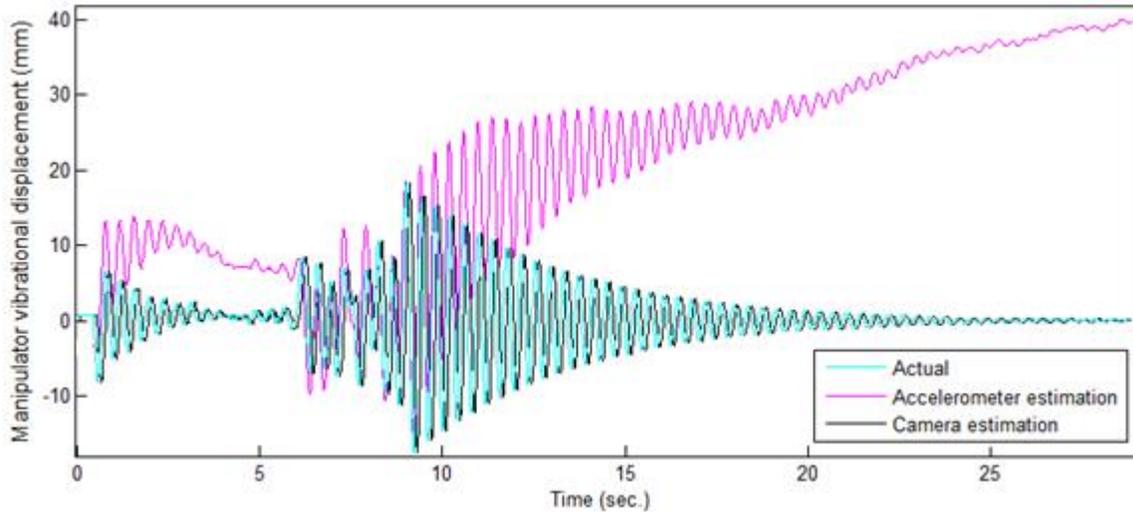
**Fig. 6.13** Comparison of accelerometer's estimation and camera's estimation for 2.01 V input voltage

From Fig. 6.13 above, it can be seen that first sixth cycles of vibration output from the accelerometer are accurate, but at an offset of 2.5 mm. This is due to the sudden acceleration of the mobile platform which the accelerometer reads when the robot starts to move from rest position, which became prominent at high step voltage input of 2.01 V. However, the camera's output is still showing accurate match with the actual output (measured from the IR distance sensor, refer to section 5.7.1 about the sensors) with the same delay. This is because the camera is attached to the base of the manipulator, which moves together with the mobile platform.



**Fig. 6.14** Comparison of accelerometer's estimation and camera's estimation for 2.68 V input voltage

Fig. 6.14 and 6.15 revealed the same phenomenon as Fig. 6.13 for the accelerometer's outputs. The sudden acceleration of the mobile platform caused the first estimation of the vibration cycle to offset by 10 mm for Fig. 6.14 and 12 mm for 6.15. These shows that the movement of the mobile platform accounted for the accelerometer's output errors.

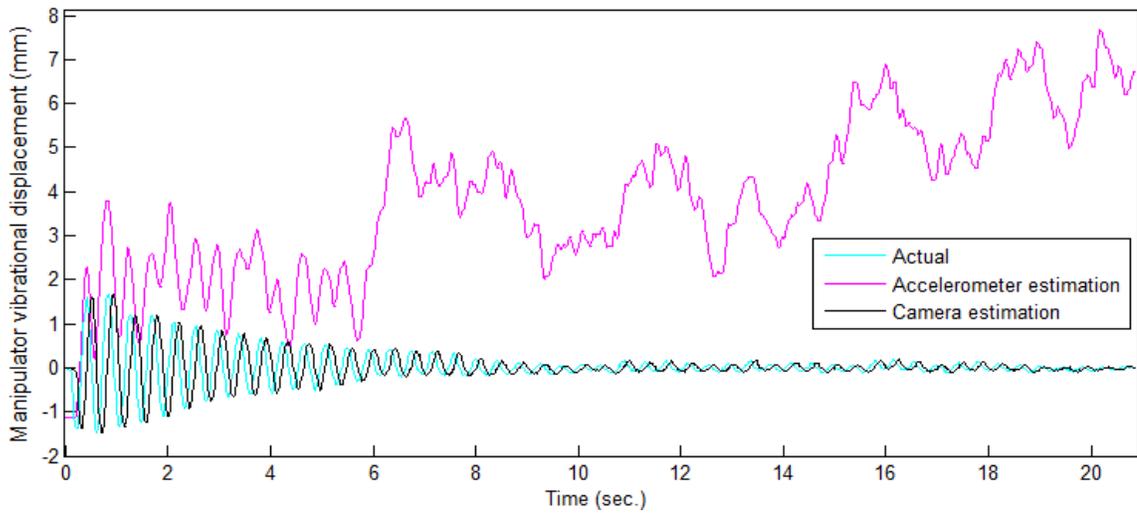


**Fig. 6.15** Comparison of accelerometer's estimation and camera's estimation for 3.35 V input voltage

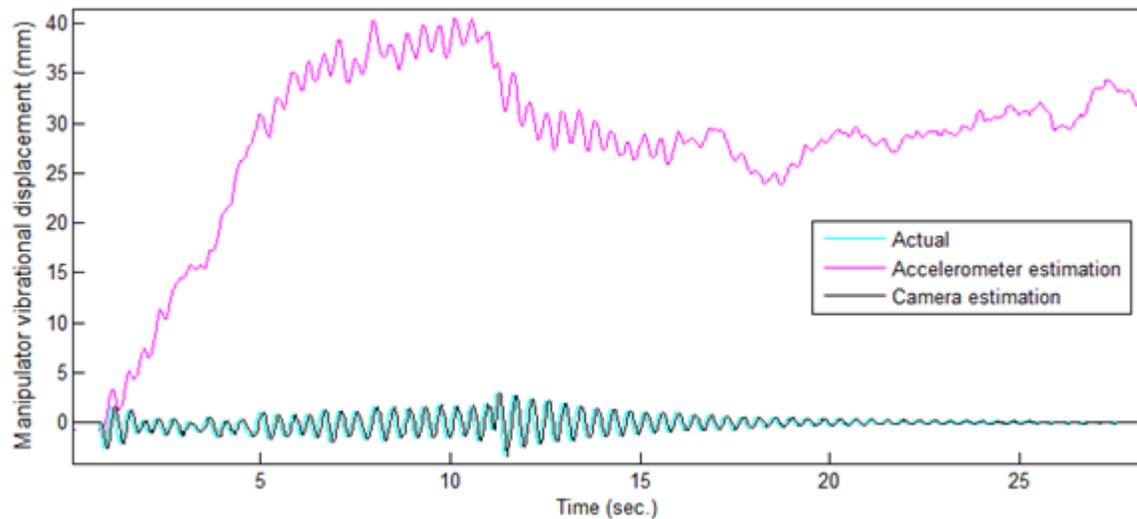
### 6.3.2 With 10 g payload, tested with various input voltages

Fig. 6.16 to 6.20 illustrates the outputs for 10 g payload attached to the tip of the manipulator. Fig. 6.16 illustrated the sensors' outputs at 0.67 V step input. It can be seen that accelerometer's output is visible for only first three vibration cycles, but does not match the actual vibration. This is due to the sudden acceleration of the mobile platform from rest, this attributed to the initial offset of the displacement computation of 2 mm for the first cycle. The remaining signals does match, this could be due to the low signal-to-noise ratio at low velocity of the entire platform. On the other hand, the camera's output showed good match of the vibration at a delay.

Fig. 6.17 illustrated the outputs at 1.34 V step input. It can be seen from the accelerometer's output that only the first cycle matched the actual vibration. The subsequence cycles then drifted to as high as 40 mm, and is hardly be recognised as vibration signal. This is due to the payload that resulting in low vibration frequency, resulting in low signal-to-noise ratio at low velocity of the platform. But, the camera's output showed good match of the vibration at a delay.



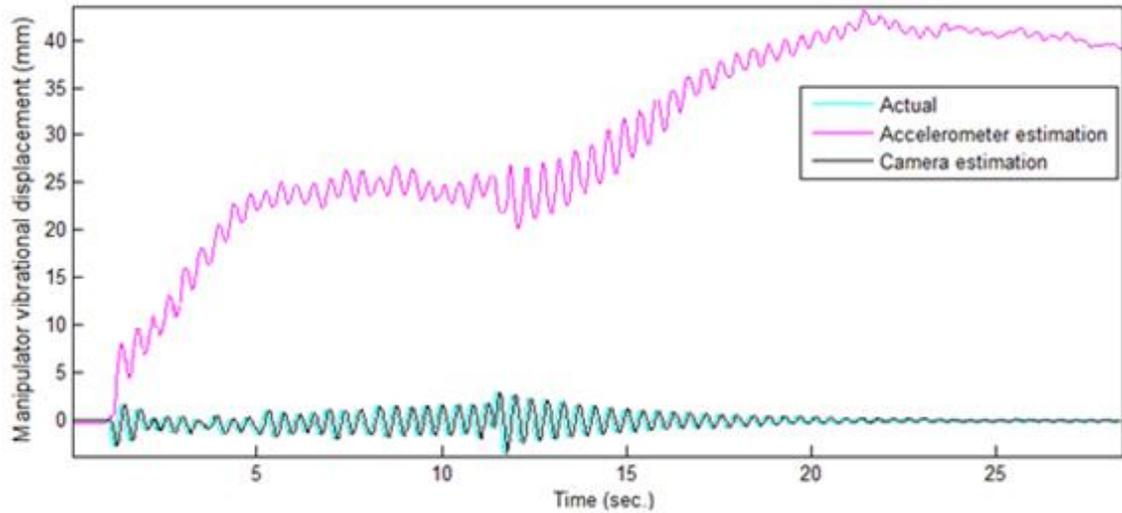
**Fig. 6.16** Comparison of accelerometer's estimation and camera's estimation for 0.67 V input voltage, and 10 g payload at the tip of the manipulator



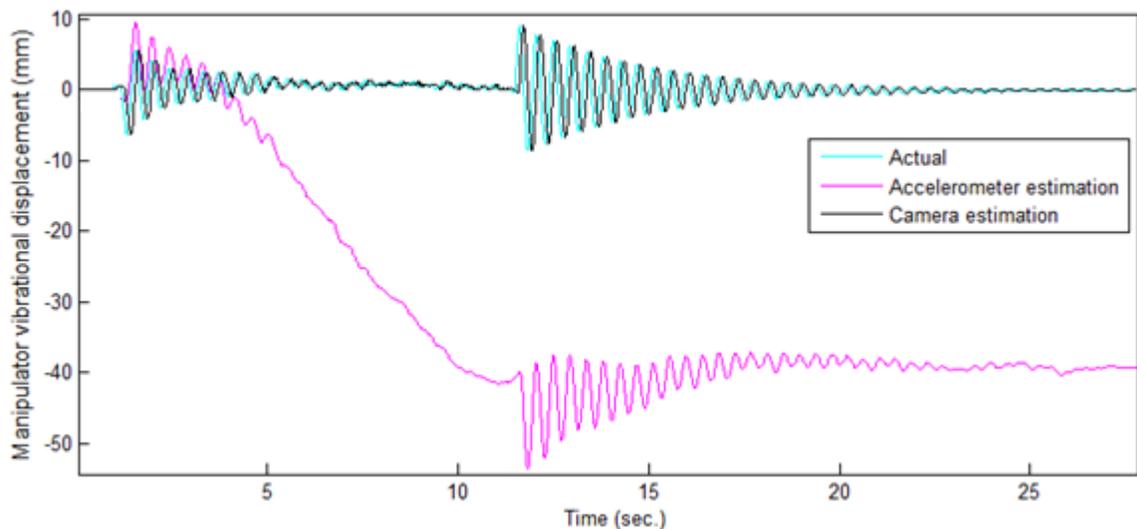
**Fig. 6.17** Comparison of accelerometer's estimation and camera's estimation for 1.34 V input voltage, and 10 g payload at the tip of the manipulator

At 2.01 V step input in Fig. 6.18, due to the acceleration from rest the displacement offset at the first vibration cycle is around 7 mm even through the vibration amplitude does match the actual vibration. Due to the movement of the mobile platform, the displacement output drifted as 25 mm, and when the platform stopped at 12<sup>th</sup> second the

displacement output drifted up to 40 mm. The camera's output for displacement computation remains to be accurate at a delay of 0.097 seconds.



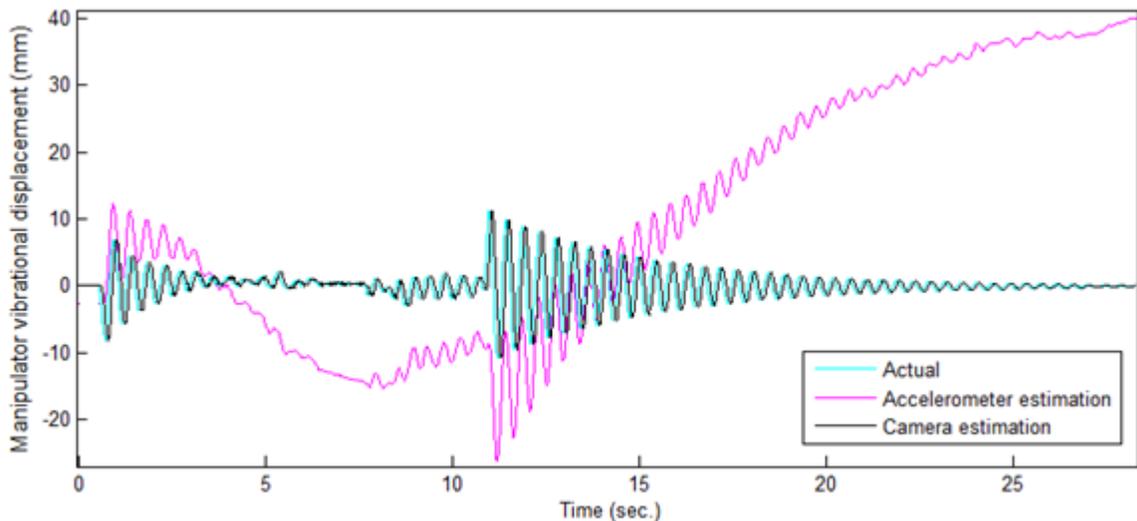
**Fig. 6.18** Comparison of accelerometer's estimation and camera's estimation for 2.01 V input voltage, and 10 g payload at the tip of the manipulator



**Fig. 6.19** Comparison of accelerometer's estimation and camera's estimation for 2.68 V input voltage, and 10 g payload at the tip of the manipulator

At 2.68 V step input in Fig. 6.19 above, we can see good match of five cycles of computed vibration from the accelerometer's output at an offset of 3 mm due to acceleration of the platform from rest. This is due to the stable platform velocity after initial acceleration. However, the output then drifted exponentially to 46 mm at the subsequence motion. This is contributed by the unstable platform velocity that follows due to the weight of the entire robot and cables pulling. When the platform stopped, it depicts a good computation of the vibration output at the 46 mm offset. If this offset is compensated, the computed vibration measurement can match quite well at this 12 second point onwards.

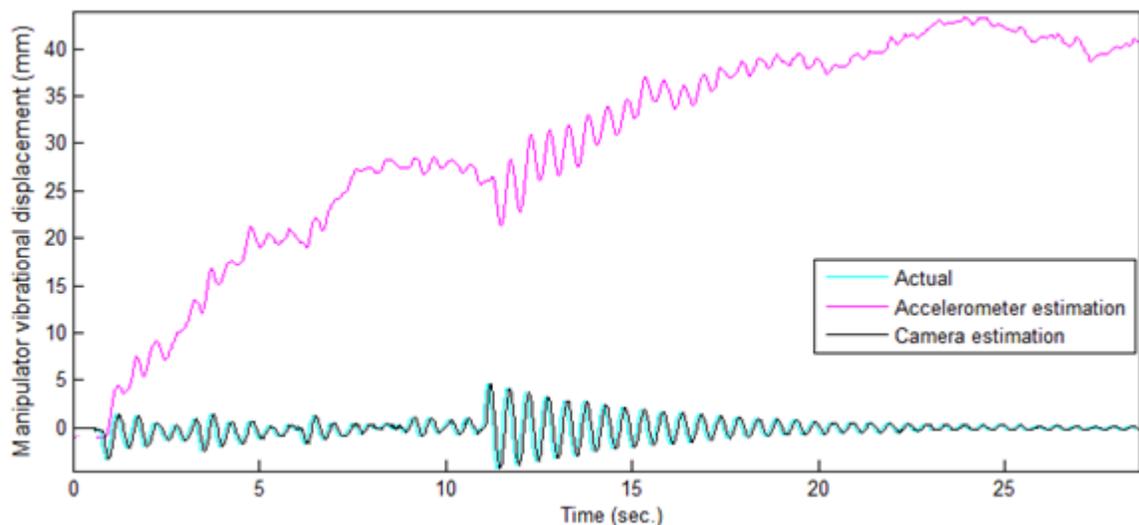
Similarly, at 3.35 V step input in Fig. 6.20 below, the first 5 cycles of vibration computed by the accelerometer has good match with the actual vibration, but at an offset of 5 mm. the subsequent output drifted to 18 mm. When the platform stopped at 11 second, the vibration estimation match the actual vibration but drifting to the other direction.



**Fig. 6.20** Comparison of accelerometer's estimation and camera's estimation for 3.35 V input voltage, and 10 g payload at the tip of the manipulator

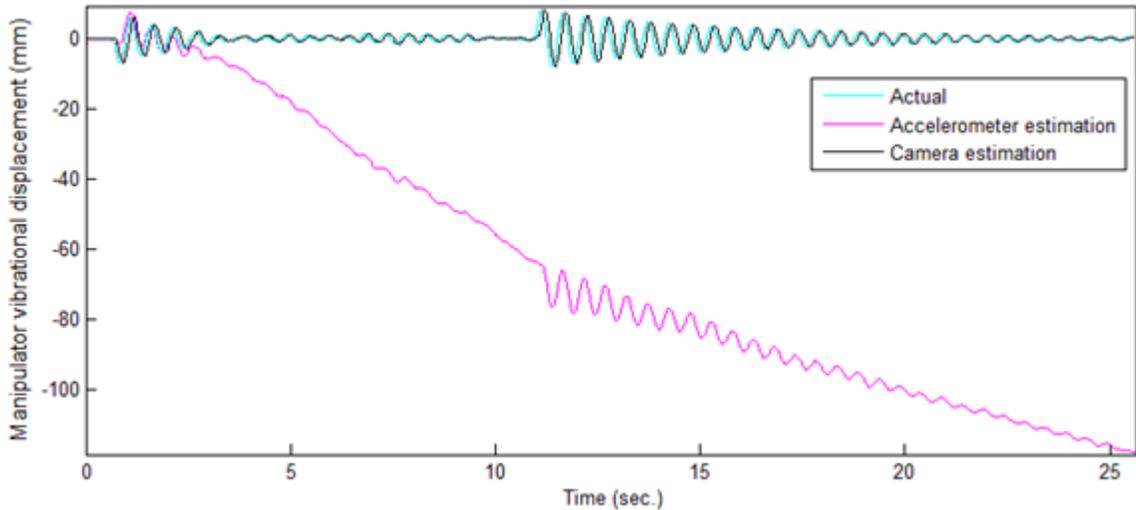
### 6.3.3 With 30 g payload, tested with various input voltages

Fig. 6.21 to 6.24 illustrates the outputs for 30 g payload. As the input for 0.67 V is too low to move the robot with 30 g payload, the test was carried out from 1.34 V step input as shown in Fig. 6.21. The computed output from the accelerometer does not match the actual vibration during the moving of the platform, as the noise-to-signal ratio is too high attributed by the irregular movement of the platform. The computed vibration when the platform stopped also does not match well due to the low vibration rate. The measurement from the camera contributed good match at a delay.

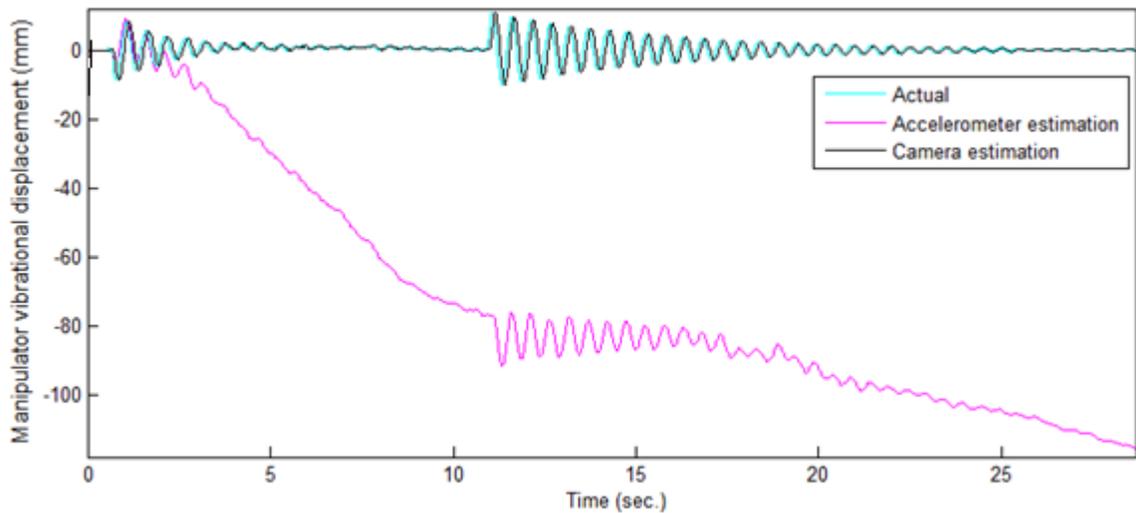


**Fig. 6.21** Comparison of accelerometer's estimation and camera's estimation for 1.34V input voltage, and 30 g payload at the tip of the manipulator

At the 2.01 V (Fig. 6.22) and 2.68 V (Fig. 6.23) step inputs, the first three vibration cycles of the accelerometer's output match well and then exponentially until the platform stopped at 11<sup>th</sup> second. The whole computed signal from the accelerometer drifted up to 120 mm at the end of 25 seconds.

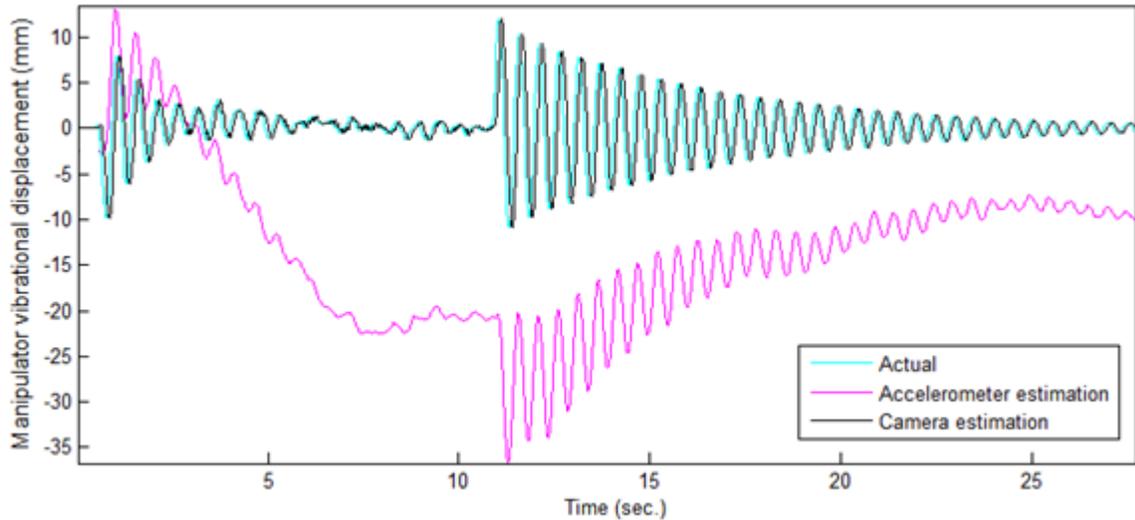


**Fig. 6.22** Comparison of accelerometer's estimation and camera's estimation for 2.01V input voltage, and 30 g payload at the tip of the manipulator



**Fig. 6.23** Comparison of accelerometer's estimation and camera's estimation for 2.68V input voltage, and 30 g payload at the tip of the manipulator

As shown in Fig. 6.24, at the 3.35V step input, the first three vibration cycles match with 5 mm offset due to the initial acceleration of the platform. It then drifts until the platform stopped. The drifting vibration measurement from the accelerometer match well at stopped position. This showed that at higher input voltage the movement of the platform is more stable and thus contributed to better vibration measurement from the accelerometer.



**Fig. 6.24** Comparison of accelerometer's estimation and camera's estimation for 3.35V input voltage, and 30 g payload at the tip of the manipulator

From the above investigations, it can be seen from the plots that the accelerometer's measurement for the vibration exhibits drifting errors with time, with the drift as high as 120 mm. The drift is contributed by the combination of accelerometer's internal errors and the motion of moving platform which is detected as acceleration signal. Thus, accelerometer estimation is only good for short term accuracy for vibration measurements. The accuracy is good for minimum of one vibration cycle only, and up to five cycles for stable platform movements.

The cameras' estimation is delayed by 0.0968 seconds due to the time delay in capturing the image and the processing time required for converting the image into displacement output. In image acquisition, it is not possible to determine exactly the time where the image was being captured. A camera with frame-rate of maximum 60 frames per second (fps) would have the image being captured at anytime within the timeframe of 16.7 milliseconds. Furthermore, there is time required for the program to load and process the frames to convert them into displacement information. Thus, increasing the camera frame rate, the image acquisition and processing time will eventually become the actual bottleneck. From the captured images using camera, there were negligible frames drops noticed, thus we assumed that there is no frames drop.

With the reasons above, it is therefore not possible to accurately or real-time for computation of the vibrational displacement of the manipulator based on either the accelerometer or camera alone. It is envisioned that fusion of the two sensors will provide more accurate and real-time estimations. The fusion algorithm should harmonize the advantages of both sensors so as to improve the measurement accuracy for the vibration of the flexible link manipulator. The sensor fusion algorithm developed in section 5.2 in chapter 5 will be tested in the following section.

## **6.4 Test results with sensor fusion algorithm**

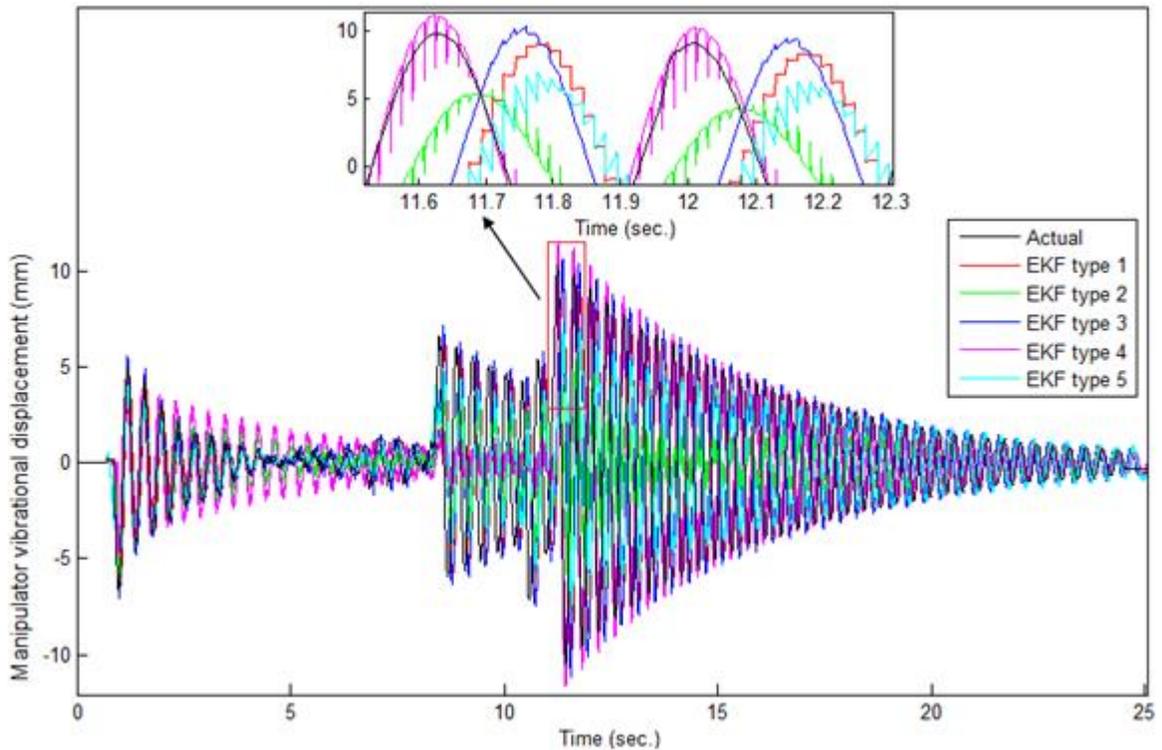
The test results for the test methodology as set in section 5.8, in the methodology chapter, are presented here. The results without camera data correlation are presented first, and followed by the results with camera data correlation.

### **6.4.1 Tests without camera data correlation**

Fig. 6.25 presents the sensor fusion estimation for the comparisons of the 5 types of EKF's for the step input voltage of 2.68V to the driving motor of the mobile platform. In this test, there is no correlation of the camera data with the accelerometer data. The camera data is lag by 0.0968 seconds. It can be seen that the output from type 1 EKF is shifted by 0.18 seconds, and has step effect. Type 2 EKF has 0.08 seconds output shifted, but has spiking due to adjustment when the camera data become available. Type 3 EKF has 0.14 seconds output shifted, but has best match to the actual output. Type 4 EKF has no shift of the output, but has spiking due to adjustment when the camera data become available. Type 5 EKF has output shifted by 0.19 seconds, and has rippling output.

In conclusion, for fusion algorithms that made use of previous camera data or extrapolation of previous camera data have smooth output signals, but showing output drifts. For the accelerometer only data, there is drifting of the estimation, and the output is showing rippling. This is cause by signal adjustment when camera data presents.

On the other hand, for estimation that made use of model in place of camera data shows no drift of the output, but has output surging effects due to adjustment when camera data became available.

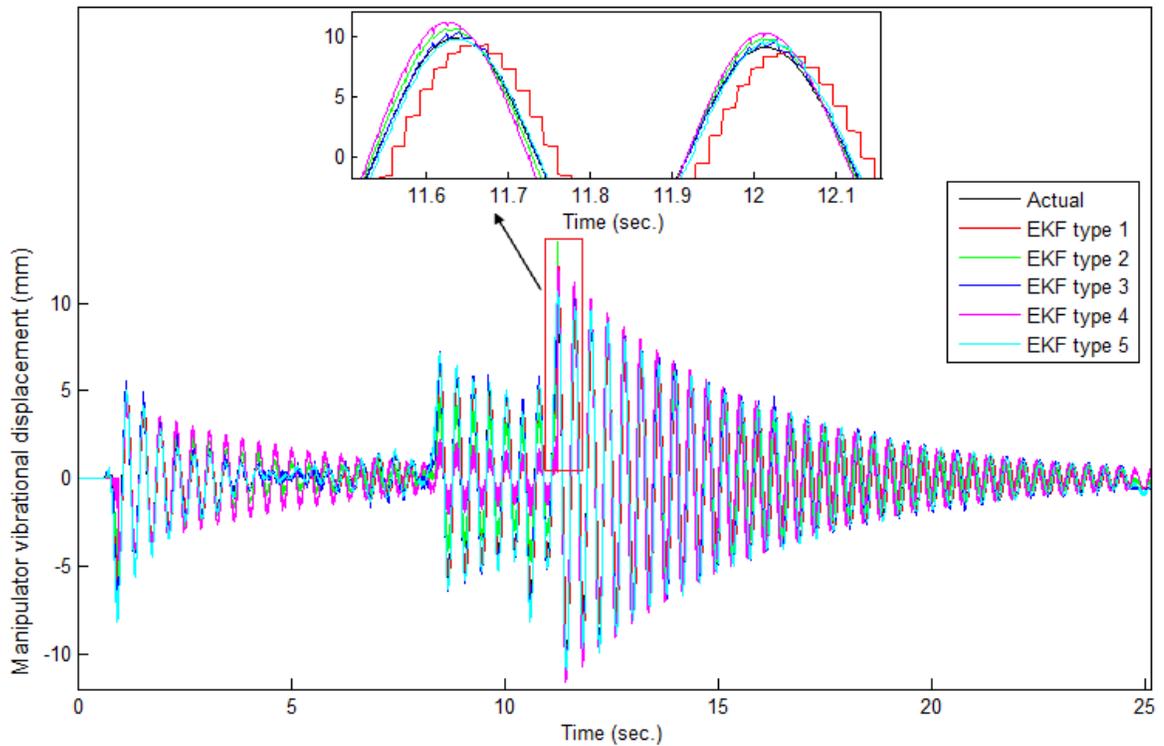


**Fig. 6.25** Comparison of Extended Kalman filter sensor fusion methods for 2.68V input

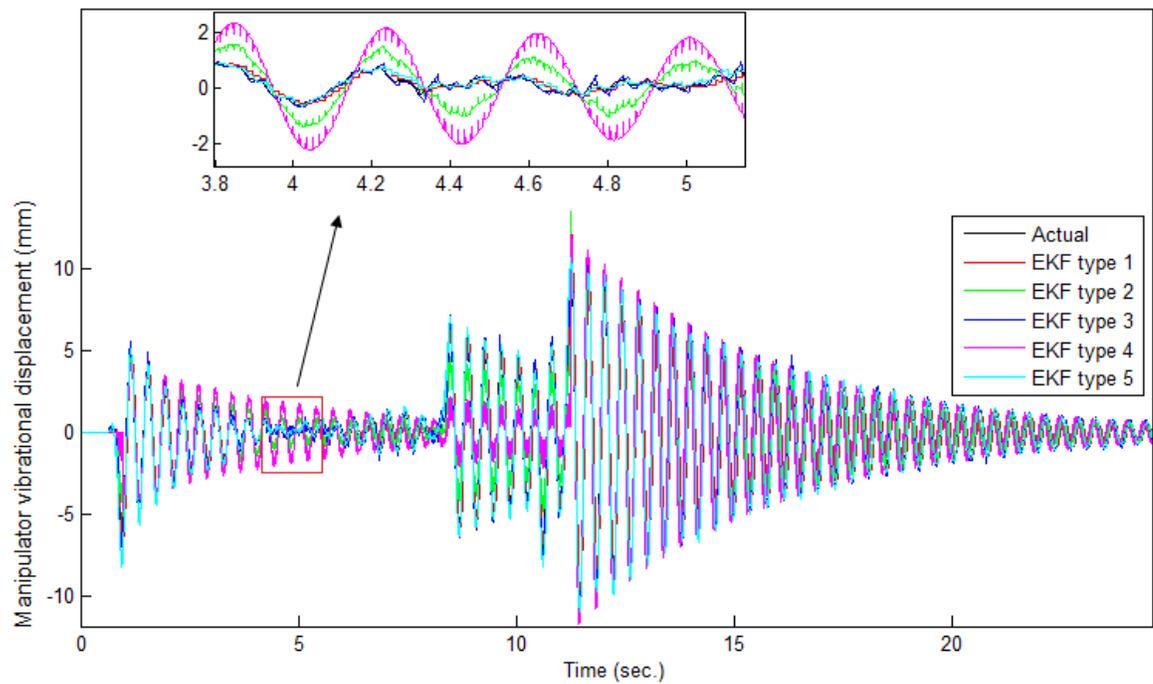
#### 6.4.2 Tests with camera data correlation

Fig. 6.26 presents the sensor fusion estimation for the comparisons of the 5 types of EKFs with camera data correlation with accelerometer data, for the step input voltage of 2.68V to the driving motor of the mobile platform. It is now 0.02 seconds drift for the type 1 EKF. Type 3 EKF has best match to the actual displacement, followed by Type 5 EKF, manifesting that the fusion with extrapolation of previous camera data provides good accuracy to the estimation. Type 2 and Type 4 EKF have slight overshoot of the estimation at peak displacement of manipulator's tip. This denotes that the model based EKF does not contribute good improvement to the measurement accuracy if the model does not accurately match system output.

It therefore evidenced that the model-based EKF sensor fusion algorithm for vibration tracking does not give rise to more accurate tracking to the vibration of the flexible manipulator due to the unmodelled dynamics of the mobile platform. With correlation to the camera data, the accelerometer fusion with extrapolated camera data contributed to best measurement outputs.



**Fig. 6.26** Comparison of Extended Kalman filter sensor fusion methods to estimate manipulator vibration for 2.68V input (with camera correlation) at instant when mobile platform stopped

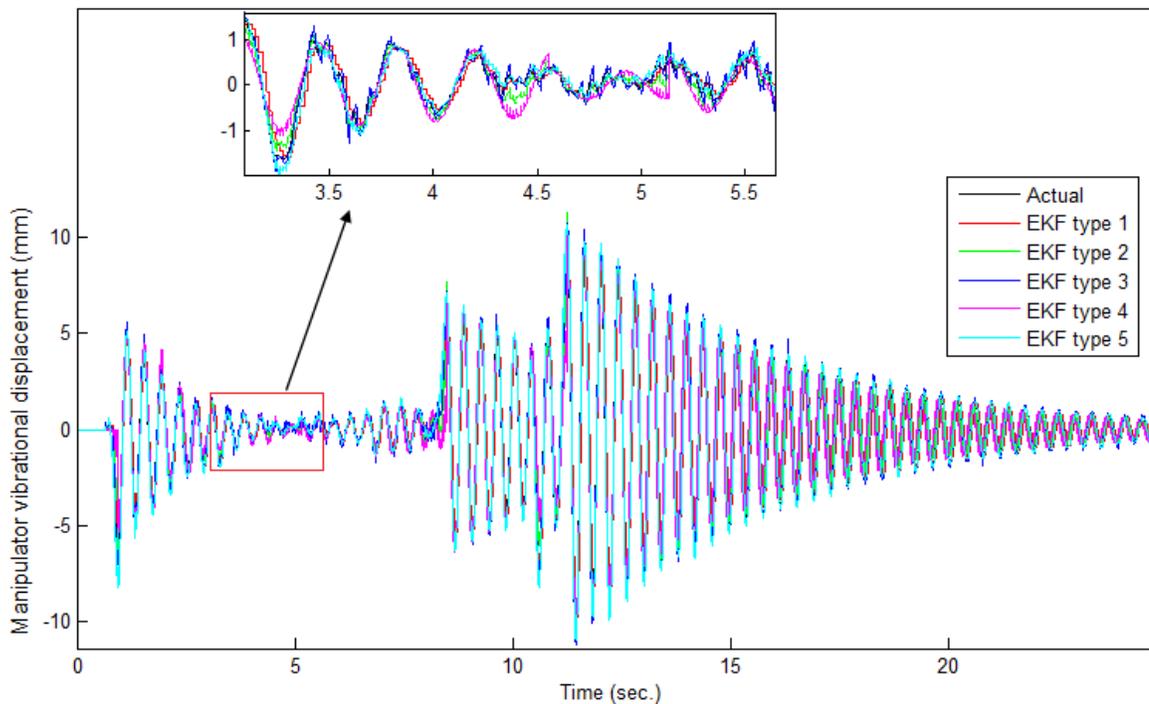


**Fig. 6.27** Comparison of Extended Kalman filter sensor fusion methods to estimate manipulator vibration for 2.68V input (with camera correlation) during motion

However, the tracking for the vibration during motion of the mobile platform is mismatching due to unstable vibration frequency and amplitude, as illustrated in Fig. 6.27 above.

### 6.4.3 Tests with windowed frame for vibration tracking

To improve the unstable amplitude problem, windowed frame is used. Every windowed frame for the period of previous one cycles of vibration is then used to recalculate and update the modelled vibration. Fig. 6.28 illustrates the improvement of the match between the computed outputs to the actual vibration. It can be seen that there are good match for all EKF types. EKF types 1, 3 and 5 matches well with the actual vibration, while EKF types 2 and 4 having some errors due to the residual modelled error.



**Fig. 6.28** Comparison of Extended Kalman filter sensor fusion methods to estimate manipulator vibration for 2.68V input (with camera correlation and recalculation of modelled vibration at fixed interval) during motion

Tests were then initiated by manually exciting the tip of the beam, and allow it to vibrate freely. Fig. 6.29 depicts the outputs based on standard EKF. In case of EKF Type 5

(accelerometer only), there is no drifting but showing rippling outputs. This could be due to EKF only compute the output based on the accelerometer's signal due to the period where the camera data are absent. Once the next camera data arrives, the EKF computes the next output using the fusion of the camera's and accelerometer's data. This results in the rippling output. Fig. 6.30 illustrates the phenomenon.

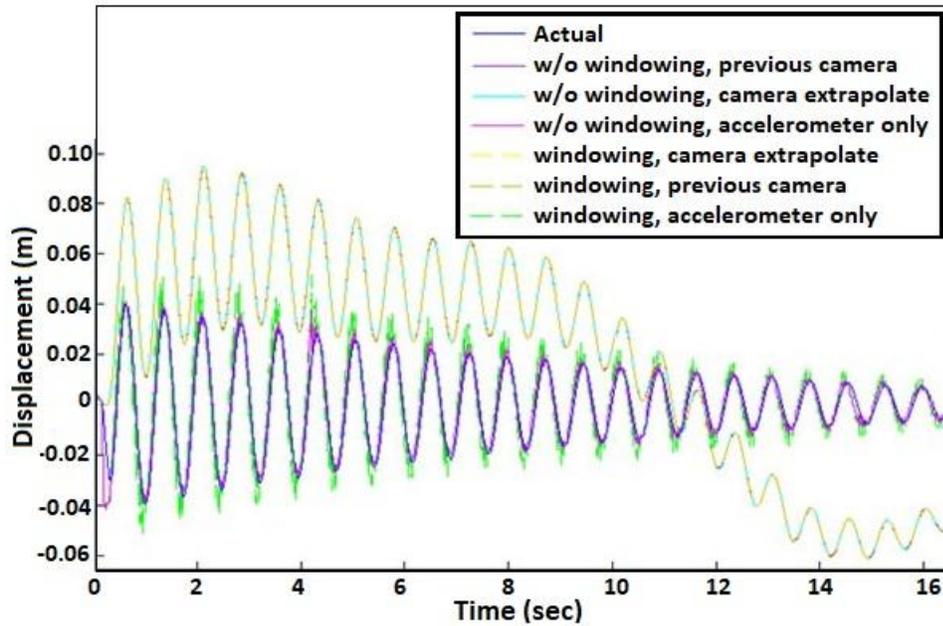


Fig. 6.29 Comparison of sensor fusion methods using Extended Kalman filter

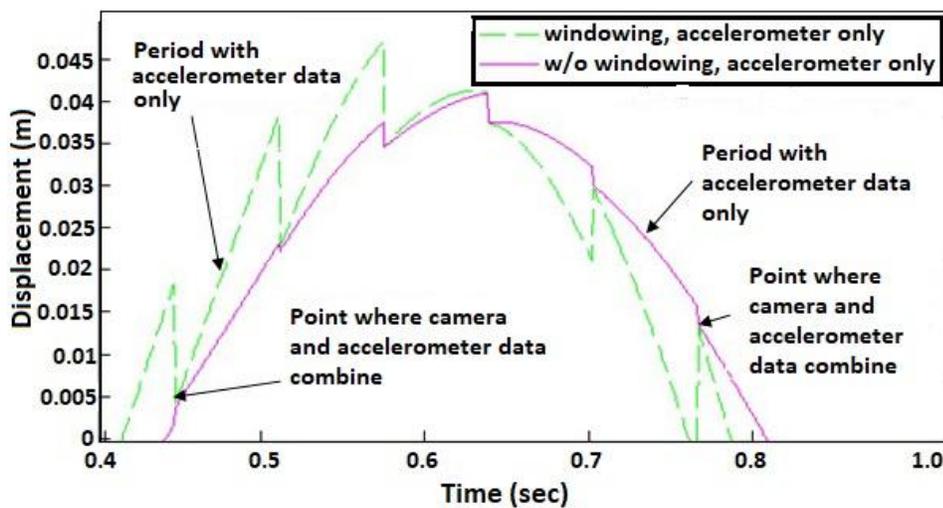
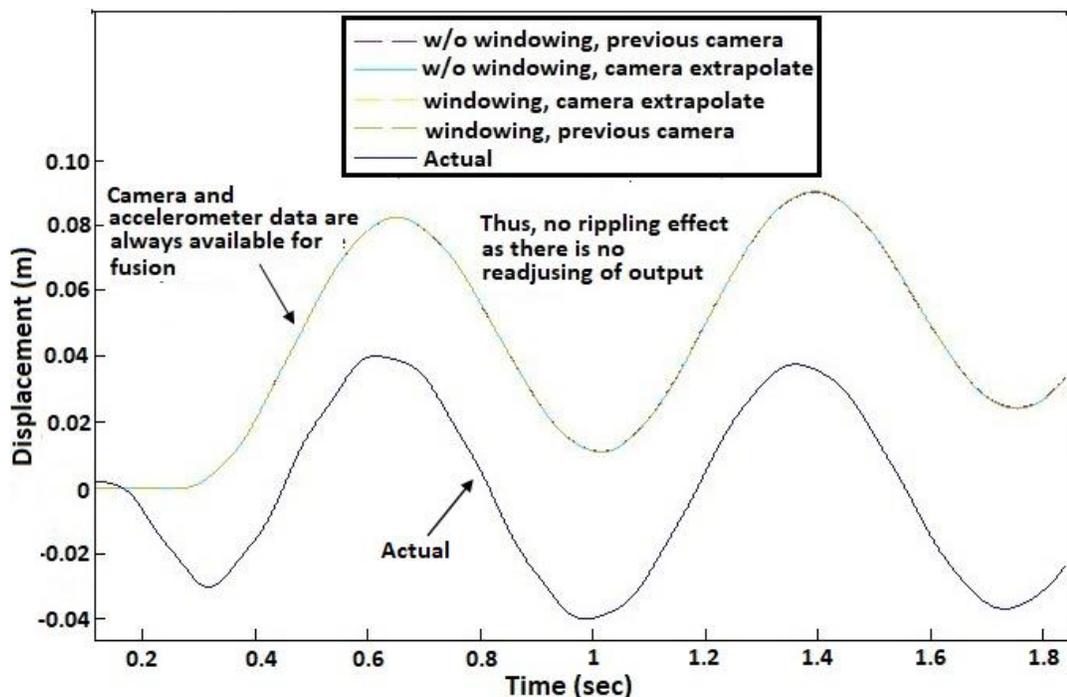


Fig. 6.30 Illustration of EKF Type 5 output

Conversely, for outputs that are based on data from previous camera's output or based on extrapolation of data from previous camera's output, which is the case of EKF Type 1, EKF Type 2 and EKF Type 3, the output signals seems to be smooth but shows drifting outputs. Here, the data from the camera is either using the data from previous camera's output or the extrapolation of data from previous camera outputs in order to fuse with the data from the accelerometer. Therefore, both the data that comes from the camera and the accelerometer fused to produce the EKF output. As a consequence, the output drifts could be due to the effect of the acceleration drift. Fig. 6.31 illustrates the phenomenon.



**Fig. 6.31** Illustration of case (2) and case (3) EKF output

Fig. 6.32 depicts the outputs from the model-based Extended Kalman filter, EKF Type 4. All outputs shows no drift, while EKF Type 5 output for both with and without windowing of accelerometer data appears to have rippling of the signals. This could be due to the adjustment of the signal when camera data presents. The remaining outputs

are smooth, with windowing previous accelerometer data and extrapolation of previous camera data illustrating best match to the actual displacement.

Therefore, we can accept the fact that the model-based Extended Kalman filter sensor fusion algorithm (EKF Type 4) for vibration tracking holds accurate and smooth outputs for tracking vibration the of flexible manipulator.

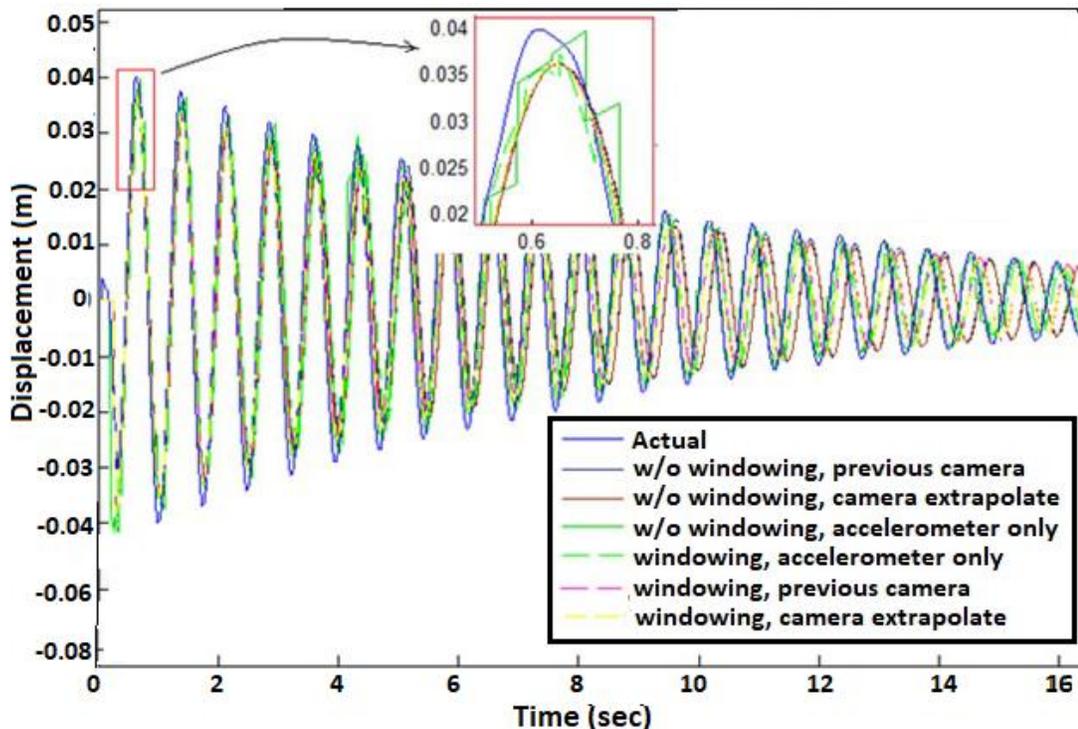


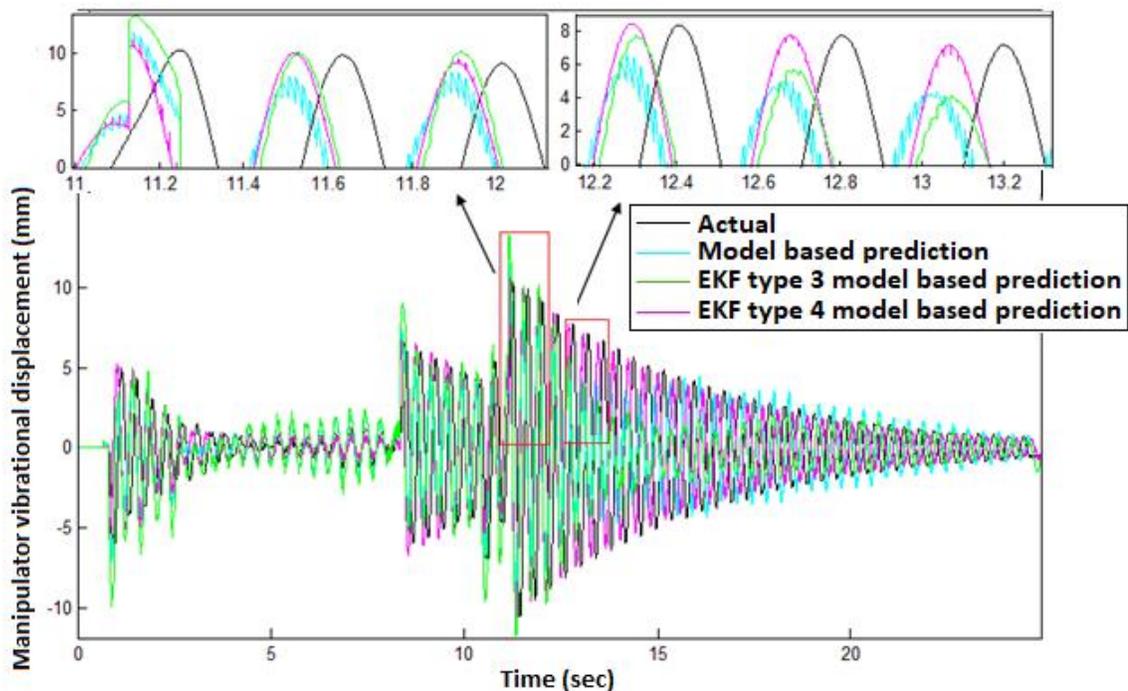
Fig. 6.32 Comparison of sensor fusion methods using model-based Extended Kalman filter, EKF Type 4

## 6.5 Validation results with predictive algorithm

With the methodology for the validation tests set in section 5.9, we present here the results that compare the prediction accuracy between model-based algorithms, Type 3 EKF model based algorithm and Type 4 EKF model based algorithm. Section 6.5.1 presents the results that compare these three types of predictive algorithms without payload at the manipulator's tip, while 6.5.2 presents the results comparing the three types of predictive algorithms with different payload and at different input voltages to drive the mobile platform.

### 6.5.1 Validation test results comparing the predictive algorithms

Fig. 6.33 below depicts outputs that compare three types of predictive algorithms. It can be seen that there is rippling output for the model based prediction, which is due to adjustment when the camera data become available. It does not match well with the amplitude of the actual displacement. The type 3 EKF and type 4 EKF model based predictions have better match with the actual displacement at the beginning of excitation when the mobile platform stopped, but the type 3 EKF model based prediction became less accurate in terms of amplitude as the vibration depreciates. It is noticed that type 4 EKF maintain well match with the actual displacement during the transient response of the manipulator, which signifies to be best choice for vibration prediction of the MFLM. During the initial prediction when the manipulator changes in response due to stopping of the mobile platform, there is an overshoot for the three types of prediction algorithms when the sensor updates. This is due to the sudden change in the sensor output and updated to the prediction input.



**Fig. 6.33** Comparison of predictive algorithm for 0.12 seconds ahead prediction of vibration

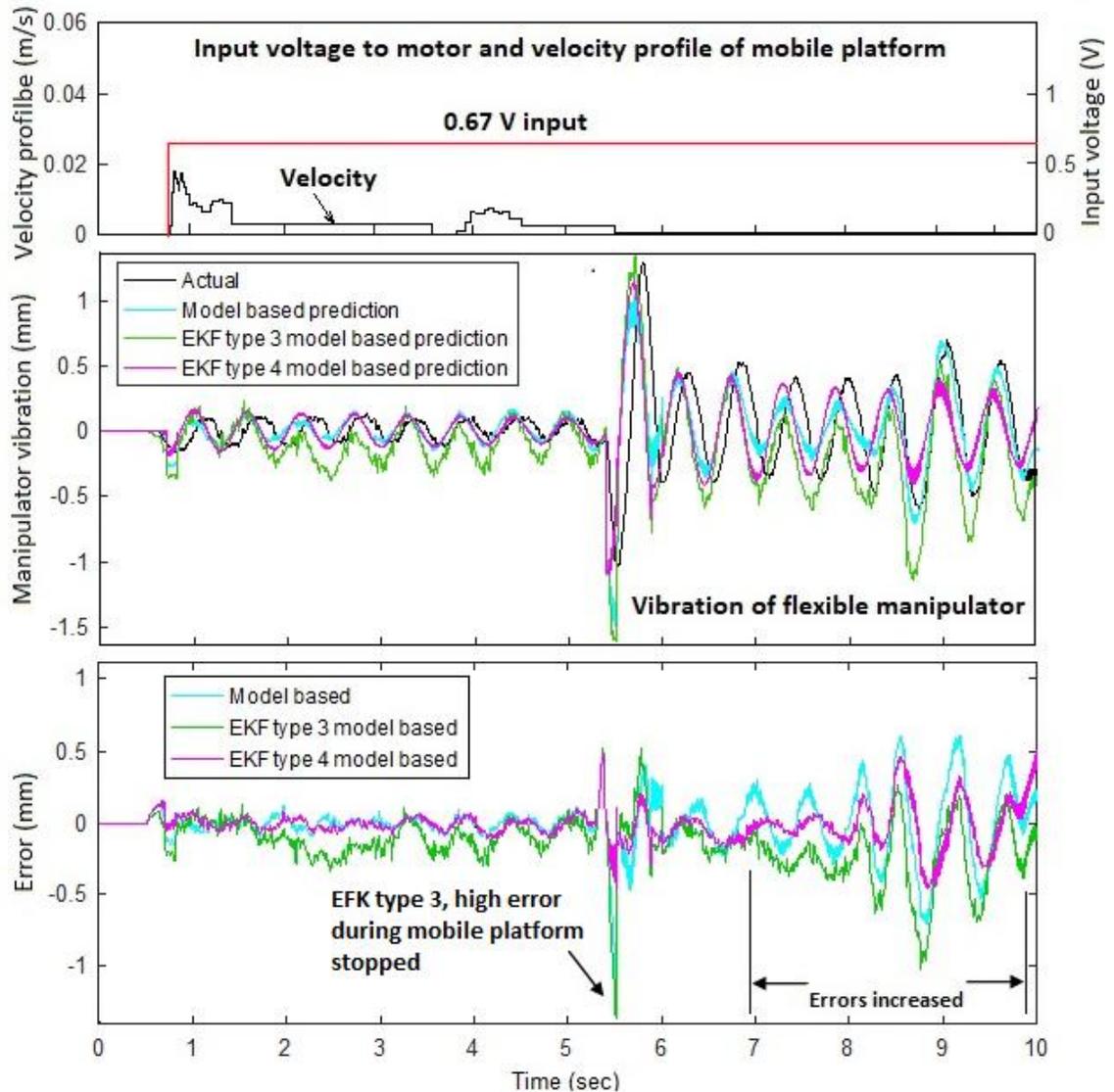
### 6.5.2 Validation tests with different payloads and input voltages

Test verifications were carried out with 0.67 V, 1.34 V, 2.01 V, 2.68 V and 3.35 V input (these voltages are used due to the programme of the Arduino microcontroller as explained in second paragraph of section 5.6.2) to the driving motor of the platform, and with 10 g, 30 g, 50 g and 70 g load at the tip of the manipulator for each input voltage, respectively. For weight of 80 g and above, the platform begins to overturn, thus the tests stopped at 70 g load. In addition, because of the space constraint of 2.4 m, each test was run for 10 seconds straight path of the mobile platform movement. In order to remove any external factors that will contribute to the errors due to disturbances, the empirical cable that will pull the platform has been removed, so that the entire robot is only acted by its own weight.

To drive the motor, an Arduino microcontroller was programmed to send input signal to the motor driver board (MD10C), which send PWM voltage to the motor. The pulsation signal from the encoder of the motor was then acquired by NI 9411 device through Matlab. The acquired pulsation signal was then converted into velocity data using Matlab. (The code for the Matlab and Arduino are furnished in Appendix D and E)

The model based prediction algorithm, type 3 and type 4 EKF model based prediction algorithms were compared. Fig. 6.34 to 6.38 illustrates the predicted vibration outputs for 50 g payload attached at the tip of the flexible link manipulator for 0.67 V, 1.34 V, 2.10 V, 2.67 V and 3.35 V, respectively.

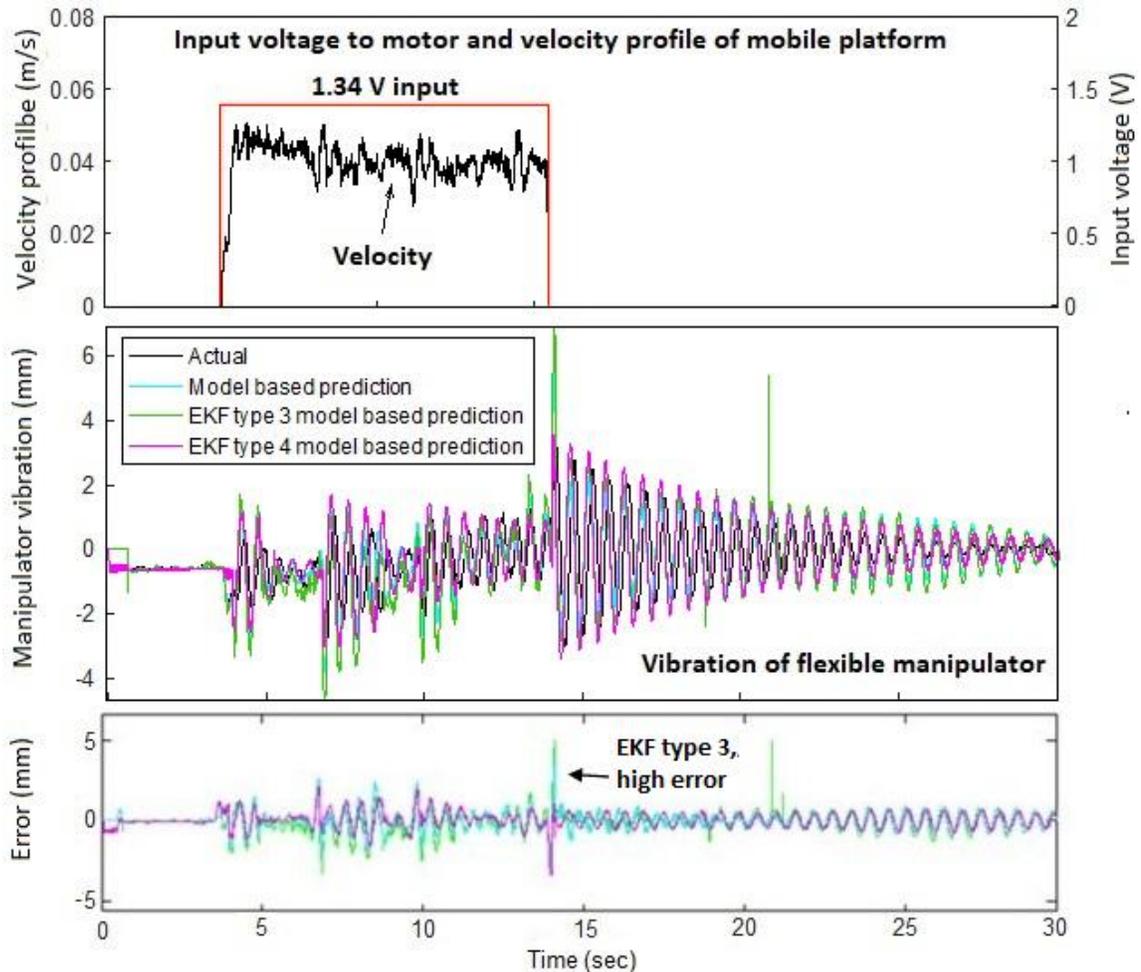
Fig. 6.34 below illustrates the vibration of the mobile flexible manipulator at 0.67 V input to the motor. As the platform velocity is very slow the vibration is smooth during the motion of the mobile platform. However, the input voltage of 0.67 V can only drive the platform for 5 seconds and stopped. The EKF type 3 experiences the highest error as compared to the two other methods. And during when the platform stopped, EKF type 3 encounters a spike in error due to the sudden change in the vibration. The vibration error for the three types of algorithms then increases from the 5.5 second onwards. This could be due to the unknown jerking of the platform as the small voltage of 0.67 V was still trying to drive motor. EKF type 4 experiences the lowest error for the entire duration.



**Fig. 6.34** Comparison of model based, EKF type 3 model based prediction and EKF type 4 model based prediction for 50g payload at the manipulator tip for 0.67 V input

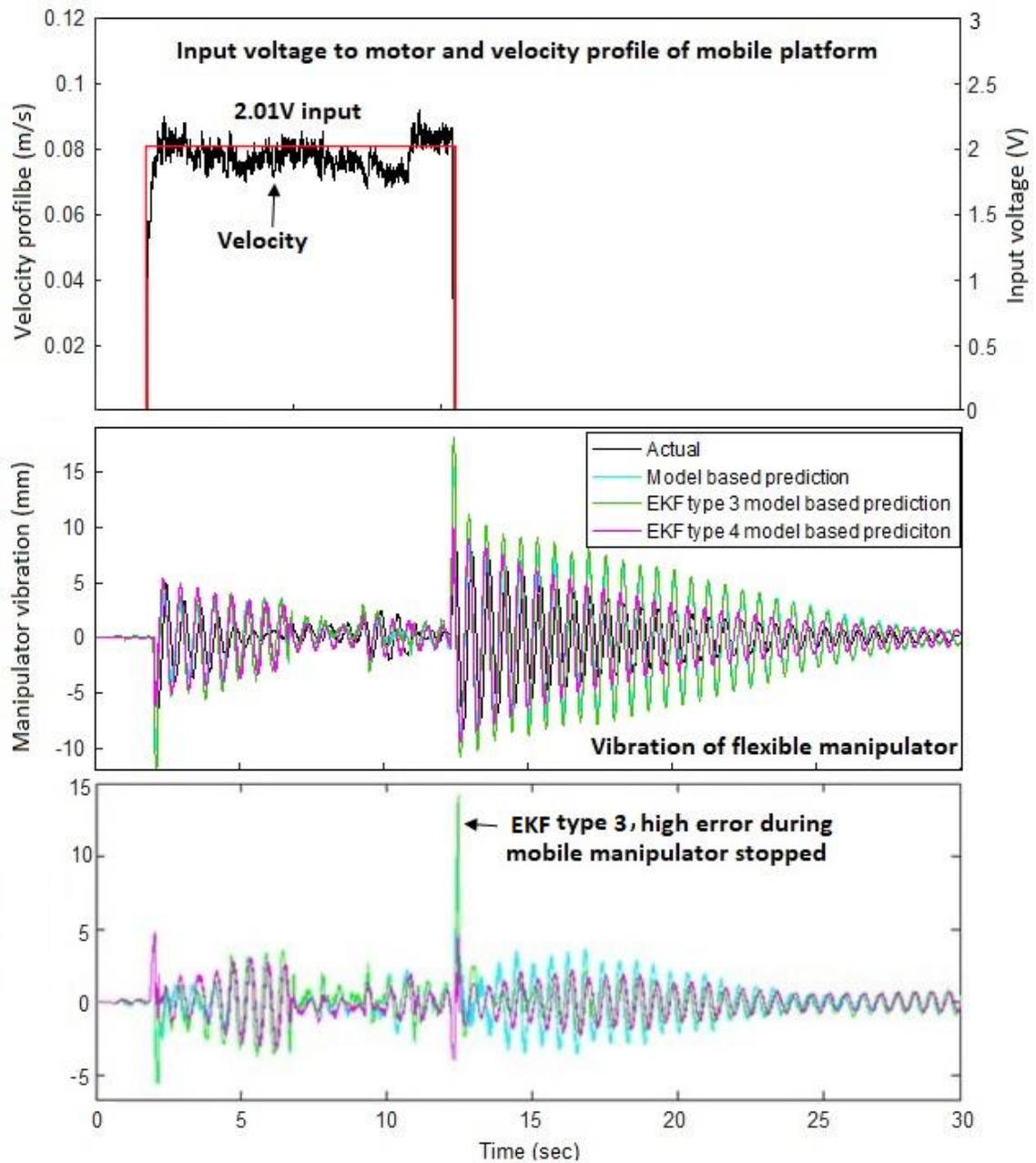
Fig. 6.35 illustrates the vibration of the mobile flexible manipulator at 1.34 V input to the motor. The platform velocity is fluctuating around 0.04 m/s, thus resulting in irregular vibration of the flexible manipulator in the duration of the platform movement. It can be seen that EKF type 3 experiences highest error as compared to the model based prediction and the EKF type 4, while the EKF type 4 has lowest errors. During the instantaneous starting and instantaneous stopping of the mobile platform, the all three

types of prediction method carry a spike in error, with EKF type 3 carries highest error. EKF type 4 carries lowest overall errors.



**Fig. 6.35** Comparison of model based, EKF type 3 model based prediction and EKF type 4 model based prediction for 50g payload at the manipulator tip for 1.34 V input

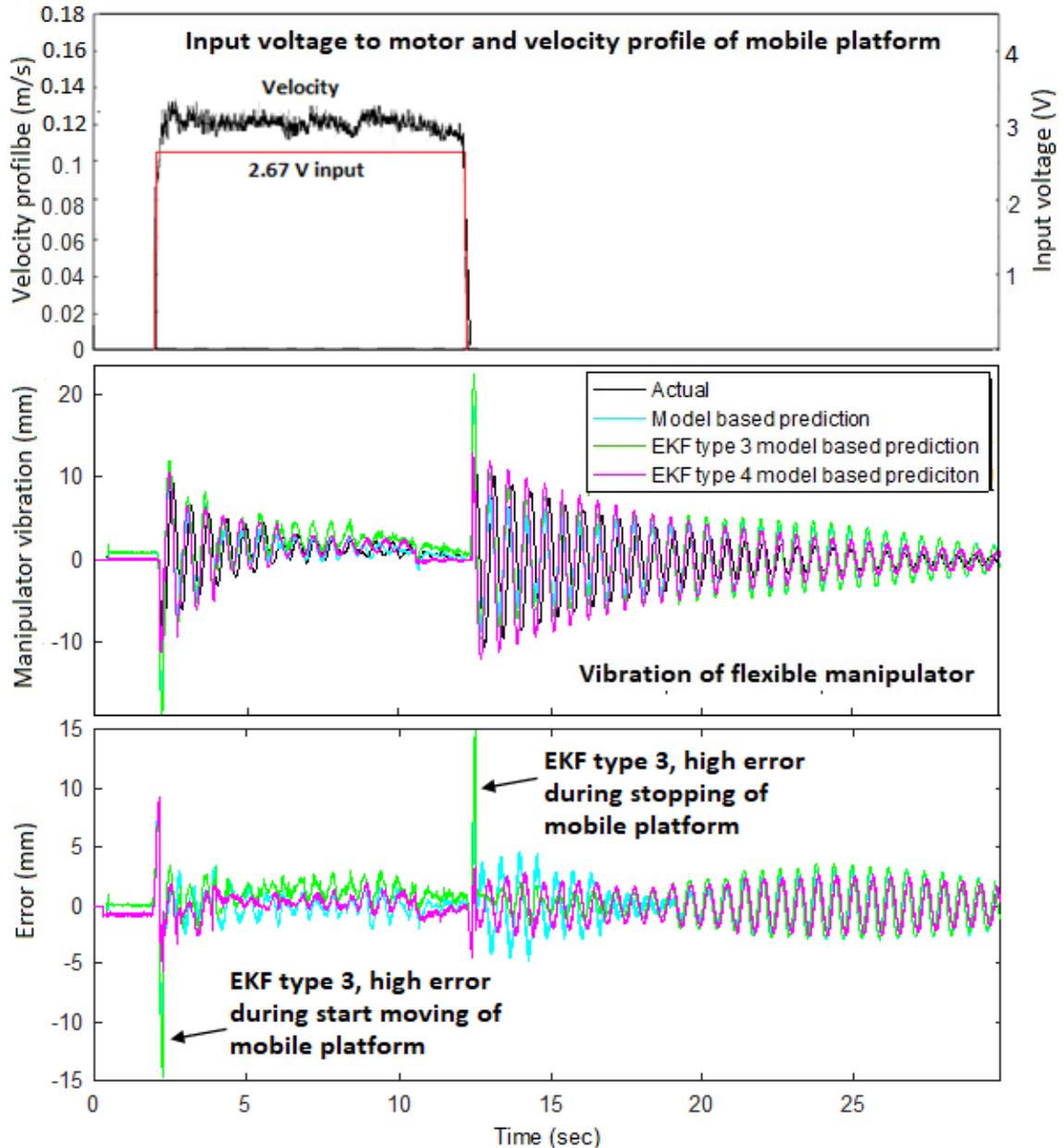
Fig. 6.36 illustrates the vibration of the mobile flexible manipulator at 2.01V input to the motor. Due to better driving torque at 2.01V input voltage, the entire mobile platform was able to moves at lower fluctuating velocity around at 0.08 m/s. It can be seen that the EKF type 3 experiences highest errors. During the instantaneous starting and instantaneous stopping of the mobile platform, and all three types of prediction methods carry an error surge with EKF type 3 highest in error. EKF type 4 carries lowest overall errors.



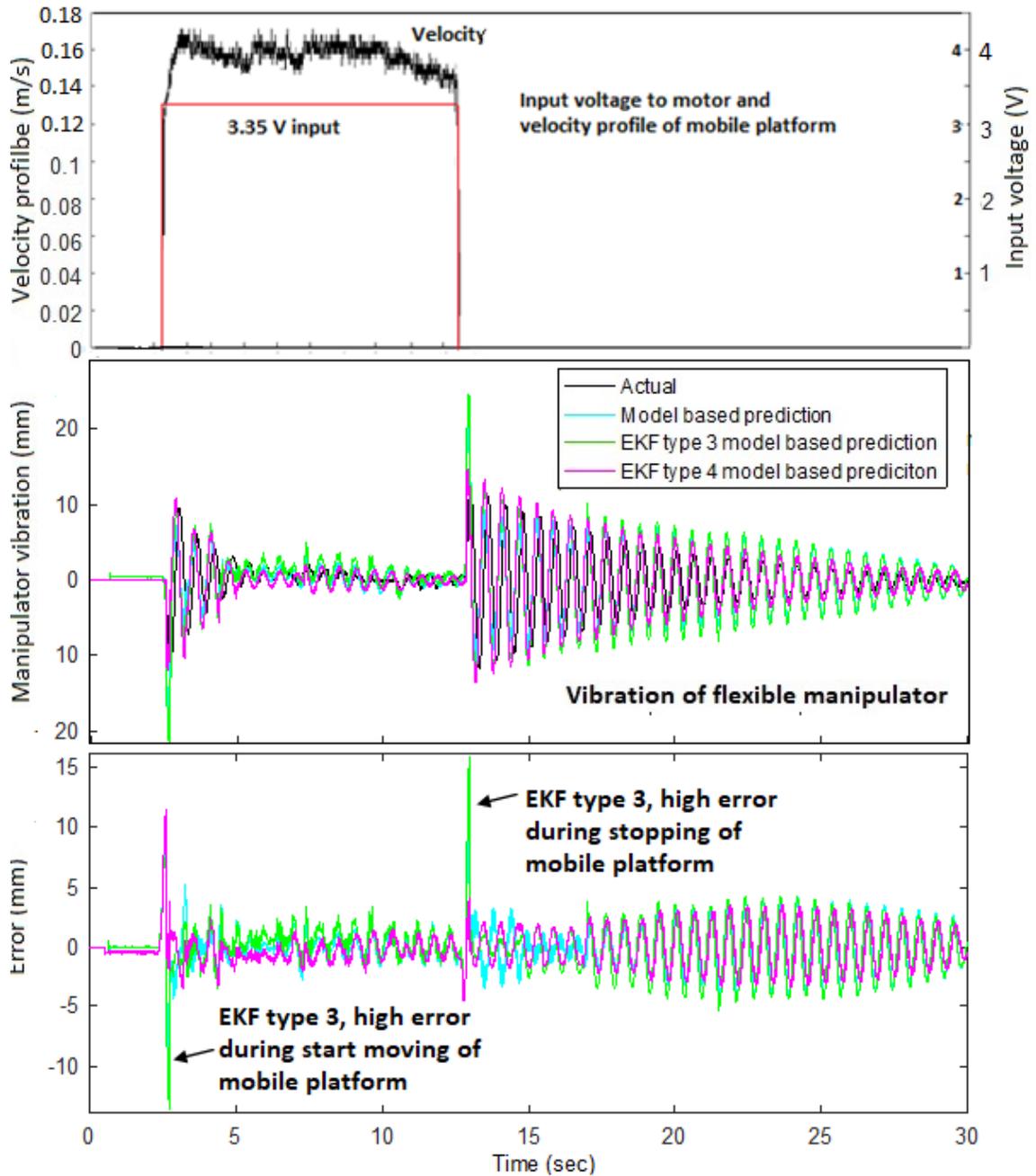
**Fig. 6.36** Comparison of model based, EKF type 3 model based prediction and EKF type 4 model based prediction for 50g payload at the manipulator tip for 2.01V input

Fig. 6.37 and 6.38 illustrate the vibration of the mobile flexible manipulator at 2.67V and 3.35V input to the motor, respectively. Due to the higher driving torque at high enough voltages the entire mobile platform moves at a more consistent velocity at around 0.12 m/s (for 2.67V) and 0.16 m/s (for 3.35V). The vibration peaked at the

instantaneous starting, and then diminishes until instantaneous stopping of the mobile platform, where the vibration peaked and then diminishes until rest. From the errors, it can be seen that EKF type 3 carries highest error, while the EKF type 4 carries lowest errors.



**Fig. 6.37** Comparison of model based, EKF type 3 model based prediction and EKF type 4 model based prediction for 50g payload at the manipulator tip for 2.67V input



**Fig. 6.38** Comparison of model based, EKF type 3 model based prediction and EKF type 4 model based prediction for 50g payload at the manipulator tip for 3.35V input

From the test results as discussed above, it can be seen that there are spikes in the errors during when the platform instantaneously starts to move and when the platform instantaneously stops. These spikes in the errors are due to the sudden change in the vibrations effected by the abrupt change in the platform velocities.

Among the three types of predictive algorithms, the EKF type 4 model-based prediction algorithm observed to be best in terms of errors. It draws from the test results that the predictions errors are within 10 mm error for type 3 EKF model based prediction, and within 5 mm errors for model based prediction and type 4 EKF model based prediction. Compared to model based prediction, type 4 EKF model based prediction has lower errors. It therefore deduced that our proposed EKF model-based predictive algorithm with modelled input delivers best vibration prediction of vibration for the tip MFLM.

**Table 6.1** Comparison of errors for the three types of predictive algorithms

Predictive type		Model based		Type 3 EKF model based		Type 4 EKF model based	
Payload	Error	RMSE (mm)	Max error (mm)	RMSE (mm)	Max error (mm)	RMSE (mm)	Max error (mm)
	Speed						
No load	Speed 1	0.0011	0.79	0.0011	1.41	0.0008	0.55
	Speed 2	0.0036	4.2	0.0033	5.08	0.0028	3.83
	Speed 3	0.0013	4.89	0.0015	14.94	0.0012	4.96
	Speed 4	0.0039	5.08	0.0042	15.09	0.0035	5.17
	Speed 5	0.0052	5.37	0.0058	13.78	0.0046	5.05
10 g	Speed 1	0.0011	0.76	0.0011	1.4	0.0008	0.51
	Speed 2	0.0036	4.15	0.0033	5.07	0.0027	3.82
	Speed 3	0.0013	4.84	0.0015	14.94	0.0012	4.95
	Speed 4	0.0039	5.07	0.0042	15.05	0.0034	5.15
	Speed 5	0.0052	5.32	0.0058	13.75	0.0046	5.01
30g	Speed 1	0.001	0.71	0.001	1.37	0.0007	0.51
	Speed 2	0.0035	4.18	0.0032	5.02	0.0027	3.83
	Speed 3	0.0012	4.8	0.0014	14.88	0.0011	4.94
	Speed 4	0.0038	5.07	0.0041	15.01	0.0034	5.13
	Speed 5	0.0052	5.3	0.0057	13.72	0.0045	5
50 g	Speed 1	0.001	0.7	0.001	1.37	0.0007	0.5
	Speed 2	0.0035	4.1	0.0032	5	0.0027	3.8
	Speed 3	0.0012	4.8	0.0014	14.87	0.0011	4.9
	Speed 4	0.0038	5	0.0041	15	0.0034	5.1
	Speed 5	0.0051	5.3	0.0057	13.58	0.0045	5
70 g	Speed 1	0.001	0.67	0.0009	1.34	0.0007	0.47
	Speed 2	0.0034	4.08	0.0032	4.96	0.0026	3.77
	Speed 3	0.0011	4.72	0.0013	14.8	0.001	4.85
	Speed 4	0.0037	5	0.0041	14.98	0.0034	5.07
	Speed 5	0.005	5.3	0.0057	14.55	0.0044	4.98
Mean RMSE		0.002948		0.003112		0.0025	

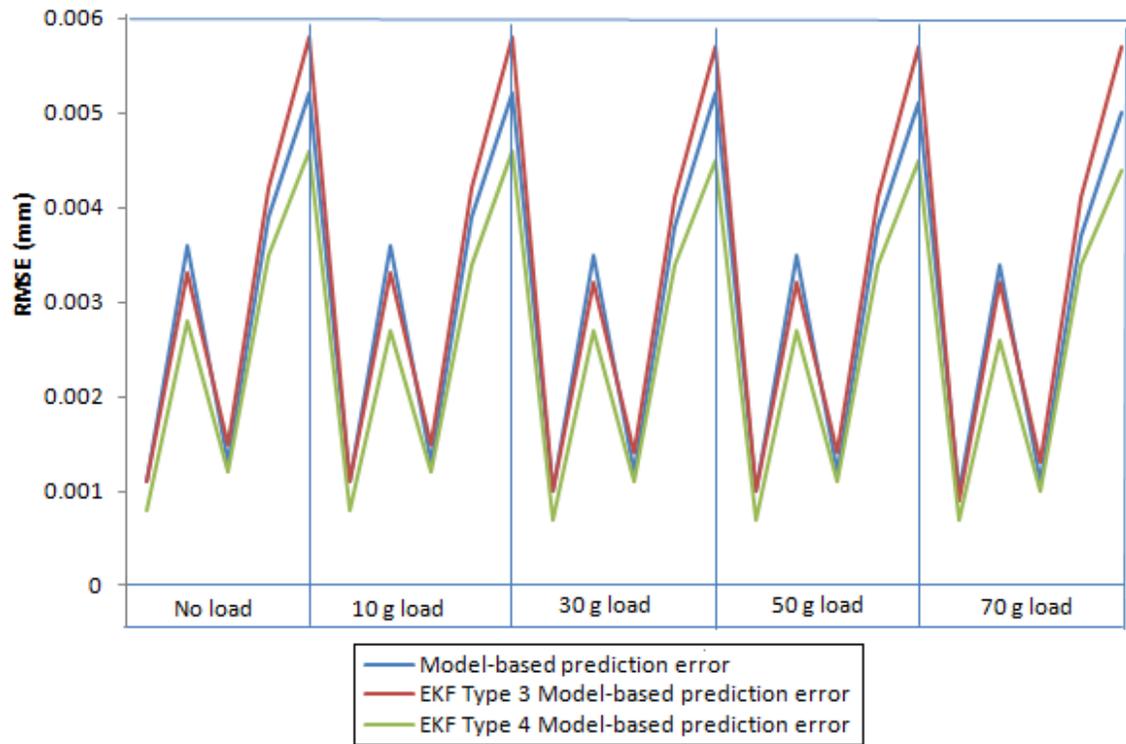
The tests for no load, 10 g load, 30 g load and 70 g load at the manipulator's tip have similar results, thus are not presented.

Table 6.1 tabulates the RMSE and max errors for the three types of model prediction methods for various payloads at the tip of the MFLM. Fig. 6.39 and 6.40 below illustrate the RMSE and maximum in graphic form. It can be seen that, for all the three types of predictive algorithm, the RMSE error increase when we increase speed of the mobile platform. When we increase the weight of the payload tip of the manipulator, there shows no increase in errors.

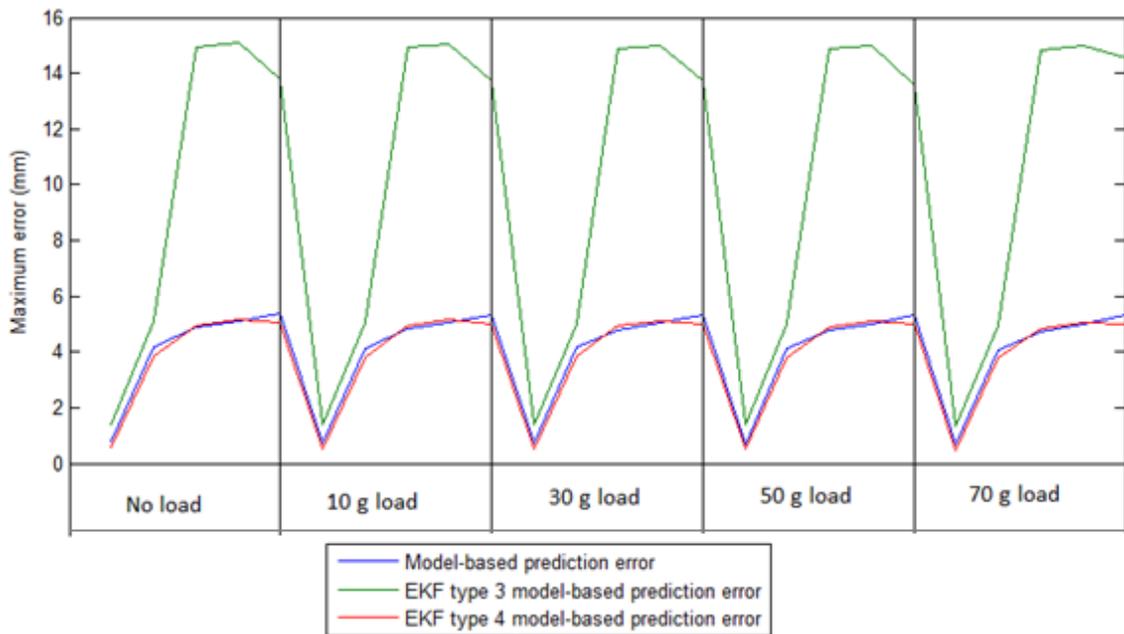
In terms of RMSE error, the EKF type 3 model-based predictive algorithm has highest errors, while the EKF type 4 model-based predictive algorithm has lowest errors. As well, for the maximum error, the EKF type 3 model-based predictive algorithm has very high errors as compared to the model-based predictive algorithm and EKF type 4 model-based predictive algorithm. This is due to the modelled input to the algorithm that helped to improve the accuracies.

We therefore concluded that the EKF type 4 model-based predictive algorithm is the best for used in predicting the vibration at the tip of the MFLM. This algorithm is based on the data from modelled MFLM when the camera data are absent and using the modelled MFLM's input to the EKF. The prediction errors are within 5 mm, which has met the aim as set out in chapter 1.

Type 3 EKF model-based predictive algorithm is based on the extrapolation of data from the camera and no MFLM's modelled input to the EKF. This type of predictive algorithm does not hold good vibration prediction of the tip of MFLM. There is as errors of above 14 mm.



**Fig. 6.39** RMSE for the three types of predictive algorithms



**Fig. 6.40** Absolute maximum error for the three types of predictive algorithms

## 6.6 Conclusion

The experimental test methodologies that were derived in the previous chapter have been carried and presented in this chapter.

The analysis of the flexible manipulator was carried out for static vibration simulations. With the given parameters for the flexible beam, it was noticed that only mode one vibration is existed in the system. Comparing the actual vibration with the simulated vibration showed that the model created in chapter 3 for flexible beam is quite accurate. However, due to the fact that the actual vibration frequency of the beam actually increases slightly as the vibration amplitude diminishes, resulting in a gradual increase in simulation when comparing the actual vibration to the modelled vibration. This phenomenon could not be modelled accurately, hence a gradual increase in error in the simulation when the vibration diminish in time. But, the first few seconds of accurate match is sufficient for use in the model predictive algorithm use in this research.

The tests on the mobile flexible link manipulator (MFLM) showed that the actual tip vibration does not match well with the simulated vibration on the MFLM. This is due to the systematic and non-systematic dynamics singularity of the entire system which made it impossible to model accurately. When disturbance was introduced into the tests (as in section 6.2.4 and 6.2.5), the simulated vibration was not comparable with the actual vibration as the model could not account the disturbances. Thus, the main drawback of the model is the inability to predict the motions when disturbance is introduced. The results hence illustrated that the simulated vibration of the manipulator and the velocity of the platform was not comparable with the actual results from the tests on the MFLM.

First, the measurement of the platform velocity has to be accurate, in order to improve the estimation of actuation input to the flexible link manipulator (FLM). Then, the tracking of the resulting vibration amplitude of the FLM is also difficult to be modelled accurately for the MFLM. Therefore, model based prediction of the MFLM motion could not be obtained accurately. Accurate sensor fusion with various sensors is thus necessary to be incorporated into prediction system to improve the prediction accuracy.

The second part of the chapter was about acquiring the signals from the sensors that are attached to the MFLM. Tests were carried out with 0.67V, 1.34V, 2.01V, 2.68V and

3.35V input to the driving motor of the mobile platform, with no payload, 10 g payload and with 30 g payload at the tip of the flexible manipulator. The results showed that the accelerometer's measurement drift as height as 120 mm. The vibration tracking using accelerometer is only accurate for up to 5 cycles of vibrations, measured at the tip of manipulator. The vibration tracking from the camera has a lag of up to 0.1 seconds, and that the capturing time of the image could not be accurately determine. Furthermore, there are frames drop which could not be detected by the algorithm. Therefore, the measurement of vibration at the tip of the mobile flexible link manipulator cannot be attained through either accelerometer or camera alone. The fusion of both sensor types would be required to provide a real-time and accurate vibration measurement.

Sensor fusion system was then tested to verify the accuracy of the system in tracking the vibration of the MLFM. EKF based sensor fusion algorithm was developed for fusing the accelerometer and camera data. Five types of EKF fusion algorithms were developed, for with and without modelled input to the EKF when camera data are not available. This includes camera extrapolation when camera data is not available. Cross-correlation function was developed to correlate the accelerometer data with the camera. Time frame windowing was also carried out to improve the accelerometer's signal tracking. Test verification showed that the model-based EKF fusion algorithm (EKF Type 4) have good match with the actual vibration of the manipulator, and has smooth outputs.

During the vibration prediction tests using the model-based prediction, EKF Type 3 and EKF Type 4, showed that EKF Type 4 (i.e. the EKF based prediction algorithm with modelled input) has better performance when compared to the other two. The prediction errors are within 5 mm for the vibration estimation, but with surging errors for the sudden moving off and sudden stopping of the platform. This showed the drawback of this algorithm when predicting vibration that subjected sudden changes. Further improvement in the prediction algorithm is thus suggested for future works.

## **7 Conclusions and future works**

This chapter presents the summary and conclusions of current works, and followed by a summary forecast of the future works.

### **7.1 Conclusions of current works**

Industrial robots have seen their successes in the manufacturing environments. As the technology advances, many other fields and domestic robots were developed. Areas such as hospital, homecare, agriculture, education, and etc., are seeing potentials for robotic requirements. It is envisioned that home helpers and other service robots will be the future trend for bringing comfort and helping the elderly and disabled. However, there are several requirements for the service robots: these are safety, low price, small size, low weight, and low power consumption.

One solution to the above requirements is to implement a mobile robot with flexible link manipulator (MFLM). This way, it is possible to develop an economical, light-weight, small-foot print and safe robot. Unfortunately, these types of robots have limitations. It was identified that the MFLMs are subjected to vibrations and fluctuations along links and joints. These resulted in increased amount of inaccuracies in the motion to be measured at the tip.

This research is therefore driven towards improving the accuracies and robustness for the motion measurement system suitable for used in MFLMs. The aim is to discover the receptiveness of applying MEMS and other low cost sensors for tracking the motion paths for this robotic system.

The literature survey identified that the commonly used sensors for flexible manipulators estimation are strain gauges, accelerometer, camera, position sensitive device and range sensors. During sensor selection accelerometer and camera were chosen for measurement of flexible link manipulator, while the webcam and encoder were chosen for estimating the motion of the mobile platform.

The accelerometer has high data rate, but with unbounded cumulative error. It was found to contain high level of noises and an offset errors. Four sets of smoothing filters were

being tested, and Lowess (local regression smoothing filter) gave best smoothing result in attenuating the noisy sensor signals. However, the filtered signals still contain moving errors (low frequency error) at static condition. These errors propagate with time and distance. During dynamic experiment at various speeds, it was found that better displacement accuracy can be obtained with higher travelling speed. It was then recommended that accelerometer be used for motion estimation for speed above 0.15m/s, but above 0.25m/s will give better accuracy.

Camera, on the other hand has have low date rate and low accuracy at detecting high speed moving object, and contain blurred image at high velocity of the moving object. But it has high accuracy for low speed detection of the moving object. The position information does not drift in time.

The model of the mobile platform and flexible link manipulator were then developed. Experiments were carried out on the motor and platform, and were then able to acquire the mechanical parameters of the motor using development models. Experimental verification of the model found that the motion out computed by the model was not accurate due to systematic and non-systematic errors that could not be modelled. Thus, sensor fusion is required to improve the real-time modelling the manipulator vibration.

While the work presented has an improvement over the model based prediction method, it is not without its limitations. First, the model is developed is for planar motion, but can be expand to three dimensional analysis with the expense of increased computational complexity. Second, there are spikes in the prediction error during the abrupt start and stop. This is due to the error in modelling for the dynamics in the sudden change of the motion of the MFLM when subjected to step inputs.

However, the experimental results shown that the EKF model-based prediction that based on modelled input yielded good prediction, which are within 5 mm prediction error. The resulting errors are better than the objective set in the chapter 1 of this research, which was set at 1 cm error. Future works would aim at minimising the spike errors to within 5 mm by improving the model towards disturbance and sudden change in motions of the MFLM.

## 7.2 Proposed future works

Further researches will be required to improve the prediction when the platform is subject to sudden change in motion, so that the error can be improved to below 5 mm and without any spike errors in the prediction.

The proposed future works will be as follows:

1. Develop sensor fusion algorithm using combination of Extended Kalman filter (EKF) and fuzzy logic, and other methods to improve the prediction accuracies and robustness.
2. The current work has not implemented the prediction algorithm on the control of the MFLM. Thus, the future works would be to develop a controller that is based on the proposed EKF model-based prediction algorithm for feedback control.
3. For the image recognition method, a better image recognition algorithm can be developed to recognise distorted images due to robot movement, such as NN based method. Future work also suggested investigating the use of webcam as a cheap means for replacing the high speed camera used in this research. Webcam for vibration estimation of the flexible manipulator has not been successful in the current research due to the huge amount of images dropped out when capturing vibration at the tip of the flexible manipulator. Also, the images captured using webcam were very blurred when capturing vibration of the flexible manipulator.

## Bibliography

- [1] B. Siciliano, and O. Khatib. *Springer Handbook of Robotics*. Springer-Verlag Berlin Heidelberg, 2016.
- [2] L. Iocchi, J. Ruiz-del-Solar, and T. van der Zant. Domestic Service Robots in the Real World. *Journal of Intelligent & Robotic Systems*, vol. 66, no. 1 - 2, pp. 183–186, Apr. 2012.
- [3] A. Flores-Abad, O. Ma, K. Pham, and S. Ulrich. A review of space robotics technologies for on-orbit servicing. *Progress in Aerospace Sciences*, vol. 68, pp. 1–26, 2014.
- [4] S. Ljungblad, J. Kotrbova, M. Jacobsson, H. Cramer, and K. Niechwiadowicz. Hospital robot at work: something alien or an intelligent colleague? *Proceedings of the ACM Conference on Computer Supported Cooperative Work*, pp. 177-186, Seattle, Washington, USA — Feb. 11 - 15, 2012.
- [5] Q. Yu, C. Yuan, Z. Fu, and Y. Zhao. An autonomous restaurant service robot with high positioning accuracy. *Industrial Robot: An International Journal*, vol. 39, no. 3, pp. 271-281, 2012.
- [6] K. Yamazaki, R. Ueda, S. Nozawa, M. Kojima, et al. Home-Assistant Robot for an Aging Society. *Proceedings of the IEEE*, vol. 100, no. 8, pp. 2429 - 2441, Aug. 2012.
- [7] J. Ma, D. Jin, Z. Wei, T. Chen, and H. Wen. Boundary Control of a Flexible Manipulator Based on a High Order Disturbance Observer with Input Saturation. *Hindawi, Shock and Vibration*, vol. 2018, Article ID 2086424, 10 pages, 2018.
- [8] M. Sayahkarajy, Z. Mohamed, and A.A. Mohd Faudzi. Review of modelling and control of flexible-link manipulators. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, April 20. 2016.
- [9] V. Feliu, F.J. Castillo, F. Ramos, and J.A. Somolinos. Robust tip trajectory tracking of a very lightweight single-link flexible arm in presence of large payload changes. *Mechatronics*, vol. 22, no. 5, pp. 594-613, Aug. 2012.
- [10] H.N. Rahimi and M. Nazemizadeh. Dynamic analysis and intelligent control techniques for flexible manipulators: a review. *Advanced Robotics*, vol. 28, no. 2, pp. 63-76, 2014.

- [11] Z. Li, C. Yang, C.Y Su, and W Ye. Adaptive fuzzy-based motion generation and control of mobile under-actuated manipulators. *Engineering Applications of Artificial Intelligence*, vol. 30, pp. 86-95, April 2014.
- [12] A.M. Abdullahi, Z. Mohamed, and M. Marwan Nafea. Resonant Control of a Single-Link Flexible Manipulator. *Jurnal Teknologi (Sciences & Engineering)*, vol. 67, no. 5 pp. 35 – 39, 15 Jan. 2014.
- [13] J. Nikolic, M. Burri, J. Rehder, S. Leutenegger, C. Huerzeler, and R Siegwart. A UAV system for inspection of industrial facilities. *IEEE Aerospace Conference*, Big Sky, MT, USA, 2-9 Mar. 2013.
- [14] C.T Kiang, A. Spowage, and C.K. Yoong. Review of control and sensor system of flexible manipulator. *Journal of Intelligent and Robotic System*, Springer, Jan. 2015.
- [15] S.G. Tzafestas. *Introduction to Mobile Robot Control*. Elsevier Inc, USA, 2014.
- [16] P.F. Dunn. *Fundamentals of Sensors for Engineering and Science*. CRC Press, Taylor & Francis Group, USA, 2012
- [17] E. Garcia-Fidalgo and A. Ortiz. Vision-based topological mapping and localization methods: A survey. *Robotics and Autonomous Systems*, vol. 64, pp. 1-20, Feb. 2015.
- [18] L.H. Erickson and S.M. LaValle. Navigation among visually connected sets of partially distinguishable landmarks. *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4829-4835, 2012.
- [19] H. Zhang, C. Zhang, W. Yang, and C. Chen. Localization and navigation using QR code for mobile robot in indoor environment, *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Zhuhai, China, 6-9 Dec. 2015.
- [20] C.L Shih and Y.T Ku. Image-Based Mobile Robot Guidance System by Using Artificial Ceiling Landmarks. *Journal of Computer and Communications*, vol 4, pp. 1-14, 2016.
- [21] G. Lan, J. Wang, and W. Chen. An improved indoor localization system for mobile robots based on landmarks on the ceiling. *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Qingdao, China, 2 Mar. 2017.

- [22] R. Dubay, M. Hassan, C. Li, and M. Charest. Finite element based model predictive control for active vibration suppression of a one-link flexible manipulator. *ISA Transactions*, vol. 53, no. 5, pp. 1609-1619, Sep. 2014.
- [23] X. Jiang, A. Konno, and M. Uchiyama. Wrist camera-based vibration suppression control for a flexible manipulator. *Advance Robotics*, vol. 25, no. 6–7, pp. 805 – 823, 2011.
- [24] A. Vandini, A. Salerno, C.J. Payne, and G.Z. Yang. Vision-based motion control of a flexible robot for surgical applications. *IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, China, 29 Sep. 2014.
- [25] A. Suarez, G. Heredia, and A. Ollero. Vision-Based Deflection Estimation in an Anthropomorphic, Compliant and Lightweight Dual Arm. *ROBOT 2017: Third Iberian Robotics Conference*. Advances in Intelligent Systems and Computing, vol. 694. Springer, Cham, pp 332-344, 21 Dec. 2017.
- [26] L. Lu, Z. Chen, B. Yao, and Q. Wang. A two-loop performance-oriented tip-tracking control of a linear-motor-driven flexible beam system with experiments. *IEEE Transactions on Industrial Electronics*, vol.60, no.3, pp.1011,1022, Mar. 2013.
- [27] J.G. Webster and H Eren. *Measurement, instrumentation, and sensors handbook: spatial, mechanical, thermal, and radiation measurement, 2nd edn*. CRC Press, Taylor and Francis Group, USA, 2014.
- [28] N.Y. Ko and T.Y. Kuc. Fusing Range Measurements from Ultrasonic Beacons and a Laser Range Finder for Localization of a Mobile Robot. *Sensors 2015*, vol. 15, no. 5, pp. 11050-11075, 2015.
- [29] C Wang, W Chen, M Tomizuka. Robot end-effector sensing with position sensitive detector and inertial sensors. *IEEE International Conference on Robotics and Automation (ICRA)*, USA, 28 June 2012.
- [30] M.T. Hussein. A review on vision-based control of flexible manipulators. *Journal Advanced Robotics*, vol. 29, no. 24, 2015.
- [31] M.S. Nixon, and A.S. Aguado. *Feature Extraction and Image Processing for Computer Vision, 3rd edn*. Elsevier Science, 2012.

- [32] H. Zhao and Z. Wang. Motion measurement using inertial sensors, ultrasonic sensors, and magnetometers with extended Kalman filter for data fusion. *IEEE Sensors Journal*, Vol. 12, no. 5, May 2012.
- [33] Y. Li, S. Tong, and T. Li. Adaptive fuzzy output feedback control for a single-link flexible robot manipulator driven DC motor via backstepping. *Nonlinear Analysis: Real World Applications*, vol. 14, no. 1, pp. 483–494, 2013.
- [34] F. Jacob. *Handbook of Modern Sensors: Physics, Designs, and Applications*, 5<sup>th</sup> edn. Springer International Publishing, Switzerland, 2016.
- [35] D.K. Shaeffer. MEMS inertial sensors: A tutorial overview. *IEEE Communications Magazine*, vol. 51, no. 4, Apr. 2013.
- [36] V. Kempe. *Inertial MEMS: Principles and Practice*. Cambridge University Press, UK, 2011.
- [37] T.B. Jones and N.G. Nenadic. *Electromechanics and MEMS*. Cambridge University Press, UK, 2013.
- [38] N. Ganganath and H. Leung. Mobile robot localization using odometry and Kinect sensor. *IEEE International Conference on Emerging Signal Processing Applications (ESPA)*, Las Vegas, NV, USA, 12-14 Jan. 2012.
- [39] P. Axelsson, Mikael Norrlöf. Method to Estimate the Position and Orientation of a Triaxial Accelerometer Mounted to an Industrial Manipulator. *10th IFAC Symposium on Robot Control International Federation of Automatic Control, Dubrovnik, Croatia*, vol. 45, no. 22, pp. 283 - 288, Sep. 5-7, 2012..
- [40] R. Mautz and S. Tilch. Survey of optical indoor positioning systems. *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 21-23, Sept. 2011.
- [41] H. Maître. From Photon to Pixel: *The Digital Camera Handbook*. John Wiley & Sons, Incorporated, USA, 27 Oct. 2015.
- [42] Tim Moynihan. CMOS Is Winning the Camera Sensor Battle, and Here's Why. *PC World*, techhive.com. Dec. 29, 2011.
- [43] J. Balsam, H.A. Bruck, Y. Kostov, and A. Rasooly. Image stacking approach to increase sensitivity of fluorescence detection using a low cost complementary

- metal-oxide-semiconductor (CMOS) webcam. *Sensors and Actuators B: Chemical*, vol. 171–172, pp. 141-147, Aug. – Sep. 2012.
- [44] T.D. Tan, N.T. Anh. Three-axis piezoresistive accelerometer with uniform axial sensitivities. *Second International Conference on Intelligent Systems, Modelling and Simulation (ISMS)*, Kuala Lumpur, Malaysia, 2011.
- [45] A. Noureldin, T.B. Karamat, and J. Georgy. *Fundamentals of Inertial Navigation, Satellite-based Positioning and their integration*. Springer Heidelberg, NW, 2013.
- [46] L. Cheng, C.D. Wu, and Y.Z. Zhang. Indoor robot localization based on wireless sensor networks. *IEEE Transactions on Consumer Electronics*, vol. 57, no. 3, Aug. 2011.
- [47] G.G. Anagnostopoulos and M. Deriaz. Accuracy enhancements in indoor localization with the weighted average technique. *SENSORCOMM, 8th International Conference on Sensor Technologies and Applications*, 2014.
- [48] Q. Dong and W. Dargie. Evaluation of the reliability of RSSI for indoor localization. *International Conference on wireless Communications in Unusual and Confined Areas (ICWCUCA)*, Clermont Ferrand, France, 28-30, Aug. 2012.
- [49] C.K. Chui and G. Chen. *Kalman Filtering with Real-Time Applications*, 5th edn. Springer International Publishing. 2017.
- [50] M. Doumiati, A. Charara, A. Victorino, and D. Lechner. *Vehicle Dynamics Estimation Using Kalman Filtering: Experimental Validation*. John Wiley & Sons, Inc., USA, 2013.
- [51] F. Auger, M. Hilairet, and J.M. Guerrero. Industrial Applications of the Kalman Filter: A Review. *IEEE Transactions on Industrial Electronics*, vol. 60, no. 12, Dec. 2013.
- [52] B. Farhang-Boroujeny. *Adaptive Filters: Theory and Applications, 2nd edn*. John Wiley & Sons, Ltd, UK, 2013.
- [53] R.C. Luo and C.C. Lai. Enriched indoor map construction based on multisensor fusion approach for intelligent service robot. *IEEE Transactions on Industrial Electronics*, 2012.
- [54] J.F. Vasconcelos, B. Carneira, C. Silvestre, P. Oliveira, and P. Batista. Discrete-Time Complementary Filters for Attitude and Position Estimation: Design,

- Analysis and Experimental Validation. *IEEE Transactions On Control Systems Technology*, vol. 19, no. 1, Jan. 2011.
- [55] P. Marantos, Y. Koveos, and K.J. Kyriakopoulos. UAV State Estimation Using Adaptive Complementary Filters. *IEEE Transactions on Control Systems Technology*, vol. 24, no. 4, pp. 1214 - 1226, Jul. 2016.
- [56] L. Benziane, A. El Hadri, A. Seba, A. Benallegue, and Y. Chitour. Attitude Estimation and Control Using Linear like Complementary Filters: Theory and Experiment. *IEEE Transactions on Control Systems Technology*, vol. 24, no. 6, Nov. 2016.
- [57] J.M. Pak, C.K. Ahn, Y.S. Shmaliy, and M.T. Lim. Improving Reliability of Particle Filter-Based Localization in Wireless Sensor Networks via Hybrid Particle/FIR Filtering. *IEEE Transactions on Industrial Informatics*, vol. 11, no. 5, pp. 1089 – 1098, Oct. 2015.
- [58] H. Nurminen, A. Ristimaki, S. Ali-Loytty, and R. Piche. Particle filter and smoother for indoor localization. *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Montbeliard-Belfort, France, 19 May 2014.
- [59] R. Belohlavek and G. J. Klir. *Concepts and Fuzzy Logic*. MIT Press, Cambridge, England, 2011.
- [60] G. Cicirelli, A. Milella, and D. Di Paola. RFID tag localization by using adaptive neuro-fuzzy inference for mobile robot applications. *Industrial Robot: An International Journal*, vol. 39, no. 4, pp.340-348, 2012.
- [61] M.A. Khanesar, E. Kayacan, M. Teshnehlab, and O. Kaynak. Extended Kalman Filter Based Learning Algorithm for Type-2 Fuzzy Logic Systems and Its Experimental Evaluation. *IEEE Transactions on Industrial Electronics*, vol. 59, no. 11, pp. 4443 – 4455, Nov. 2012.
- [62] A. El-Sherbiny, M.A. Elhosseini, A.Y. Haikal. A comparative study of soft computing methods to solve inverse kinematics problem. *Ain Shams Engineering Journal*, 2017.
- [63] D. Satchidananda, G. Susmita, and C. Sung-Bae. *Integration of Swarm Intelligence and Artificial Neural Network*. World Scientific Publishing Co Pte Ltd, 2014.

- [64] A.V. Duka. ANFIS Based Solution to the Inverse Kinematics of a 3DOF Planar Manipulator. *Procedia Technology*, vol. 19, pp. 526-533, 2015.
- [65] D. Andre, A. Nuhic, T. Soczka-Guth, and D.U. Sauer. Comparative study of a structured neural network and an extended Kalman filter for state of health determination of lithium-ion batteries in hybrid electric vehicles. *Engineering Applications of Artificial Intelligence*, vol. 26, pp. 951–961, 2013.
- [66] A.M. Hasan, K. Samsudin, and A.R. Ramli. Optimizing of ANFIS for estimating INS error during GPS outages. *Journal of the Chinese Institute of Engineers*, Vol. 34, no. 7, 2011.
- [67] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, vol. 15, pp. 1929-1958, 2014.
- [68] C. Sun, W. He, and J. Hong. Neural Network Control of a Flexible Robotic Manipulator Using the Lumped Spring-Mass Model. *IEEE transactions on systems, man, and cybernetics: systems*, vol. 47, no. 8, Aug. 2017.
- [69] H. Esfandiari, M.H. Korayem, and M. Haghpanahi. Optimal Trajectory Planning for Flexible Mobile Manipulators under Large Deformation Using Meta-heuristic Optimization Methods. *International Journal of Advanced Design and Manufacturing Technology*, Mar. 2016.
- [70] S. Wei and Y. Wu. Modeling and finite-time tracking control for mobile manipulators with affine and holonomic constraints. *Journal of Systems Science and Complexity*, vol. 29, no. 3, pp. 589-601, 2016.
- [71] M. Abdolvand and M.H. Fatehi. Model-base Predictive Control for Vibration Suppression of a Flexible Manipulator. *UKACC International Conference on Control*, Cardiff, UK, 3-5 Sep. 2012.
- [72] P. Boscariol and V. Zanutto. Design of a controller for trajectory tracking for compliant mechanisms with effective vibration suppression. *Robotica*, vol. 30, no. 1, pp. 15-29, Jan. 2012.
- [73] G.B. Avanzini, A.M. Zanchettin, and P. Rocco. Reactive constrained model predictive control for redundant mobile manipulators. *Intelligent Autonomous*

- Systems 13. Advances in Intelligent Systems and Computing*, vol. 302. Springer, Cham, pp. 1301-1314, 2016.
- [74] M. Bakhti. Active vibration control of a flexible manipulator using model predictive control and kalman optimal filtering. *International Journal of Engineering Science and Technology (IJEST)*, vol. 5, no.01, Jan. 2013.
- [75] X. Wei and S. Liu. Computational effective predictive end-point trajectory control of flexible manipulators with measureable feedback lag. *Research Journal of Applied Sciences, Engineering and Technology*, vol. 4, no. 20, pp. 3875-3884, 2012.
- [76] H.R. Heidari, M.H. Korayem, M. Haghpanahi, and V.F. Batlle. Optimal trajectory planning for flexible link manipulators with large deflection using a new displacements approach. *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 72. no. 3 – 4, pp. 287 – 300, Dec. 2013.
- [77] M.P.P. Reddy and J. Jacob. Vibration control of flexible link manipulator using sdre controller and kalman filtering. *Studies in Informatics and Control*, vol. 26, no. 2, pp. 143-150, Jun. 2017.
- [78] Y. Ding and X. Xiao. Speed Control and Resonance Suppression of Flexible Joint System based on singular perturbation method and Kalman Filter. *IECON 2016 - 42nd Annual Conference of the IEEE on Industrial Electronics Society*, Florence, Italy, 22 Dec. 2016.
- [79] H. Ahmad and T. Namerikawa. Extended Kalman filter-based mobile robot localization with intermittent measurements. *Systems Science & Control Engineering: An Open Access Journal*, vol. 1, No. 1, pp. 113–126, 2013.
- [80] H Tang and J. Wan. Robust adaptive control based on neural networks for mobile manipulators. *IEEE International Conference on Computer and Communications (ICCC)*, Chengdu, China, 14-17 Oct. 2016.
- [81] C. Najjian, Y. Hao, H. Xiangdong, A. Changsheng, T. Chenglong, and L. Xiangkui. Adaptive robust control of the mobile manipulator based on neural network. *IEEE 11th Conference on Industrial Electronics and Applications (ICIEA)*, pp. 2425 - 2430, Hefei, China, 24 Oct. 2016.

- [82] C.C. Morris, and R.M. Stark. *Fundamentals of Calculus*. John Wiley & Sons, Incorporated, 2015.
- [83] S. Stančin, S. Tomažič. Time- and Computation-Efficient Calibration of MEMS 3D Accelerometers and Gyroscopes. *Sensors 2014*, vol. 14, no. 8, pp. 14885-14915, 13 Aug. 2014.
- [84] A. Hughes, and W. Drury. *Electric Motors and Drives: Fundamentals, Types and Applications, 4<sup>th</sup> edn*. Elsevier Science, Waltham, USA, 2013
- [85] I. Virgala and M. Kelemen. Experimental friction identification of a dc motor. *International Journal of Mechanics and Applications*, vol. 3, no. 1, pp. 26–30, 2013.
- [86] I. Virgala, P. Frankovský, and M. Kenderová. Friction effect analysis of a DC motor. *American Journal of Mechanical Engineering*, vol. 1, no.1, pp. 1-5, 2013.
- [87] G.S. Chen. *Handbook of Friction-Vibration Interactions*. Cambridge: Woodhead Publishing, UK. 2014.
- [88] M. Meywerk. *Vehicle Dynamics*. John Wiley & Sons, Incorporated, UK. 2015.
- [89] A. Khajepour, M.S. Fallah, and A. Goodarzi. *Electric and Hybrid Vehicles: Technologies, Modeling and Control - a Mechatronic Approach*. John Wiley & Sons, Incorporated, UK. 2014.
- [90] W. Chen, Q. Wang, H. Xiao, L. Zhao, and M. Zhu. *Integrated Vehicle Dynamics and Control*. John Wiley & Sons, Incorporated, Singapore, 2016.
- [91] R.C. Dorf, and R.H. Bishop. *Modern Control Systems*, 13<sup>th</sup> Edn. Pearson, 2017.
- [92] W. Ding and Y. Shen. Analysis of Transient Deformation Response for Flexible Robotic Manipulator Using Assumed Mode Method. *2nd Asia-Pacific Conference on Intelligent Robot Systems (ACIRS)*, Wuhan, China, 16-18 Jun. 2017.
- [93] S. Moberg, E. Wernholt, S. Hanssen, and T. Brogårdh. Modeling and Parameter Estimation of Robot Manipulators Using Extended Flexible Joint Models. *Journal of Dynamic Systems, Measurement, and Control*, vol. 136, no. 3, Feb 10, 2014.
- [94] W. He, Y. Ouyang, and J. Hong. Vibration Control of a Flexible Robotic Manipulator in the Presence of Input Deadzone. *IEEE Transactions on Industrial Informatics*, vol. 13, no. 1, FEB. 2017.

- [95] R.D. Blevins. *Formulas for Dynamics, Acoustics and Vibration*. John Wiley & Sons, Inc., UK, 2016.
- [96] J.S. Wu. *Analytical and Numerical Methods for Vibration Analysis*. John Wiley & Sons, Inc., Singapore, 2013.
- [97] W. Zhang. The impulse spectrum method for vibration suppression of a flexible multilink robot arm. *Journal of Vibration and control*, Jun. 19, 2017.
- [98] A.L. Galloway. *Mechanical Vibrations: Types, Testing and Analysis*. Galloway, Nova Science Publishers, 2011
- [99] A. Abe. Trajectory planning for residual vibration suppression of a two-link rigid-flexible manipulator considering large deformation. *Mechanism and Machine Theory*, vol. 44, no. 9, pp. 1627-1639, Sep. 2009.
- [100] C. Lalanne. *Mechanical Vibration and Shock Analysis, Sinusoidal Vibration, 3<sup>rd</sup> edn*. ISTE Ltd, London, UK, 2014.
- [101] H. Esfandiari, M. Haghpanahi, and M. H. Korayem. Large deformation modeling of flexible manipulators to determine allowable load. *Structural Engineering and Mechanics*, vol. 62, no. 5, pp. 619-629, Jun. 2017.
- [102] K. Lochan, B. K. Roy, and B. Subudhi. SMC Controlled Chaotic Trajectory Tracking of Two-Link Flexible Manipulator with PID Sliding Surface. *IFAC (International Federation of Automatic Control)*, vol. 49, no. 1, pp. 219-224, 2016.
- [103] T. Chai and R. R. Draxler. Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature. *Geoscientific Model Development Discussions*. vol. 7, no. 1, pp. 1525–1534, 2014.
- [104] S. Särkkä. *Bayesian Filtering and Smoothing*. Cambridge University Press, UK, 2013.
- [105] G Nützi, S. Weiss, D. Scaramuzza, and R. Siegwart. Fusion of IMU and Vision for Absolute Scale Estimation in Monocular SLAM. *Journal of Intelligent and Robotic System*, vol 61, no. 1 – 4, pp. 287–299, (2011).
- [106] G. Dubus, On-line estimation of time varying capture delay for vision-based vibration control of flexible manipulators deployed in hostile environments. Intelligent Robots and Systems (IROS), *IEEE/RSJ International Conference*, pp. 3765–3770, 2010.

- [107] S. Mallikarjunaiah and S.N. Reddy. Design of PID Controller for Flexible Link Manipulator. *International Journal of Engineering Research and Applications*, vol. 3, no. 1, pp.1207-1212, 2013.
- [108] H. Azami, K. Mohammadi, B. Bozorgtabar. An Improved Signal Segmentation Using Moving Average and Savitzky-Golay Filter. *Journal of Signal and Information Processing*, vol. 3, pp. 39-44, 2012.
- [109] Loader C. *Smoothing: Local Regression Techniques*. In: Gentle J., Härdle W., Mori Y. (eds) *Handbook of Computational Statistics*. Springer Handbooks of Computational Statistics. Springer, Berlin, Heidelberg, 2012.

## Appendix A     Reviews of encoders

Encoders are electro-mechanical devices that provide position information through bitstreams of 1 and 0. There are two types of encoders: linear and rotary encoders with wide variability in sizes, feature, and capabilities. For a typical incremental pulse rotary encoder [27], angular acceleration can be estimated as  $\alpha$  (rad/s<sup>2</sup>),

$$\alpha = 2J(n_1 - n_2)/PT^2 \tag{A.1}$$

where  $n_1$  and  $n_2$  pulses are counted in two successive intervals of duration  $T$ . When a high pulse rate,  $P$  is pulses per revolution.

### A.1    Types of encoders

Encoders can be classified into mechanical contact, optical (photoelectric), and magnetic types [27]. In mechanical contact encoders, the brushes or finger sensors read the disk. For non-contact magnetic encoders, the disk is magnetically coded. The magneto-resistive sensor detects changes in magnetic flux, and interprets the magnetic code as a series of on and off states.

In an optical (photoelectric) encoder, a lamp, LED or laser light source is projected through thin slits on the disk. The disk can be glass, plastic with thin lines etched into a coating or plating, or a metal disk with etched line openings. A photoreceptor (optical sensors) on the opposite side of the disk detects the light and converts it into series of electrical pulses.

There are absolute encoders and incremental encoders. Incremental encoders are the simplest and lower cost than absolute encoders. In an incremental encoder, there are evenly spaced opaque radial lines on the surface of the code disk. When it rotates light source passes through the transparent segments generating a series of square wave outputs [34]. The controller counts the pulses to determine the speed or its position relative to the previous position. The resolution is determined by the number of lines on the disk.

In an absolute encoder, the code disk consists of multiple concentric tracks of opaque and transparent patterns on the coded disc, which provide a unique output for every

position [34]. Gray coder is most commonly used, where only one single bit changes with the transition from one measuring step to the next. For very high resolution natural binary coders are also frequently used. Each track has its own independent photo-detector for providing the unique position output. The number of tracks determines the binary bit-resolution of the encoder. A 12-bit absolute encoder will then has 12 tracks.

## **A.2 Comparison of encoders**

Among the three types of encoders, the magnetic and optical encoders are commonly used, while optical encoders are most common due to their high accuracy, high reliability high speeds, low costs and high resolutions [34]. Some offer more than 1 million counts per rotation. Magnetic encoders offer good resolution, can operate in harsh and dirty conditions, and have low power consumptions. However, they cannot achieve very high speeds.

As for the incremental and absolute encoder types, the incremental encoders provide more resolution at a lower cost, and they are good for measuring speed and acceleration. The main problem is that they cannot determine their location at start-up, thus the position count is lost when there is a power failure or some other interruption. Thus a homing sequence to find a reference pulse is required during initialization.

Absolute encoders always know their exact position through the information stored in nonvolatile memory, in case of a power failure they can regain their position information, thus homing is not required. But, they are more complex and more expensive than incremental encoders, further they are not as suitable for measuring speed or acceleration.

## Appendix B Datasheet for sensors

### B.1 Accelerometer datasheet

Source: SparkFun Electronics for Freescale Semiconductor (MMA7260Q)

Website: <https://www.sparkfun.com/datasheets/Accelerometers/MMA7260Q-Rev1.pdf>

**Freescale Semiconductor**  
Technical Data

MMA7260Q  
Rev 1, 06/2005

### ±1.5g - 6g Three Axis Low-g Micromachined Accelerometer

The MMA7260Q low cost capacitive micromachined accelerometer features signal conditioning, a 1-pole low pass filter, temperature compensation and g-Select which allows for the selection among 4 sensitivities. Zero-g offset full scale span and filter cut-off are factory set and require no external devices. Includes a Sleep Mode that makes it ideal for handheld battery powered electronics.

#### Features

- Selectable Sensitivity (1.5g/2g/4g/6g)
- Low Current Consumption: 500  $\mu$ A
- Sleep Mode: 3  $\mu$ A
- Low Voltage Operation: 2.2 V – 3.6 V
- 6mm x 6mm x 1.45mm QFN
- High Sensitivity (800 mV/g @1.5 g)
- Fast Turn On Time
- High Sensitivity (1.5 g)
- Integral Signal Conditioning with Low Pass Filter
- Robust Design, High Shocks Survivability
- Pb-Free Terminations
- Environmentally Preferred Package
- Low Cost

#### Typical Applications

- HDD MP3 Player : Freefall Detection
- Laptop PC : Freefall Detection, Anti-Theft
- Cell Phone : Image Stability, Text Scroll, Motion Dialing, E-Compass
- Pedometer : Motion Sensing
- PDA : Text Scroll
- Navigation and Dead Reckoning : E-Compass Tilt Compensation
- Gaming : Tilt and Motion Sensing, Event Recorder
- Robotics : Motion Sensing

ORDERING INFORMATION			
Device Name	Temperature Range	Case No.	Package
MMA7260Q	- 20 to +85°C	1622-01	QFN-16, Tube
MMA7260QR2	- 20 to +85°C	1622-01	QFN-16, Tape & Reel

**MMA7260Q**

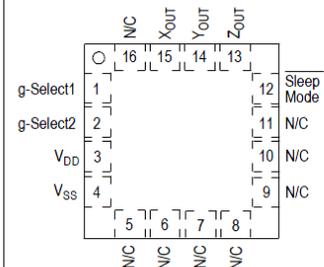
**MMA7260Q: XYZ AXIS  
ACCELEROMETER**  
±1.5g/2g/4g/6g

**Bottom View**



**16 LEAD  
QFN  
CASE 1622-01**

**Top View**



**Figure 1. Pin Connections**

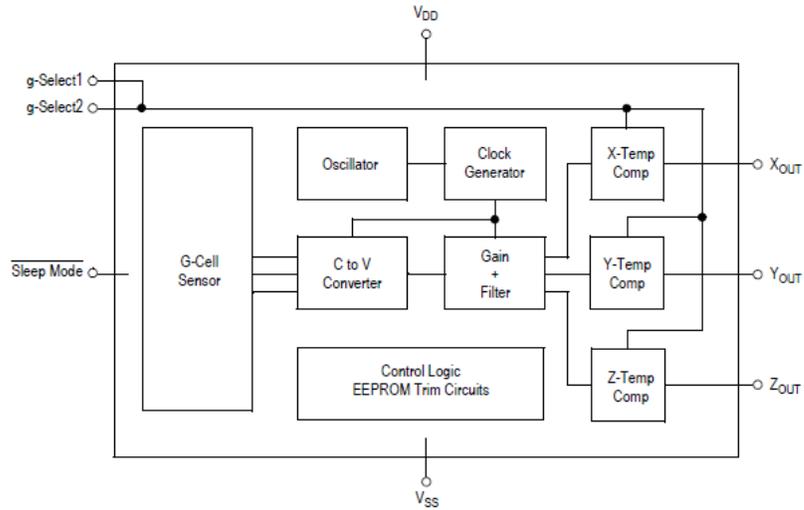


Figure 2. Simplified Accelerometer Functional Block Diagram

Table 1. Maximum Ratings

(Maximum ratings are the limits to which the device can be exposed without causing permanent damage.)

Rating	Symbol	Value	Unit
Maximum Acceleration (all axis)	$g_{max}$	$\pm 2000$	g
Supply Voltage	$V_{DD}$	-0.3 to +3.6	V
Drop Test <sup>(1)</sup>	$D_{drop}$	1.8	m
Storage Temperature Range	$T_{stg}$	-40 to +125	°C

1. Dropped onto concrete surface from any axis.

#### CTRO STATIC DISCHARGE (ESD)

**WARNING: This device is sensitive to electrostatic discharge.**

Although the Freescale accelerometer contains internal 2000 V ESD protection circuitry, extra precaution must be taken by the user to protect the chip from ESD. A charge of over 2000 volts can accumulate on the human body or associated test equipment. A charge of this magnitude can

alter the performance or cause failure of the chip. When handling the accelerometer, proper ESD precautions should be followed to avoid exposing the device to discharges which may be detrimental to its performance.

**Table 2. Operating Characteristics**Unless otherwise noted:  $-20^{\circ}\text{C} \leq T_A \leq 85^{\circ}\text{C}$ ,  $2.2\text{ V} \leq V_{\text{DD}} \leq 3.6\text{ V}$ , Acceleration = 0g, Loaded output<sup>(1)</sup>

Characteristic	Symbol	Min	Typ	Max	Unit
Operating Range <sup>(2)</sup>					
Supply Voltage <sup>(3)</sup>	$V_{\text{DD}}$	2.2	3.3	3.6	V
Supply Current	$I_{\text{DD}}$	—	500	800	$\mu\text{A}$
Supply Current at Sleep Mode <sup>(4)</sup>	$I_{\text{DD}}$	—	3	10	$\mu\text{A}$
Operating Temperature Range	$T_A$	-20	—	+85	$^{\circ}\text{C}$
Acceleration Range, X-Axis, Y-Axis, Z-Axis					
g-Select1 & 2: 00	$g_{\text{FS}}$	—	$\pm 1.5$	—	g
g-Select1 & 2: 10	$g_{\text{FS}}$	—	$\pm 2.0$	—	g
g-Select1 & 2: 01	$g_{\text{FS}}$	—	$\pm 4.0$	—	g
g-Select1 & 2: 11	$g_{\text{FS}}$	—	$\pm 6.0$	—	g
Output Signal					
Zero g ( $T_A = 25^{\circ}\text{C}$ , $V_{\text{DD}} = 3.3\text{ V}$ ) <sup>(5)</sup>	$V_{\text{OFF}}$	1.485	1.65	1.815	V
Zero g	$V_{\text{OFF}}, T_A$	—	$\pm 2$	—	$\text{mg}/^{\circ}\text{C}$
Sensitivity ( $T_A = 25^{\circ}\text{C}$ , $V_{\text{DD}} = 3.3\text{ V}$ )					
1.5g	$S_{1.5g}$	740	800	860	$\text{mV}/\text{g}$
2g	$S_{2g}$	555	600	645	$\text{mV}/\text{g}$
4g	$S_{4g}$	277.5	300	322.5	$\text{mV}/\text{g}$
6g	$S_{6g}$	185	200	215	$\text{mV}/\text{g}$
Sensitivity	$S, T_A$	—	$\pm 0.03$	—	$\%/^{\circ}\text{C}$
Bandwidth Response					
XY	$f_{-3\text{dB}}$	—	350	—	Hz
Z	$f_{-3\text{dB}}$	—	150	—	Hz
Noise					
RMS (0.1 Hz – 1 kHz) <sup>(4)</sup>	$\eta_{\text{RMS}}$	—	4.7	—	$\text{mVrms}$
Power Spectral Density RMS (0.1 Hz – 1 kHz) <sup>(4)</sup>	$\eta_{\text{PSD}}$	—	350	—	$\mu\text{g}/\sqrt{\text{Hz}}$
Control Timing					
Power-Up Response Time <sup>(6)</sup>	$t_{\text{RESPONSE}}$	—	1.0	2.0	ms
Enable Response Time <sup>(7)</sup>	$t_{\text{ENABLE}}$	—	0.5	2.0	ms
Sensing Element Resonant Frequency					
XY	$f_{\text{GCELL}}$	—	6.0	—	kHz
Z	$f_{\text{GCELL}}$	—	3.4	—	kHz
Internal Sampling Frequency	$f_{\text{CLK}}$	—	11	—	kHz
Output Stage Performance					
Full-Scale Output Range ( $I_{\text{OUT}} = 30\ \mu\text{A}$ )	$V_{\text{FSO}}$	$V_{\text{SS}}+0.25$	—	$V_{\text{DD}}-0.25$	V
Nonlinearity, $X_{\text{OUT}}$ , $Y_{\text{OUT}}$ , $Z_{\text{OUT}}$	$\text{NL}_{\text{OUT}}$	-1.0	—	+1.0	%FSO
Cross-Axis Sensitivity <sup>(8)</sup>	$V_{\text{XY, XZ, YZ}}$	—	—	5.0	%

1. For a loaded output, the measurements are observed after an RC filter consisting of a 1.0 k $\Omega$  resistor and a 0.1  $\mu\text{F}$  capacitor to ground.

2. These limits define the range of operation for which the part will meet specification.

3. Within the supply range of 2.2 and 3.6 V, the device operates as a fully calibrated linear accelerometer. Beyond these supply limits the device may operate as a linear device but is not guaranteed to be in calibration.

4. This value is measured with g-Select in 1.5g mode.

5. The device can measure both + and – acceleration. With no input acceleration the output is at midsupply. For positive acceleration the output will increase above  $V_{\text{DD}}/2$ . For negative acceleration, the output will decrease below  $V_{\text{DD}}/2$ .

6. The response time between 10% of full scale Vdd input voltage and 90% of the final operating output voltage.

7. The response time between 10% of full scale Sleep Mode input voltage and 90% of the final operating output voltage.

8. A measure of the device's ability to reject an acceleration applied 90° from the true axis of sensitivity.

Figure 4. Pinout Description

Table 4. Pin Descriptions

Pin No.	Pin Name	Description
1	g-Select1	Logic input pin to select g level.
2	g-Select2	Logic input pin to select g level.
3	V <sub>DD</sub>	Power Supply Input
4	V <sub>SS</sub>	Power Supply Ground
5 - 7	N/C	No internal connection. Leave unconnected.
8 - 11	N/C	Unused for factory trim. Leave unconnected.
12	Sleep Mode	Logic input pin to enable product or Sleep Mode.
13	Z <sub>OUT</sub>	Z direction output voltage.
14	Y <sub>OUT</sub>	Y direction output voltage.
15	X <sub>OUT</sub>	X direction output voltage.
16	N/C	No internal connection. Leave unconnected.

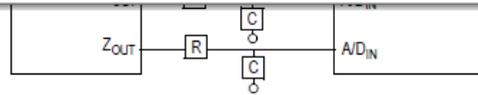
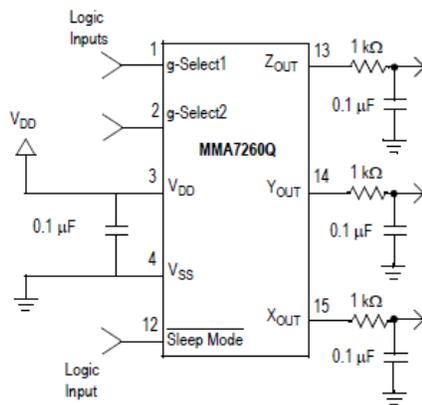


Figure 6. Recommended PCB Layout for Interfacing Accelerometer to Microcontroller

NOTES:

1. Use 0.1 µF capacitor on V<sub>DD</sub> to decouple the power source.
2. Physical coupling distance of the accelerometer to the microcontroller should be minimal.
3. Flag underneath package is connected to ground.
4. Place a ground plane beneath the accelerometer to reduce noise, the ground plane should be attached to all of the open ended terminals shown in Figure 6.
5. Use an RC filter with 1.0 kΩ and 0.1 µF on the outputs of the accelerometer to minimize clock noise (from the switched capacitor filter circuit).
6. PCB layout of power and ground should not couple power supply noise.
7. Accelerometer and microcontroller should not be a high current path.
8. A/D sampling rate and any external power supply switching frequency should be selected such that they do not interfere with the internal accelerometer sampling frequency (11 kHz for the sampling frequency). This will prevent aliasing errors.

## B.2 Camera datasheet

Source: Firefly® MV

Website: <http://www.lustervision.com/UploadFile/20141214/fireflymv-FW-datasheet.pdf>

# Firefly® MV

SMALL + INEXPENSIVE + USB OR 1394

- 14 different combinations of form factor and interface
- 1394a or USB 2.0 digital interface
- Ultra compact at 44 x 34 x 24.4 mm
- Under \$200\* USD

The Firefly MV line offers a total of 14 different combinations of form factor and interface that are designed to address a wide variety of applications in industrial and non-industrial imaging, such as object and gesture tracking, optical character recognition (OCR), augmented reality, and multitouch interface technology.

\* 0.3MP model only, when purchasing 5 or more cameras.



Specification	FFMV-03M2M/C (BW or Color)	FMVU-03MTM/C (BW or Color)	FMVU-13S2C (Color)
Image Sensor Type	1/3" progressive scan CMOS		
Shutter Type	Global shutter using Micron TrueSNAP™ technology		Rolling shutter
Image Sensor Model	Micron MT9V022		Sony IMX035LQR-C
Maximum Resolution	752 (H) x 480 (V)		1328 (H) x 1048 (V)
Pixel Size	6.0 μm x 6.0 μm		3.63 μm x 3.63 μm
Analog-to-Digital Converter	On-chip 10-bit ADC		On-chip 10 / 12-bit ADC
Video Data Output	8 and 16-bit digital data		
Image Data Formats	Y8, Y16 (monochrome), 8-bit and 16-bit raw Bayer data (color models)		
Digital Interface	6-pin IEEE 1394a for camera control, video data, power	5-pin Mini-B USB 2.0 for camera control, video data, power	
Transfer Rates	400 Mb/s	480 Mb/s	
Maximum Frame Rate	752x480 at 60 FPS • 320x240 at 112 FPS (region of interest) 320x240 at 122 FPS (2 x 2 pixel binning)		Free-running acquisition: 1328x1048 at 23FPS, 664x524 at 60 FPS (2x2 pixel binning) 640x480 at 60 FPS (center cut-out mode) Triggered acquisition: Frame rate is half the specified rate. Maximum frame rate in external trigger mode: 30FPS
Partial Image Modes	pixel binning and region of interest modes via Format_7		pixel binning or center cut-out (640x480) mode via Format_7
General Purpose I/O Ports	7-pin JST GPIO connector, 4 pins for trigger and strobe, 1 pin +3.3 V, 1 Vext pin for external power		
Gain Control	automatic / manual, 0 dB to 12 dB		automatic / manual, 0 dB to 18 dB
Shutter Speed	automatic / manual, 0.12 ms to 512 ms		automatic / manual, 0.03 ms to 8000 ms
Gamma	0 to 1 (enables 12-bit to 10-bit companding)		0.5 to 4.00
Synchronization	via external trigger, software trigger, or free-running <sup>1</sup>	via external trigger	
External Trigger Modes	I2C v1.31 Trigger Modes 0 and 3		
Power Requirements	8 to 30 V via IEEE-1394, less than one (1) Watt	4.75 to 5.25 V via the Mini-B USB 2.0 or GPIO connector, less than one (1) Watt	
Dimensions (L x W x H)	24.4 x 44 x 34 mm		
Mass	37 g (including tripod adapter)		
Camera Specifications	I2C 1394-based Digital Camera Specification v1.31		
Memory Storage	three memory channels for user configurable power-up settings		
Lens Mount	CS-mount (5mm C-mount adapter included) • M12 microlense mount <sup>2</sup>		
Compliance	CE, FCC Class B, RoHS		
Operating Temperature	0° to 45°C		
Storage Temperature	-30° to 60°C		
Warranty	1 year		
Configuration Options	board-level <sup>2</sup> • microlens <sup>2</sup> • mini 1394 connector <sup>2</sup> • metal case		

# Firefly<sup>®</sup> MV Specifications

## Plug and Play

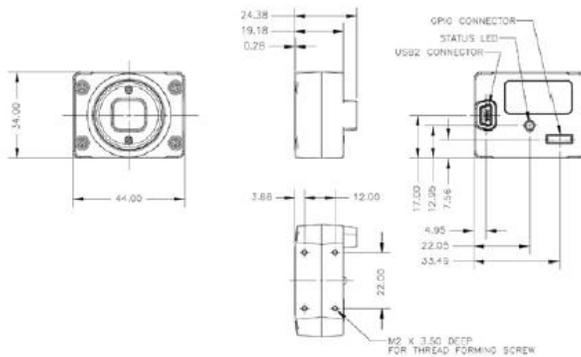
The Universal Serial Bus (USB) 2.0 and IEEE-1394a standards provide high bandwidth interfaces that are well-suited for digital imaging, and are some of the most well-established and popular serial interfaces in use today. Both interfaces are widely available on a variety of hardware platforms, including Macintosh and PC systems, and a variety of laptop, desktop, and small form factor (embedded) systems. PC interfaces such as built-in, PCI, PCI Express, CardBus or ExpressCard are supported.

## Digital Interfaces for Power and Control

Both USB 2.0 and 1394a provide power and data over one cable which minimizes the need for additional cables or external power sources. The 1394a FireflyMV is ideal for multi-camera applications, while the new USB 2.0 model is ideal for single camera applications that can be supported on multiple hardware platforms. The camera also provides a 7-pin general purpose I/O connector. The GPIO connector can be used to synchronize the camera to external devices, such as an external trigger source, power external circuitry, or power the camera via an external device.

## Dimensional Drawings

Measurements in mm  
CAD drawings for board level are available online at [www.ptgrey.com/support](http://www.ptgrey.com/support)



## Development Kit

This kit has all the hardware and software required for rapid design and prototyping, including a license of the FlyCapture<sup>®</sup> software development kit (SDK), data cable, and GPIO wiring harness. The FlyCapture software development kit (SDK) is included with all Point Grey Imaging Products. The SDK is designed to allow image acquisition and camera control, and comes with a variety of C/C++ source code examples.

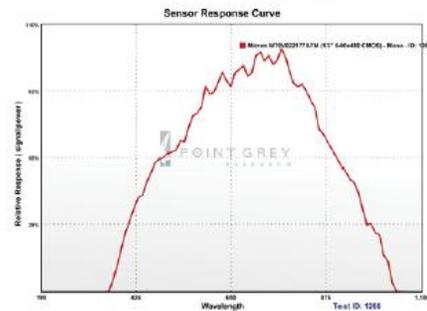
## Updatable Firmware

The field-programmable gate array (FPGA) chip controls all camera functionality, including exposure, resolution and frame rate, pixel binning, user memory channels and more. It can also be updated with new functionality in the field.

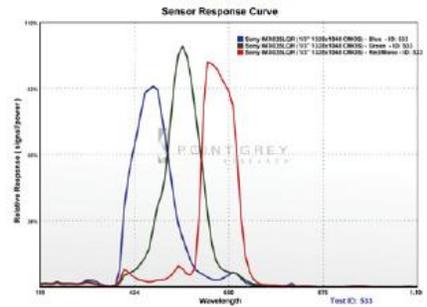
## Near-IR Performance

The monochrome Firefly MV is highly sensitive in the near-IR with QE greater than 35%.

FFMV-03M2 / FMVU-03T2



FMVU-13S2C



## B.3 Range sensor datasheet

### B.3.1 Long range measurement

Source: Sharp GP2Y0A710K0F IR sensor datasheet

Website: [https://www.parallax.com/sites/default/files/downloads/28998-Sharp-Datasheet-gp2y0a710k\\_e.pdf](https://www.parallax.com/sites/default/files/downloads/28998-Sharp-Datasheet-gp2y0a710k_e.pdf)

**SHARP**

GP2Y0A710K0F

# GP2Y0A710K0F

Distance Measuring Sensor Unit  
Measuring distance: 100 to 550 cm  
Analog output type



#### ■ Description

GP2Y0A710K0F is a distance measuring sensor unit, composed of an integrated combination of PSD (position sensitive detector), IRED (infrared emitting diode) and signal processing circuit.

The variety of the reflectivity of the object, the environmental temperature and the operating duration are not influenced easily to the distance detection because of adopting the triangulation method.

This device outputs the voltage corresponding to the detection distance. So this sensor can also be used as a proximity sensor.

#### ■ Features

1. Long distance type  
Distance measuring range : 100 to 550 cm
2. Analog output type
3. Package size : 58×17.6×22.5 mm
4. Consumption current : Typ. 30 mA
5. Supply voltage : 4.5 to 5.5 V

#### ■ Agency approvals/Compliance

1. Compliant with RoHS directive(2002/95/EC)

#### ■ Applications

1. Projector (for auto focus)
2. Robot cleaner
3. Auto-switch for illumination, etc.
4. Human body detector
5. Amusement equipment  
(Robot, Arcade game machine)

■ Absolute Maximum Ratings (T<sub>a</sub>=25°C, V<sub>CC</sub>=5V)

Parameter	Symbol	Rating	Unit
Supply voltage	V <sub>CC</sub>	-0.3 to +7	V
Output terminal voltage	V <sub>O</sub>	-0.3 to V <sub>CC</sub> +0.3	V
Operating temperature	T <sub>opr</sub>	-10 to +60	°C
Storage temperature	T <sub>stg</sub>	-40 to +70	°C

■ Electro-optical Characteristics (T<sub>a</sub>=25°C, V<sub>CC</sub>=5V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Average supply current	I <sub>CC</sub>	L=150cm (Note 1)	—	30	50	mA
Distance measuring	ΔL	(Note 1)	100	—	550	cm
Output voltage	V <sub>O</sub>	L=100cm (Note 1)	2.3	2.5	2.7	V
Output voltage differential	ΔV <sub>O1</sub>	Output voltage difference between L=100cm and L=200cm (Note 1)	0.5	0.7	0.9	V
	ΔV <sub>O2</sub>	Output voltage difference (L=100cm→200cm)/ Output voltage difference (L=200cm→550cm)(Note 1,2)	1.25	1.55	1.85	V

\* L : Distance to reflective object

Note 1 : Using reflective object : White paper (Made by Kodak Co., Ltd. gray cards R-27·white face, reflectance; 90%)

Note 2 : The value at 550 cm is the average of 20 times distance measuring.

■ Recommended operating conditions

Parameter	Symbol	Conditions	Rating	Unit
Supply voltage	V <sub>CC</sub>		4.5 to 5.5	V

Fig. 1 Timing chart

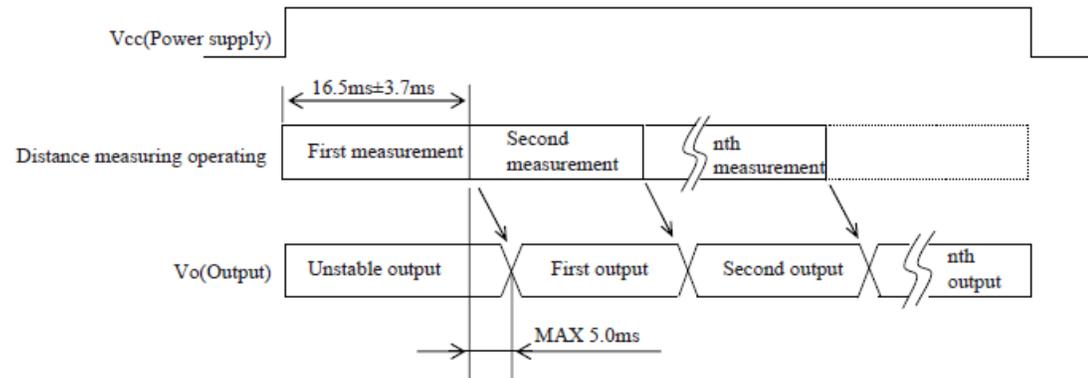
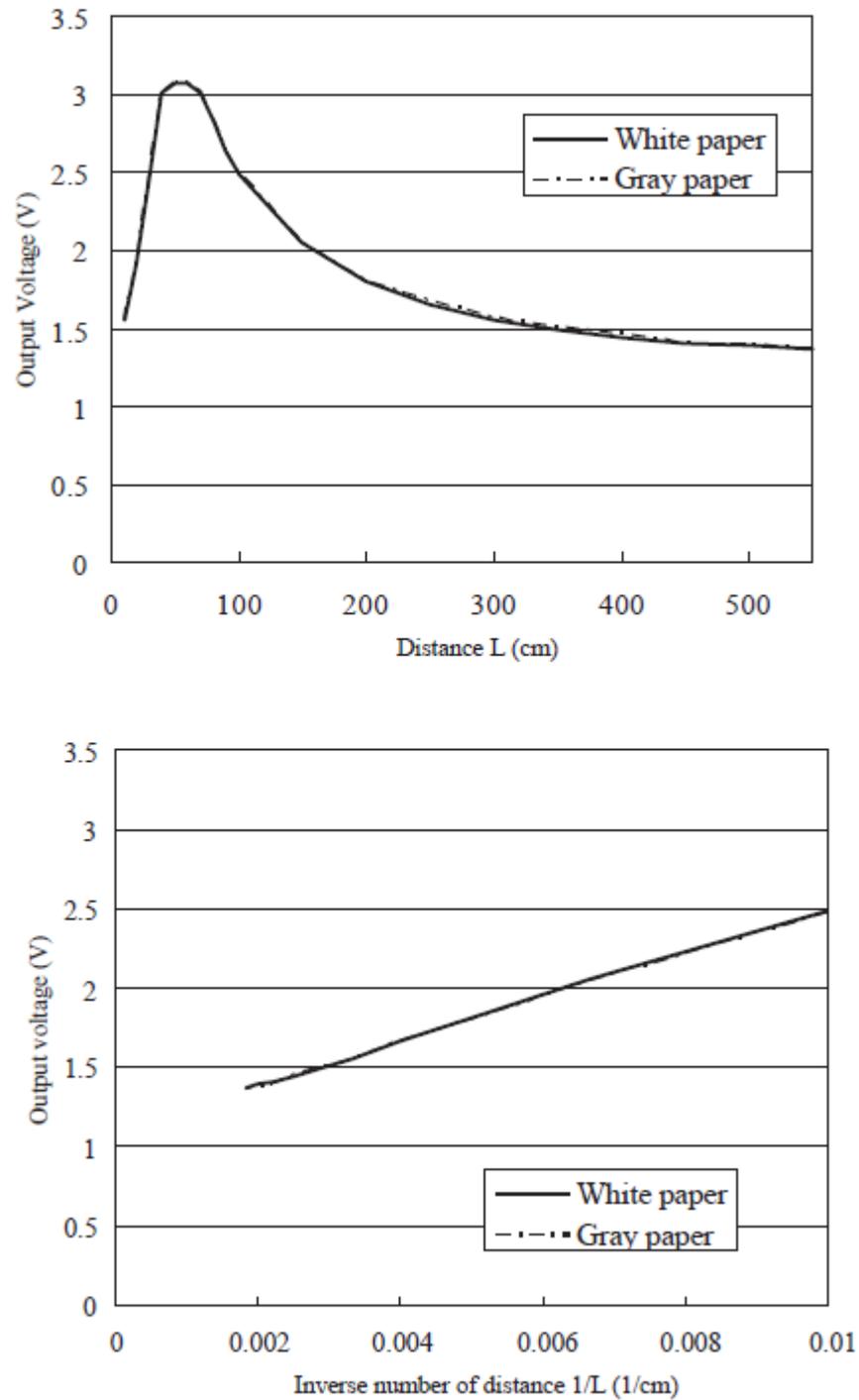


Fig. 2 Example of distance measuring characteristics(output)



Note. Reflection : White paper (reflectance : 90%)  
: Gray paper (reflectance: 18%)

## B.3.2 Short range measurement

Source: GP2Y0A21YK Datasheet - Sharp Electronics | DatasheetLib.com

Website: [http://www.datasheetlib.com/datasheet/835456/gp2y0d21yk\\_sharp-electronics.html#datasheet](http://www.datasheetlib.com/datasheet/835456/gp2y0d21yk_sharp-electronics.html#datasheet)

**SHARP**

GP2Y0A21YK/GP2Y0D21YK

# GP2Y0A21YK/ GP2Y0D21YK

## I Features

1. Less influence on the color of reflective objects, reflectivity
2. Line-up of distance output/distance judgement type  
Distance output type (analog voltage) : GP2Y0A21YK  
Detecting distance : 10 to 80cm  
Distance judgement type : GP2Y0D21YK  
Judgement distance : 24cm  
(Adjustable within the range of 10 to 80cm [Optionally available])
3. External control circuit is unnecessary
4. Low cost

## I Applications

1. TVs
2. Personal computers
3. Cars
4. Copiers

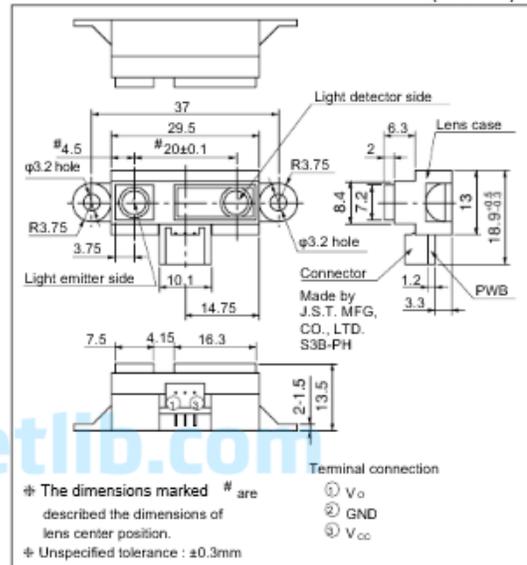
## I Absolute Maximum Ratings (T<sub>a</sub>=25°C, V<sub>CC</sub>=5V)

Parameter	Symbol	Rating	Unit
Supply voltage	V <sub>CC</sub>	-0.3 to +7	V
Output terminal voltage	V <sub>O</sub>	-0.3 to V <sub>CC</sub> +0.3	V
Operating temperature	T <sub>OPR</sub>	-10 to +80	°C
Storage temperature	T <sub>STG</sub>	-40 to +70	°C

## General Purpose Type Distance Measuring Sensors

### I Outline Dimensions

(Unit : mm)



■ Recommended Operating Conditions

Parameter	Symbol	Rating	Unit
Operating supply voltage	V <sub>CC</sub>	4.5 to +5.5	V

■ Electro-optical Characteristics

(T<sub>a</sub>=25°C, V<sub>CC</sub>=5V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Distance measuring range	ΔL	<sup>*1</sup> <sup>*3</sup>	10	-	80	cm
Output terminal voltage	GP2Y0A21YK V <sub>O</sub>	L=80cm <sup>*1</sup>	0.25	0.4	0.55	V
	V <sub>OH</sub>	Output voltage at High <sup>*1</sup>	V <sub>CC</sub> -0.3	-	-	V
Output terminal voltage	GP2Y0D21YK V <sub>OL</sub>	Output voltage at Low <sup>*1</sup>	-	-	0.6	V
	V <sub>OL</sub>	Output voltage at Low <sup>*1</sup>	-	-	0.6	V
Difference of output voltage	GP2Y0A21YK ΔV <sub>O</sub>	Output change at L=80cm to 10cm <sup>*1</sup>	1.65	1.9	2.15	V
Distance characteristics of output	GP2Y0D21YK V <sub>O</sub>	<sup>*1</sup> <sup>*4</sup> <sup>*2</sup>	21	24	27	cm
	V <sub>O</sub>	<sup>*1</sup> <sup>*4</sup> <sup>*2</sup>	21	24	27	cm
Average Dissipation current	I <sub>CC</sub>	L=80cm <sup>*1</sup>	-	30	40	mA

Note) L : Distance to reflective object

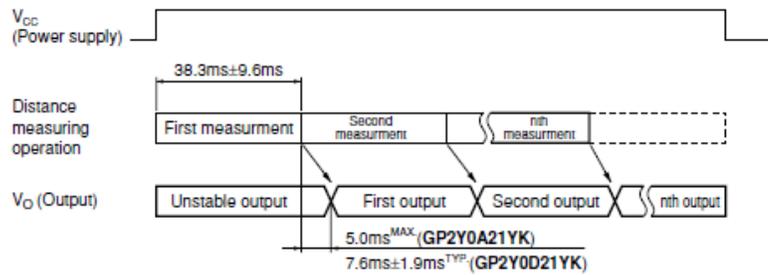
\*1 Using reflective object : White paper (Made by Kodak Co. Ltd. gray cards R-27 · white face, reflective ratio ; 90%)

\*2 We ship the device after the following adjustment : Output switching distance L=24cm±3cm must be measured by the sensor

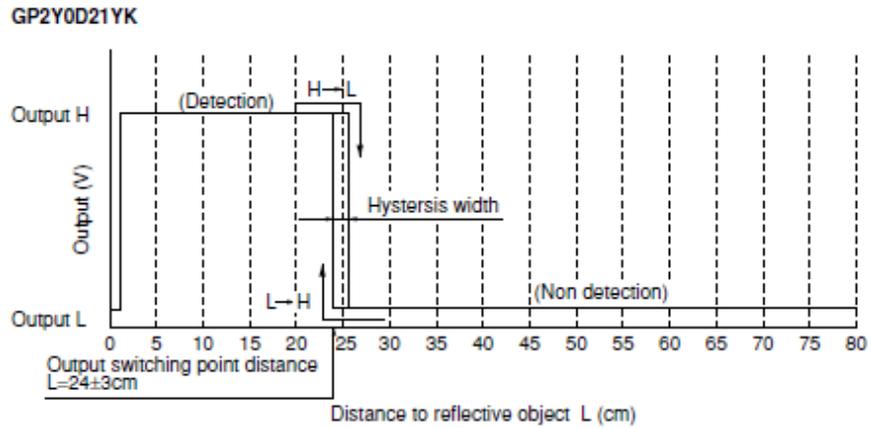
\*3 Distance measuring range of the optical sensor system

\*4 Output switching has a hysteresis width. The distance specified by V<sub>O</sub> should be the one with which the output L switches to the output H

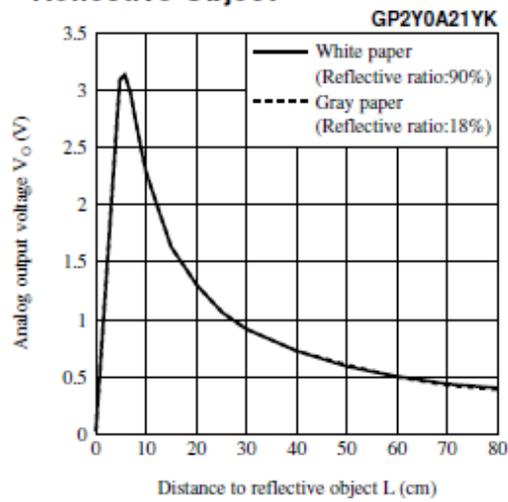
Fig.3 Timing Chart



**Fig.4 Distance Characteristics**



**Fig.5 Analog Output Voltage vs. Distance to Reflective Object**



## B.4 Encoder datasheet

Source: 47:1 Metal Gearmotor 25Dx52L mm with 48 CPR Encoder - SGBotic

Website: [https://www.sgbotic.com/index.php?dispatch=products.view&product\\_id=1408](https://www.sgbotic.com/index.php?dispatch=products.view&product_id=1408)

### 47:1 Metal Gearmotor 25Dx52L mm with 48 CPR Encoder



Pololu

This cylindrical, 2.54" x 0.98" x 0.98" brushed DC gearmotor with a **46.85:1** metal gearbox is a lower-current alternative to the [HP version](#). It has an integrated 48 CPR quadrature encoder on the motor shaft, which provides **2249 counts per revolution** of the gearbox's output shaft. These units have a 0.315"-long, 4 mm-diameter D-shaped output shaft. This gearmotor is also available [without an encoder](#).

Key specs at 6 V: 120 RPM and 80 mA free-run, 50 oz-in (3.6 kg-cm) and 2.2 A stall.

Select options:

[Description](#) [Specs \(9\)](#) [Pictures \(6\)](#) [Resources \(1\)](#) [FAQs \(0\)](#) [On the blog \(0\)](#)

#### Details for item #2285

Exact gear ratio:  $\frac{22 \times 22 \times 22 \times 22 \times 24}{12 \times 10 \times 10 \times 10 \times 10} \approx 46.85:1$

This cylindrical, 2.54" x 0.98" x 0.98" brushed DC gearmotor with a 46.851:1 metal gearbox has an integrated 48 CPR quadrature encoder on the motor shaft, which provides 2249 counts per revolution of the gearbox's output shaft. These units have a 0.315"-long, 4 mm-diameter D-shaped output shaft.

These motors are intended for use at 6 V. In general, these kinds of motors can run at voltages above and below this nominal voltage, so they should comfortably operate in the 3 - 9 V range, though they can begin rotating at voltages as low as 1 V. Higher voltages could start negatively affecting the life of the motor.

These gearmotors have output shafts with a diameter of 4 mm. This [universal aluminum mounting hub for 4mm shafts](#) can be used to mount wheels and mechanisms to the gearmotor's output shaft.

The face plate has two mounting holes threaded for M3 screws. You can use the [25D mm metal gearmotor bracket](#) to mount the gearmotor to your project via these mounting holes and the screws that come with the bracket.

The integrated two-channel Hall effect encoder is used to sense the rotation of a magnetic disk on a rear protrusion of the motor shaft. The quadrature encoder provides a resolution of 48 counts per revolution of the motor shaft when counting both edges of both channels. To compute the counts per revolution of the gearbox output, multiply the gear ratio by 48.

**Specifications:**

- Gear ratio: 47:1
- Free-run speed @ 6V: 120 rpm
- Free-run current @ 6V: 80 mA
- Stall current @ 6V: 2200 mA
- Stall torque @ 6V: 50 oz-in (3.6kgf.cm)
- Lead length: ~8"

**Dimensions:**

- Size: 25D x 64L mm
- Weight: 3.58 oz
- Shaft diameter: 4 mm

## B.5 Current sensor datasheet

Source: Acs712 Datasheet PDF | Free Acs712 Download | alldatasheet.com

Website: <http://www.alldatasheet.com/view.jsp?Searchword=Acs712>

### ACS712

*Fully Integrated, Hall Effect-Based Linear Current Sensor with 2.1 kVRMS Voltage Isolation and a Low-Resistance Current Conductor*

#### Description (continued)

loss. The thickness of the copper conductor allows survival of the device at up to 5× overcurrent conditions. The terminals of the conductive path are electrically isolated from the sensor leads (pins 5 through 8). This allows the ACS712 current sensor to be used in applications requiring electrical isolation without the use of opto-isolators or other costly isolation techniques.

The ACS712 is provided in a small, surface mount SOIC8 package. The leadframe is plated with 100% matte tin, which is compatible with standard lead (Pb) free printed circuit board assembly processes. Internally, the device is Pb-free, except for flip-chip high-temperature Pb-based solder balls, currently exempt from RoHS. The device is fully calibrated prior to shipment from the factory.

#### Selection Guide

Part Number	Packing*	T <sub>A</sub> (°C)	Optimized Range, I <sub>p</sub> (A)	Sensitivity, Sens (Typ) (mV/A)
ACS712ELCTR-05B-T	Tape and reel, 3000 pieces/reel	-40 to 85	±5	185
ACS712ELCTR-20A-T	Tape and reel, 3000 pieces/reel	-40 to 85	±20	100
ACS712ELCTR-30A-T	Tape and reel, 3000 pieces/reel	-40 to 85	±30	66

\*Contact Allegro for additional packing options.

#### Absolute Maximum Ratings

Characteristic	Symbol	Notes	Rating	Units
Supply Voltage	V <sub>CC</sub>		8	V
Reverse Supply Voltage	V <sub>RCC</sub>		-0.1	V
Output Voltage	V <sub>IOUT</sub>		8	V
Reverse Output Voltage	V <sub>RIOUT</sub>		-0.1	V
Reinforced Isolation Voltage	V <sub>ISO</sub>	Pins 1-4 and 5-8; 60 Hz, 1 minute, T <sub>A</sub> =25°C	2100	V
		Voltage applied to leadframe (I <sub>p</sub> + pins), based on IEC 60950	184	V <sub>peak</sub>
Basic Isolation Voltage	V <sub>ISO(bsc)</sub>	Pins 1-4 and 5-8; 60 Hz, 1 minute, T <sub>A</sub> =25°C	1500	V
		Voltage applied to leadframe (I <sub>p</sub> + pins), based on IEC 60950	354	V <sub>peak</sub>
Output Current Source	I <sub>IOUT(source)</sub>		3	mA
Output Current Sink	I <sub>IOUT(sink)</sub>		10	mA
Overcurrent Transient Tolerance	I <sub>p</sub>	1 pulse, 100 ms	100	A
Nominal Operating Ambient Temperature	T <sub>A</sub>	Range E	-40 to 85	°C
Maximum Junction Temperature	T <sub>J(max)</sub>		165	°C

## Appendix C Smoothing filters

Filtering is necessary as long as it does not attenuate frequencies in the signal band. Important factors of filter design are the frequency response of the filter, filter order, and delay. Noisy sensor outputs are not fit for direct measurement, thus some form of sensor processing is needed to separate the actual sensor signal from the noise.

Summarized in Table C.1 there are Lowess, Savizhy Golay and moving average filters relevance to this work. Table C.2 summarizes the advantages and disadvantages of each filter. Azami et al. [108] reported that both moving average and Savitzky-Golay filters can smooth noisy data and at the same time preserve the shape of the curve. But Savitzky-Golay filter produces very little distortion in the signal in comparison to moving average filter. This approach tends to preserve features of the distribution such as relative maxima, minima and width, which are usually 'flattened' by moving average or other adjacent averaging techniques.

**Table C.1** Types of filters

<b>Filter type</b>	<b>Description</b>	<b>Ref.</b>
Moving average filters.	Capture important trends or patterns. It averages the neighboring data points, eliminating high-frequency noise, thus it is a low-pass filter. Each smoothed value is determined by neighboring data points defined within the span, giving equal weight to each data point.	108
Local regression (Loess, lowess, robust loess and robust lowess)	Weighted linear least-squares regression is performed using nearest neighbors of values defined by the span. The data point to be smoothed has the largest weight and the most influence on the fit, while data points outside the span have zero weight and no influence on the fit. Thus it does not have the boundary problems present in moving average filters.	109
Least squares smoothing filter (Also called Savitzky- Golay smoothing filter or digital smoothing polynomial filter)	Minimize the estimation error. Used to "smooth out" a noisy signal whose frequency of span (without noise) is large.	108

**Table C.2** Advantages and Disadvantages of filter types

<b>Filter type</b>	<b>Advantages</b>	<b>Disadvantages</b>	<b>Ref.</b>
Moving average filters	<ul style="list-style-type: none"> <li>▪ Easiest to understand and use.</li> <li>▪ Optimal for a common task: reducing random noise while retaining a sharp step response.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Worst filter for frequency domain encoded signals, with little ability to separate one band of frequencies from another.</li> <li>▪ Outlier gives rise to huge errors in the whole smoothed series.</li> <li>▪ Contain boundary problem.</li> </ul>	108
Local regression (Loess, lowess, robust loess and robust lowess)	<ul style="list-style-type: none"> <li>▪ Does have boundary problems.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Long processing time.</li> </ul>	109
Least squares smoothing filter (Also called Savitzky-Golay smoothing filter or digital smoothing polynomial filter)	<ul style="list-style-type: none"> <li>▪ Exploits matrix sparsity, allowing consideration of larger problems without increasing of computation effort.</li> <li>▪ Convenient and flexible for estimation and prediction problems.</li> <li>▪ Capable of converging with fewer readings and achieving greater accuracy.</li> <li>▪ Simple to implement.</li> <li>▪ Stable and robust performance against different signal conditions</li> </ul>	<ul style="list-style-type: none"> <li>▪ Cannot deal with the case where the noise subspace of multi-dimension problems.</li> <li>▪ Time consuming.</li> <li>▪ Slow convergence (due to eignvalue spread)</li> </ul>	108

## Appendix D Matlab Codes

The main program is the parallel computing program, which calls the other functions for acquisition of the displacement information from the sensors (i.e. accelerometer, camera, encoder and range sensor), and then convert the data into vibrational displacement of the flexible link manipulator.

### D.1 Parallel computing

This is the main program for starting parallel computing. The program started with opening the txt files for saving model-based prediction, EKF 3, EKF 4, velocity, input voltage and time data. It then initialize the valuables for getting current clock time, image count, time state, encoder start time, set platform velocity to zero and initialize the acceleration data matrix.

It then create the camera object for setting the starting the camera recording. Next, it create the DAQ session for NI device. This set to connect the individual input channels for acquiring the signals from the sensors.

The program then start the endless loop, where the program will run continuously until the Enter key is entered to break out of the loop. In the loop, the sub-program for encoder, camera and accelerometer in parallel computing.

Lastly, the program ends with closing all the txt files and terminating the sessions.

Source code:

---

```
% Main program for starting parallel computing

% Opening the txt files for saving of data for plotting
mbp = fopen('mbp.txt','a'); % open txt file for saving mbp data
ekf3 = fopen('ekf3.txt','a'); % open txt file for saving ekf3 data
ekf4 = fopen('ekf4.txt','a'); % open txt file for saving ekf4 data
vps = fopen('velps.txt','a'); % open txt file for saving velps data
Vi = fopen('Vin.txt','a'); % open txt file for saving Vin data
Tx = fopen('tx.txt','a'); % open txt file for saving time data

% Initializing the parameters
c0 = clock; % Record the initial time when the program started
```

```

img1 = 0; % Set initial img1 to 0.
k = 0; % Set initial time step
te = c0; % Set initial time for use by encoder
velp = 0; % Set initial velocity of the platform
axall = [0 0 0]; % Initialize an acceleration data matrix

% Setup camera object
cam = webcam('1.3M WebCam'); % Set camera object as cam

% Create a session, and add three analog input channels with the
% accelerometer measurement type.
% Input channel ai0 connected to X-axis
% Input channel ai1 connected to Y-axis
% Input channel ai2 connected to Z-axis
s = daq.createSession('ni');
addAnalogInputChannel(s, 'cDAQ1Mod3', ai0, 'Accelerometer');
addAnalogInputChannel(s, 'cDAQ1Mod3', ai1, 'Accelerometer');
addAnalogInputChannel(s, 'cDAQ1Mod3', ai2, 'Accelerometer');

% Capacitive accelerometer is 200 mV per Gravity for 6g option
s.Channels(1).Sensitivity = 0.200; % Set X-axis sensitivity
s.Channels(2).Sensitivity = 0.200; % Set Y-axis sensitivity
s.Channels(3).Sensitivity = 0.200; % Set Z-axis sensitivity

while 1 % Endless looping
    key = get(gcf, 'CurrentKey');
    if(strcmp (key , 'return'))
        disp('stopped');
        break; % To break out of while loop when Enter key is pressed
    end

parpool(2) % Starting parallel pool (parpool)

parfor i = 1:3 % Start parallel loop to get signal from camera,
    % accelerometer and encoder simultaneously
    if i == 1
        [disp, velp, te] = encoder(dist,velp,te) % Call encoder function
    else if i == 2
        [img,imgt] = camera(c0) % Call camera function
    else
        % Call accelerometer function
        k = k+1; % Next time step
        [accx,vel,posz,pos_diff,tx] = accelerometer(c0,k,velp, axall)
    end

    main(k,img,img1,imgt,accx,vel,posz,pos_diff,tx,disp,velp,te)

end % End of parfor loop

img1 = img; % Save the latest camera computed displacement

end % End of while loop

% close the txt files
fclose(mbp); % close the mbp.txt file

```

```

fclose(ekf3); % close the ekf3.txt file
fclose(ekf4); % close the ekf4.txt file
fclose(vps); % close the velps.txt file
fclose(Vi); % close the Vin.txt file

if user_input == 1 % If user enter 1
delete(gcp('nocreate')) % Terminate the existing sessions
end

```

---

## D.2 Acquire displacement data from camera

This function calls for the ‘capture\_img’ function to acquire a single image, the existence of the image and the timestamp of the image.

The function first check for the existence of the image. When the image existence, it calls for the image processing function to acquire the position of target point. It then find the time step by finding the difference between the initial time when the program starts, convert into time in seconds.

The function then returns the variables for target position and step time of the image.

Source code:

---

```

% This function call the capture image function to trigger the camera
% to capture image, and then call the process image function to convert
% the capture image into displacement.
function [img,imgt] = camera(c0)

img(imga,c,x_img) = capture_img();

if x_img == 1
    img = process_img(imga); % Call for processing the image
    c1 = c-c0; % Find the time difference from start of program
    imgt = (c1(5)*60 + c1(6))*1000; % Convert time into millisecond
end

```

---

## D.3 Acquire image from camera

This function triggers camera to snapshot a single image, and get the time stamp. It then check to see if an image has been captured.

The function then returns the variables for the image, the image's timestamp and existence of the image.

Source code:

---

```
% Function for capturing the image from the camera
function [imga,c,x_img] = capture_img()

imga = snapshot(cam); % Acquire a single image from the camera
c = clock; % Get the timestamp of the image acquired
x_img = exist ('imga'); % Check for existence of image
```

---

## D.4 Processing the image

This is the function for processing the image. It first identify the target object in the image. It then removes the current image from memory to make space for the next image to be captured.

To perform the processing of the image, it first estimates the background pixels and remove any objects that are less than 50 pixels. It performs a subtract of the background image to get a uniform background. It then creates a binary version of the image.

Next, it determines the number of objects found in the image. It then find the centroid of the object 50, and determine its x- and y- coordinates.

Finally, it computes and returns the tip position of the flexible manipulator in mm.

Source code:

---

```
% Function for processing the image and convert the marker position in
% image into X- and Y- pixel coordinates
function [img] = process_img (imga)

% Identify target object from the image
d = imga(:,:,1) > 100 & imga(:,:,2) > 100;

clear('imga'); % Clear the current image from memory
x_img = exist ('imga'); % Check for existence of image

% Estimate the value of Background Pixels
% Remove object having a radius less than 50 pixels by opening it with
% the created a disk-shaped structuring element with a radius of 50
% pixels
```

```

background = imopen(d,strel('disk',50));

% To create a more uniform background, subtract the background image,
% background, from the original image
I2 = imsubtract(d,background);

% Create a Binary Version of the Image
level = graythresh(I2);
I3 = im2bw(I2,level);

% Determine the Number of Objects in the Image
[labeled,numObjects] = bwlabel(I3,8);
graindata = regionprops(labeled,'basic');

    for jj=1:numObjects

        if graindata(jj).Area > 50
            cc = graindata(jj).Centroid;
            end
        end

% X- coordinate co-respond to the horizontal displacement and Y-
% coordinate co-respond to vertical displacement
x_coor = cc(1); % X- coordinate of the object in the image
y_coor = cc(2); % Y- coordinate of the object in the image

% Compute the tip displacement of the flexible manipulator based on X-
% coordinate. Note that 407 is the rest position of the manipulator.
% 26.02 is resolution used to convert into cm displacement (see Fig.
% 3.9 in section 3.2.3.2)
img = (407 - x_coor)/26.02;

```

---

## D.5 Acquire displacement data from accelerometer

This function calls for the 'capture\_acceleration' function. The latest acquired acceleration data is added to the end of the previous data. When the total past acceleration data reaches 1000, it removes the first row of data.

It then performs a lowpass filtering on the latest and past acceleration data acquired.

It then calls for the sortout function to remove the walking bias error. Lastly, it calls the displacement function to convert the acceleration data into velocity and position data.

The function returns the acceleration, velocity position, position, change in position and time variables.

Source code:

---

```

% This function call the acquire acceleration function to trigger the
% accelerometer to acquire acceleration signal, and then call the
% convert to displacement function to convert the signal to
% displacement and velocity
function [accx,vel,posz,pos_diff,tx] = accelerometer(c0,k,velp)

% Call function to acquire acceleration signal from accelerometer
[ax, tx] = capture_acceleration(c0);

axall = [axall; ax]; % Add latest acceleration signal into past data

if length(axall) > 1000
    axall(1,:) = []; % If more than 1000 row of data, remove first row
end

[acx] = lowpass(axall); % Filter out noisy acceleration signal
[accx] = sortout(acx); % Remove walking bias error

% Convert acceleration signal into displacement with mobile velocity
% input
[vel,posz,pos_diff] = displacement(velp,accx,tx,k)

```

---

## D.6 Acquire acceleration data

This function will acquire the x-, y- and z- gravitation acceleration data from the accelerometer with its timestamp, convert it into acceleration data. It then finds the latest time step by subtracting it with the initial start time of the program.

It finally returns the acceleration and time step.

Source code:

---

```

% Function to acquire accelerometer's data and output as acceleration
% and its timestamp
function [accx, tx] = capture_acceleration(c0)

% Acquire a single scan of data its trigger time indicating the
% absolute time the operation was triggered
[data,triggerTime] = inputSingleScan(s);

% Convert gravity data from accelerometer into acceleration
accx = data(1)*9.81; % X-axis acceleration data
accy = data(2)*9.81; % Y-axis acceleration data
accz = data(3)*9.81; % Z-axis acceleration data

% Record the time since start of program
t1 = triggerTime(1)- c0; % X-axis recorded timestamp
tx = t1(5)*60 + t1(6); % convert into second

```

---

## D.7 Convert acceleration into velocity and displacement

The function calls the 'displacement' function to convert the latest batch of 1000 acceleration data into velocity, position and difference in position. It performs the conversion from acceleration into velocity, and converts into position. It then compute the difference in the position data with current and previous points.

Finally, the function returns the velocity, position and difference in previous positions

Source code:

---

```
% This function accepts acceleration input and then convert into
% displacment ouput
function [vel,posz,pos_diff] = displacement(velp,accx,tx,k)

Ts = tx/k; % Time constant compute from average time between signal

% Convert acceleration to velocity using trapezoidal integration method
vel = Ts*cumtrapz(accx);
vel = vel - v; % offset with velocity of platform

% Convert velocity to displacement using trapezoidal integration method
posz=Ta*cumtrapz(vel);
d = mean(posz); % Find midpoint of vibration
posz = posz - d; % Offset to midpoint
pos_diff=diff(posz); % Find differences in current and previous points
```

---

## D.8 Filtering noisy acceleration signals

This function performs a lowpass filtering of the latest batch of acceleration data, and returns the filtered acceleration data.

Source code:

---

```
% This function uses lowpass filter to filter noisy acceleration signal
function [accx] = lowpass(x1)

% Lowpass filter
Fs = 2500; fn=fs/2; fc=5;
[b,a]=butter(2,fc/fn,'low');
accx = filter(b,a,x1);
```

---

## D.9 Remove walking bias errors for acceleration signal

This function removes the walking bias errors using standard deviation of the data. If the standard deviation is lower than 0.045 or more than -0.045 will set convert the data into mean value. If the mean value is more than 1 or less than -1, the data will be convert to zero.

Source code:

---

```
% This function uses the standard deviation to remove walking bias
% errors that occur during constant or zero acceleration
function [a] = sortout(x)

[n1 p1] = size(x);
k = 500;
ff = 0;
for j = 1:25:n1

    if k > n1
        k = n1;
    end

    sd = std(x(j:k));

    if (sd < 0.045) && (sd > -0.045) %0.03
        if ff == 0
            ff = j+25;
            kk = k-25;
        end

        mn = mean(x(j:k));

        if (mn < 1) && (mn > -1)
            x(ff:kk) = 0;
        else
            x(j:k) = mn;
        end
    else
        ff = 0;
    end

    k = k+25;
end

a = x;
```

---

## D.10 Encoder

This function acquires the encoders counter value and calculates the encoder linear position data. It first creates the acquisition session with the NI DAQ, and then acquires a single value of the encoder.

If there is no change in the displacement value, and if there is no displacement change for more than 1 second the latest velocity will be set to zero value if there. Otherwise it will set to previous velocity value.

If there is change in the displacement, it calculates the new displacement and velocity value.

Source code:

---

```
% This function acquires the encoders counter value and calculate the
% encoder linear position data
function [disp,velp,te] = encoder(dist,velp,te)

dd = 0.2042; % For linear displacement, one rev is 2*pi*R = 0.2042m
encoderCPR = 2249; % The optical encoder has 2249 pulses per shaft
                % revolution

s = daq.createSession('ni');
ch1 = addCounterInputChannel(s, 'cDAQ1Mod1', 'ctr0', 'Position')
ch1.EncoderType = 'X2'; % X2 decoding is used, which counts the rising
                        % and falling edges of channel A's signal
encoderCount = inputSingleScan(s) % Acquire encoder count
displace = encoderCount*dd/encoderCPR % Latest displacement

% Converting to velocity based on current displacement change or using
% previous velocity value, or zero velocity if no change in
% displacement for more than 1s.
    if displace == 0
        t = clock; % current time
        tc = t - te; % Find the change in time
        dt = tc(5)*60 + tc(6); % Convert to second
        if dt > 1
            velp = 0; % Velocity is 0 if no displacement change in
                    % more than 1 second
            te = t; % Update the clock time for change in velocity
        else
            velp = velp; % Else accept previous velocity value
        end
    else
        vel = displace/dt; % Else calculate new velocity value
        te = t; % Update the clock time for change in velocity
    end

disp = dist + displace; % Increment by the latest displacement
```

---

---

## D.11 Model for mobile platform and flexible manipulator

This function computes the simulation of the velocity of the mobile platform using equation (3.48), and then convert it into acceleration.

It then computes the displacement at the tip of the flexible manipulator using equation (3.130) and (3.131) and using the parameters for Table 5.1.

Source code:

---

```
% This function models the velocity of the mobile platform using motor
% model of equation 3.48

function [z] = motor_test(velp,dlate,ln,Vin)

% The parameters of the mobile flexible manipulator Table 5.3. This
% example uses payload 0.02kg (20g).
% Weight of flexible manipulator is 107.133g, weight of the mobile
% platform is 1.3072kg, therefore Tmass = 1.3072 + 0.107133 = 1.414333
Load = 0.02; Tmass = 1.414333 + Load; mu = 0.086; mc = 0.03; x1 = 0.09;
x2 = 0.14; lx = 0.23; h = 0.045;
Rwheel = 0.0325; Rcaster = 0.02; Raxle = 0.004; Jmotor = 0.08083;
acc = 0.05;

% Using equation 3.39, which utilizes equation 3.35, 3.33, 3.31, 3.29
% and 3.25
Tm = mu*Rwheel*(Tmass*9.81*x1 - Tmass*acc*h)/lx + 2*(mc*(Rcaster-
Raxle)*(Tmass*9.81*x2+Tmass*acc*h)/(2*lx));

J = Jmotor + (Tmass)*Rwheel^2; % Using equation 3.48
b = 0.00102; % viscous friction
Ki = 0.1297; % Torque constant of the motor
Ke = 0.482; % Back-EMF constant of the motor
R = 4.36; % Armature resistance
L = 0.0019; % Armature inductance
s = tf('s'); % TF model using a rational function in the Laplace
% variable, s

% Using equation 3.47 to compute Laplace transform of Y(s)/U(s)
P_motor = Ki/((J*s+b)*(L*s+R)+Ki*Ke);
P_mtr = -(L*s+R)/((J*s+b)*(L*s+R)+Ki*Ke);
P_m = [P_motor P_mtr]; %Using equation 3.47

% Simulating torque and voltage input
stnd = nd-st; % Duration of the voltage input
v2 = ones(1,(stnd+1))*(0.00252+Tm); % Torque from mobile platform
v1 = ones(1,(stnd+1))*Vin; % Voltage input
V = [v1; v2]; % voltage and torque inputs to the transfer function
x0 = [0 0]; % Initial inputs
```

```

% Simulate the velocity of motor shaft
t = 0:0.01:ln; % Duration of the simulation
motor_ss = ss(P_m);
h = lsim(motor_ss,V,t,x0); % Angular velocity of motor shaft

Velps = h*Rwheel; % Compute platform velocity using equation 3.49
fprintf(fileID,'%f %f\n',Velps); % Save the platform velocity data

% Converting velocity to acceleration
for i = 2:(stnd+1)
    acc(i) = (h(i)-h(i-1))/1;
    P = acc(i)*(0.2235*0.107133+Load); % m = 0.2235mbeam + load
    defl(i) = P*0.53^3/(3*190e9*2.123e-12); % w = PL^3/3EI
end

% using equation 3.126 for flexible manipulator on mobile platform
% and using the flexible beam parameters in Table 5.1
Kbeam = 0.00096*0.0288*0.53*7308.864*((6*3.14-16)/(4*3.14)) + Load;
Qbeam = 190e9*2.123e-12*3.14^4/(32*0.53^3);
P_beam = (0.107133+Load)/(Kbeam*s + Qbeam);
beam_ss = ss(P_beam);
x = lsim(beam_ss,acc,t,0);

% Simulating vibration using the wave function
[w1,loc] = wave(x(1:1000),Load); % Function call to wave at start
[w2,loc] = wave(x(end-1000:end),Load); % Function call to wave at end

% Parameter of the flexible beam and mobile platform
A = 0.00096*0.0288; % Cross-sectional area of manipulator
L = 0.053; % Length of manipulator
E = 190e9; % Young's modulus
I = 2.123e-12; % Moment of Inertia
Den = 7208.864; % Density
wldotdot = 0; % Initial acceleration of displacement if tip
you = max(w1); % Maximum computed displacement

% Set the state space model matrixs
Mq = [A*L*((6*3.14-16)/4*3.14)/Den+Load  0; 0  mB] % resultant force
      %matrix
Rq = [E*I*3.14^4*w1/(32*L^3); 0]; % Repulsive matrix
Tou = [0;  mB*vdot]; % Input torque
XX1 = [0  0  1  0;
       0  0  0  0;
       0  0  0  0;
       -Mq'*Rq  0  0  0];
XX2 = [0;  0;  0;  Mq'];

% Compute the vibration output of the manipulator
for j = 2:(stnd+1) % add 1
    qdotdot = [you(j); velps(j); wldotdot; acc(j)]; % State of the
    tip displacement . %and velocity
    XX3 = XX1*qdotdot + XX2*Tou;
    yout(j) = [1  0  0  0]*XX3;
if yout(j) < 0
    you(j) = -sqrt(abs(yout(j)));

```

```

else
    you(j) = sqrt(yout(j));
end
end
z(st:nd)=you; % first wave

```

## D.12 Modelling the manipulator vibration

This function use equation 3.75 to compute the vibration at the tip of the flexible manipulator.

Source code:

---

```

% Function use equation 3.75 to simulate the flexible beam vibration

function [x_disp,loc] = wave (wav, load)

% Find the peak of the last vibration cycle and its location
[Xo,loc] = findpeaks(abs(wav)); % Find the peak and its location

mass = 0.107 + load; % weight of the flexible manipulator + payload

% Using interpolation method to find the natural frequency of vibration
h = [0.107 0.117 0.127 0.137 0.147 0.157 0.167]; % total mass of beam +
% payload
i = [16.0719862410149994 14.16 12.8715 11.789484150148837 10.98
10.26975310155627 9.6843890106713]; % i is the natural frequency of the
% vibration determined by arbitrary
% test the beam's vibration at
% respective load in h
w = interp1(h,i,mass); % find the frequency of vibration using the
% total mass of beam

% Determine damping ratio, S
k = w^2*mass;
D = 0.027*sqrt(k*mass);
S = D/(2*sqrt(k*mass)); % damping ratio
Vo = 0; % Taking initial velocity as zero

% This is for use in equation x(t) below
wd = w*sqrt(1-S^2);
ti = atan((Xo*wd)/(Vo + S*w*Xo));
A = sqrt(((Vo + S*w*Xo)^2 + (Xo*wd)^2)/wd^2);
Z = 1;

% Simulate 10000 point of the projected length of vibration in time
for t = 1:10000
    % using equation 3.73
    x(t) = A*exp(-S*w*t*0.01)*sin(wd*t*0.01 + ti*exp(t*0.0000));
end

% To find the last wave is positive cycle or negative cycle

```

```
wav_dif = Xo - wav(loc);
if wav_dif ~= 0
    x = -x; % if latest peak is negative peak, inverse x
end
```

---

### D.13 Model based predictive algorithm

This function performs the prediction of the tip vibration using the model based predictive algorithm, where it use the acquired sensor's data and the simulated platform velocity data to predict the manipulator's tip vibration for 200 time steps ahead.

It saves the data into ekf4.txt file for plotting.

Source code:

---

```
% Model based prediction
% This algorithm use the previous displacement record, the latest
% displacement input and modelled vibration of the MFLM to predict 200
% steps ahead the future vibrational displacement at the tip of the
% flexible manipulator.

function [dlate] =
model_based_predictive(k, img, imgt, pos_diff, tx, velp, ln, Vin, dlate)

Ta=0.0001; % sampling time of accelerometer
fs=60; % sampling frequency of camera
Tc=1/fs; % sampling time of camera

A=[1 Ta; 0 1]; %process state

% Compute latest displacement-----

    if img1 == 1 % Check if latest camera data exist? If yes then...

        pos_camx(imgt)=img; % Update position information using camera
            % data measurement at the time of capture

        % Extrapolate using two previous and current camera data
        p1= [pos_camx(imgp2) pos_camx(imgp1) pos_camx(imgt)];
        p2= [imgp2 imgp1 imgt]; % Time of previous two camera data
        tf= imgt2-imgp; % Time between second previous and current data
        lt= (imgt+1):(imgt+tf); % Length of time to extrapolate
        cm(imgt+1:imgt+tf) = interp1(p2,p1,lt,'spline'); % Extrapolate
        imgp2 = imgp1; imgp1 = imgt; % Update latest camera data time

        dlate = cm(tx); % Find the latest displacement result using
            % extrapolated camera output
    else
```

```

        dlate = A*dlate + pos_diff; % Find latest displacement using
                                   % previous and latest displacement
                                   % from accelerometer estimate
    end

    pos_mbp(k,1) = dlate; % Save sensor estimated displacement to column 1

    % Compute latest displacement end-----

    % Simulate displacement using latest measured displacement, length of
    % time for simulation and input voltage
    d = motor_test(velp,dlate,ln,Vin);

    d_diff = diff(d); % Find the difference in the modelled displacement

    % Predicting 200 steps ahead. Using modelled displacement as input to
    % predict the future vibrational displacement-----

    x_hat_p = dlate; % Latest displacement value

    for kl=1:200
        x_hat_p = A*x_hat_p + d_diff(tx+kl); % Next prediction
    end
    % Prediction end-----

    pos_mbp(k,2) = x_hat_p; % Save predicted output at 200 step to column 2
    pos_mbp(k,3) = tx; % Update latest time to column 3

    % Saving latest MBP and predicted outputs for plotting purpose
    fprintf(mbp, '%f %f\n', pos_mbp); % Save data into mbp.txt file

```

---

## D.14 EKF Type 3 model based predictive algorithm

This function performs the prediction of the tip vibration using the EKF type 3 model based predictive algorithm, where it use the acquired sensor's data and the simulated platform velocity data to predict the manipulator's tip vibration for 200 time steps ahead.

It saves the data into ekf3.txt file for plotting.

Source code:

---

```

% EKF Type 3 model based prediction
% This algorithm, (when camera data is available, KF fusion is based on
% accelerometer and camera. When Camera data is absent, KF fusion is
% based on accelerometer and interpolation of camera data). The time
% update prediction is computer using accelerometer data).
% Then, the previous displacement record and the latest
% displacement input to predict 200 steps ahead the displacement of the
% tip of the flexible manipulator.

```

```

function [] =
EKF_type3_mbp(k,img,imgt,accx,pos_diff,tx,velp,ln,Vin,dlate)

Ta=0.0001; % sampling time of accelerometer
fs=60; % sampling frequency of the camera
Tc=1/fs; % sampling time of camera

A=[1 Ta; 0 1]; %process state
B=[Ta^2/2; Ta]; %process input
C=[1 0];

% measurement noises
cam_noise=1.2; % measurement noise of camera
accelero_noise=1.20; % measurement noise of accelerometer

x_hat = dlate; % Update with previous displacement estimate

% process noise covariance
Q=accelero_noise^2*[(Ta^2/2)^2 (Ta^2/2)*Ta; (Ta^2/2)*Ta Ta^2];

R1=accelero_noise^2; % measurement noise covariance for accelerometer
R2=cam_noise^2; % measurement noise covariance for camera

Px=Q; % P can be set randomly, P will coverage eventually

% Time update from EKF projection-----

    x_hatP = A*x_hat + B*accx; % State time update
    Pminusx = A*Px*A' + Q; % Error covariance time update

% Camera estimated displacement-----

    if img1 == 1 % Check if camera data exist?

        pos_camx(imgt)=img; % Update position information using camera
            % data measurement at the time of capture

        % Extrapolate using previous camera data
        p1= [pos_camx(imgp2) pos_camx(imgp1) pos_camx(imgt)];
        p2= [imgp2 imgp1 imgt]; % Time of previous two camera data
        tf= imgt2-imgp; % Time between second previous and current data
        lt= (imgt+1):(imgt+tf); % Length of time to extrapolate
        cm(imgt+1:imgt+tf) = interp1(p2,p1,lt,'spline'); % Extrapolate
        impg2 = impg1; impg1 = imgt; % Update latest camera data time

    end

% Camera estimate displacement end-----

% Measurement update
zx2= cm(tx); % Find current displacement with extrapolate of camera
data

zx1= A*x_hat + pos_diff; % Find current displacement using previous and
% current displacement from accelerometer data

```

```

% Measurement or observation update-----

% Form the innovation vector (Residuals)
rex1 = zx1 - C*x_hatP;
rex2 = zx2 - C*x_hatP ;

% compute covariance of innovation
Sx1 = C*Pminusx*C' + R1;
Sx2 = C*Pminusx*C' + R2;

% Kalman gain matrix
Kx1 = Pminusx*C'/(Sx1);
Kx2 = Pminusx*C'/(Sx2);

% update state estimate
x_hat = x_hatP + (Kx1*rex1 + Kx2*rex2);

% compute the covariance of the estimation error
ppx = pinv(Pminusx);
CSC = C*C'/(Sx1) + C*C'/(Sx2);
pxinv = ppx + CSC/2;
px = pinv(Pxinv);

dlate = x_hat; % update latest displacement
pos_ekf3(k,1) = x_hat; % EKF estimated displacement to column 1

% Measurement update end-----

% Simulate displacement using latest measured displacement, length of
% time for simulation and input voltage
d = motor_test(velp,dlate,ln,Vin); % Model displacement using latest
% displacement, length of time and
% input voltage

d_diff = diff(d); % Find the difference in the modelled displacement

% Predicting 200 steps ahead. Using modelled displacement as input to
% predict the future vibrational displacement-----

x_hat_p = x_hat; % predict using Extended Kalman filter

for kl=0:200 % Predict 200 time steps ahead
    x_hat_p = A*x_hat_p + d_diff(k+kl); % Predicted displacement
end

% Prediction end-----

pos_ekf3(k,2) = x_hat_p; % Save predicted output at 200 step to column2
pos_ekf3(k,3) = tx; % Update latest time to column 3

% Saving latest EKF3 and predicted outputs for plotting purpose
fprintf(ekf3, '%f %f\n', pos_ekf3); % Save data into mbp.txt file

```

---

## D.15 EKF Type 4 model based predictive algorithm

This function performs the prediction of the tip vibration using the EKF type 4 model based predictive algorithm, where it use the acquired sensor's data and the simulated platform velocity data to predict the manipulator's tip vibration for 200 time steps ahead.

It saves the data into ekf4.txt file for plotting.

Source code:

---

```
% EKF Type 4 model based prediction Kalmanfilter_fusion4v3
% The modified Kalman filter with vibration modelled input and
% windowing acceleration computation.
% Extrapolation of on the previous camera data
% Then, the previous displacement record and the latest
% displacement input to predict 200 steps ahead the displacement of the
% tip of the flexible manipulator.
function [] =
EKF_type4_mbp(k,img,imgt,accx,pos_diff,tx,velp,ln,Vin,dlate)

Ta=0.0001; % sampling time of accelerometer (for 13Khz is 0.00077)
fs=60; % sampling frequency of the camera
Tc=1/fs; % sampling time of camera

A=[1 Ta; 0 1]; %process state
B=[Ta^2/2; Ta]; %process input
C=[1 0];

% measurement noises
cam_noise=1.2; % measurement noise of camera
accelero_noise=1.20; % measurement noise of accelerometer

% initial estimate for manipulator displacement
x_hat = dlate; % Update previous displacement estimate

% process noise covariance
Q=accelero_noise^2*[(Ta^2/2)^2 (Ta^2/2)*Ta; (Ta^2/2)*Ta Ta^2];

R2 = accelero_noise^2; % measurement noise covariance for accelerometer
R1 = cam_noise^2; % measurement noise covariance for camera

Px=Q; % P can be set randomly, P will coverage eventually

d = motor_test(velp,dlate,ln,Vin); % Model displacement using latest
% displacement, length of time and
% input voltage

d_diff = diff(d); % Find the incremental changes of the simulated data

% Time update from EKF projection-----
```

```

x_hatP = A*x_hat + d_diff; % Using the change in modelled
                        % displacement change as input
Pminusx = A*Px*A' + Q; % Error covariance time update

% Camera estimated displacement-----

if img1 == 1 % Check if camera data exist?

    pos_camx(imgt)=img; % Update position information using camera
                        % data measurement at the time of capture

    % Extrapolate using previous camera data
    p1= [pos_camx(imgp2) pos_camx(imgp1) pos_camx(imgt)];
    p2= [imgp2 imgp1 imgt]; % Time of previous two camera data
    tf= imgt2-imgp; % Time between second previous and current data
    lt= (imgt+1):(imgt+tf); % Length of time to extrapolate
    cm(imgt+1:imgt+tf) = interp1(p2,p1,lt,'spline'); % Extrapolate
    imgp2 = imgp1; imgp1 = imgt; % Update latest camera data time

    zx2= cm(tx); % Find current displacement with extrapolate of camera
else
    zx2= d(k); % Use displacement computed by the model at current time
end

% Camera position end-----

zx1= A*x_hat + pos_diff; % Find current displacement using previous and
                        % current displacement from accelerometer data

% Measurement or observation update-----

%Form the innovation vector (Residuals)
rex1 = zx1 - C*x_hatP ;
rex2 = zx2 - C*x_hatP ;

%compute covariance of the innovation
Sx1 = C*Pminusx*C' + R1;
Sx2 = C*Pminusx*C' + R2;

% Kalman gain matrix
Kx1 = Pminusx*C'/(Sx1);
Kx2 = Pminusx*C'/(Sx2);

% update state estimate
x_hat = x_hatP + (Kx1*rex1 + Kx2*rex2)/2;

% compute the covariance of the estimation error
ppx = (eye - (Kx1*C + Kx2*C));
px = ppx*Pminusx;

dlate = x_hat; % update latest displacement
pos_ekf4(k,1) = x_hat; % EKF estimated displacement to column 1

```

```

% Measurement update end-----

% Predicting 200 steps ahead. Using modelled displacement as input to
% predict the future vibrational displacement.

x_hat_p = x_hat; %predict using Kalman filter

for kl=0:200 % Predict 200 time steps ahead
    x_hat_p = A*x_hat_p + d_diff(k+kl); % Predicted displacement
end

% Prediction end-----

pos_ekf3(k,2) = x_hat_p; % Save predicted output at 200 step to column2
pos_ekf3(k,3) = tx; % Update latest time to column 3

% Saving latest EKF4 and predicted outputs for plotting purpose
fprintf(ekf4, '%f %f\n', pos_EKF4); % Save data into ekf4.txt file

```

---

## D.16 Main function

This is the main program that runs the functions for predictive algorithms for model predictive algorithm, EKF type 3 model predictive algorithm and EKF type 4 model predictive algorithm.

Source code:

---

```

% Function that run predictive algorithms
function [] =
main(k, imgt, img1, imgt, accx, vel, posz, pos_diff, tx, disp, velp, te)

fprintf

ln = 10 - t; % Remaining time for the 10 second run duration

% If time ln is within 1 and 10, set Vin to voltage value to step to
if ln<1 | ln>10
    Vin = 0.67; % This sets input voltage Vin to 0.67V
else
    Vin = 0; % Set Vin to zero when not within the 10 second period
end
fprintf(fileID, '%f %f\n', Vin); % Save the input voltage data

% Call model based predictive algorithm
model_based_predictive(k, imgt, imgt, pos_diff, tx, velp, ln, Vin, dlate)

% Call EKF type 3 model based predictive algorithm
EKF_type3_mbp(k, imgt, imgt, accx, pos_diff, tx, velp, ln, Vin, dlate)

```

```
% Call EKF type 3 model based predictive algorithm
EKF_type4_mbp(k, img, imgt, accx, pos_diff, tx, velp, ln, Vin, dlate)
```

```
End
```

---

## D.17 Cross correlation

This function performs the cross correlation of the accelerometer and camera data to calculate the lag of the camera data. It then compute the new displacement data computed from the camera.

Source code:

---

```
% Function that that use cross correlation to find the lag between the
accelerometer's measurement and the camera's measurement
function [img_new] = x_corr(img, accx)

% Find the number of points the camera data lag the accelerometer data
[x_cor, lag] = xcorr(img, accx);
[~, I] = max(abs(x_cor));
lagt = lag(I);

img_new = img(-lagt+1:end); % New camera data without lag
```

---

## D.18 Plotting the vibration and velocity outputs, and voltage inputs

This program load all the data being save and plot the flexible manipulator's tip displacement outputs for model based prediction, EKF type 3 model based prediction and EKF type 4 model based prediction, as well as the platform velocity and input voltage.

Source code:

---

```
% Plotting the vibrational displacement

% Loading the data from txt files
mbp = load('mbp.txt');
ekf3 = load('ekf3.txt');
ekf4 = load('ekf4.txt');
Velps = load('velps.txt');
Vin = load('Vin.txt');
```

```

% Plot output from model based prediction
plot(t,mbp, 'b');
grid;
xlabel('Time (sec)'); ylabel('Displacement (m)');
title('Vibration');
legend('model predict')

hold on
plot(t,ekf3, 'y'); % Plot output from EKF type 3 model based prediction
plot(t,ekf4, 'm'); % Plot output from EKF type 4 model based prediction
hold off

% Plotting for motor_test function-----
% Plotting mobile platform velocity
g = zeros(2800,1);
g(st:nd)= Velps; % velocity of platform
tt = 0:0.01:28-0.01;
figure (2)
plot(tt,g, 'g')
xlabel('Time (sec)'); ylabel('Platform velocity (m/s)');

% Plot input voltage
input = g;
input(st:nd)=v1; %plot input voltage
figure (3)
plot(tt, input, 'r')
xlabel('Time (sec)'); ylabel('Input voltage (V)');

% Plotting simulated deflection of flexible manipulator
figure (4)
plot(x)

% Plotting vibration using wave function
figure (5)
plot(tt,w2)
xlabel('Time (sec)'); ylabel('Manipulator vibrational deflection
(mm)');

z(st:nd)=you; % first wave
figure (6)
hold on
plot(tt*0.4,-z, 'm');
xlabel('Time (sec)'); ylabel('Manipulator vibrational deflection
(mm)');

```

---

## E.1 Arduino code

This program is the Arduino code for the Arduino controller. It set the selectable speeds for 0 to 9 using the input voltage of 5 V.

Source code:

---

```
int motorPin = 3;
int dir = 7;
int apin = 2;
int bpin = 4;

void setup() {
  // setup code, run once:
  pinMode(motorPin, OUTPUT);
  pinMode(dir, OUTPUT);
  pinMode(apin, INPUT);
  pinMode(bpin, INPUT);
  analogWrite(motorPin, 0);
  digitalWrite(dir, LOW);
  Serial.begin(9600);
}

void loop() {
  // main code, run repeatedly:

  if (Serial.available())
  {
    char ch = Serial.read();
    if (ch >= '0' && ch <= '9')
    {
      int speed = ch - '0';
      analogWrite(motorPin, speed *28+3);/*28+3
      for (int i = 0; i <2000; i++)//2 is 690, 9 is 150
          //W2 is 1500, W4 is 390, W9 is 170
      {
        int encA = digitalRead(apin);
        int encB = digitalRead(bpin);
        Serial.print(encA);
        Serial.print(",");
        Serial.println(encB);
      }
      analogWrite(motorPin, 0);
    }
    else if (ch == 'a') // anti-clockwise
    {
```

```
    digitalWrite(dir, HIGH);  
  }  
  else if (ch == 'c') // clockwise  
  {  
    digitalWrite(dir, LOW);  
  }  
}  
}
```

---