Title
Vision and Inertial Sensor Based Drive Trains Control

Permalink
https://escholarship.org/uc/item/5d73p6rz

Author
Cheng, Haifei

Publication Date
2010

Peer reviewed|Thesis/dissertation

Vision and Inertial Sensor Based Drive Trains Control

by

Haifei Cheng

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering-Mechanical Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Masayoshi Tomizuka, Chair
Professor J. Karl Hedrick
Professor Donald E. Sarason

Fall 2010

Vision and Inertial Sensor Based Drive Trains Control

by

Haifei Cheng

Abstract

Vision and Inertial Sensor Based Drive Trains Control

by

Haifei Cheng

Doctor of Philosophy in Engineering-Mechanical Engineering

University of California, Berkeley

Professor Masayoshi Tomizuka, Chair

This dissertation is mainly concerned with the motion control of robot manipulators by utilizing vision and inertial sensors. It is separated into two parts: the first part focuses on vision based online trajectory generation for contour following, and the second part is about inertial and vision sensor based end-effector sensing and control for robot manipulators.

Vision camera is a useful robotic sensor since it mimics human vision and allows for non-contact measurements of the environment. Although there is a considerable amount of research on how to include the vision information in the feedback loop (visual servo), the first part of this dissertation will mainly focus on online reference trajectory generation based on vision sensor information. It is well known that most commercially available vision sensors have a relatively low sampling rate and unavoidable measurement delay due to exposure time, image transmission time and processing time etc. A systematic scheme is proposed to overcome these problems in the first part of this dissertation. The proposed scheme is different from pure visual servo since the joint loop is closed by encoder signals in the inner feedback loop while the vision information is used in the outer loop to generate the reference trajectory. Experimental results of following a variety of contours at selected speeds are presented to validate the effectiveness of the proposed reference trajectory generation scheme.

In the applications of industrial robot manipulators, it is often desirable to obtain accurate end-effector position and velocity information. Estimation based on motor-side encoders alone is often inaccurate due to joint flexibilities and kinematic errors of robot links. A vision based approach may also be insufficient due to its low sampling rate and large measurement delay. With the addition of acceleration measurements, however, a multi-rate kinematic Kalman filter (KKF) with large measurement delay

can be formulated to estimate the end-effector motion accurately without encoder signals. The estimates based on the proposed scheme are utilized as feedback signals for real-time tracking control. The effectiveness of the proposed scheme is demonstrated by experiments on a single joint setup and a two link manipulator.

To my family

Contents

List of Figures

List of Tables

Acknowledgments

During the long journey of my doctoral study, there are a lot of people I would like to acknowledge. Without their help, support, guidance and efforts I could never have reached the heights or explored the depths of Mechanical Engineering into this stage.

The first and the most important person I would like to express my sincere respect and gratitude to, is my advisor Professor Masayoshi Tomizuka. Without his depth knowledge, keen insight, persistent encouragement and long-term financial support, I would never complete my degree. Besides his help in my doctoral study, he is also a nice person with great personality. I would never forget his daily visits when I was hospitalized.

I would like to thank Professor Hedrick, Professor Packard, Professor Sarason and Professor Bartlett for their support and commitment in my qualifying examination. I would like to thank Professor Packard and Professor Horowitz for the moments they spent with me during the past five years. I would like to thank Professor Sarason again for his kindness and willingness to help me review my dissertation in the last minute.

To be honest, I am very lucky to be a member of Mechanical System Control Laboratory. I would like to express my sincere gratitude to all the past and current members for their help in many aspects, especially, Dr. Soo Jeon and Mr. Sugita Sumio for their help in my research.

The years spent in Berkeley would not have been so wonderful without my Chinese friends, I would like to thank them for sharing many precious moments with me and helping me during my hard time.

Thanks also to my family, specially my mother Mei Hong and my father Xiangding Cheng. I also would like to express my warm gratitude to my younger brother, Haishan Cheng, elder sister Hailin Hong.

Chapter 1

Introduction

Mechanical systems with drive trains have many modern industrial applications. A typical drive train in mechanical systems begins with an electrical motor and ends with a link, which may be connected to another drive train or an end-effector. The ultimate goal for control engineers is to design high performance servo systems to control drive trains in a cost-effective yet efficient manner. To achieve this goal, conventional control engineers focus on how to design sophisticated control algorithms based on the limited sensor information. However, as Micro-Electro-Mechanical Systems (MEMS) technology continues to develop, more and more sensors such as accelerometer, gyro, camera and sonar, become readily commercially available. Therefore, modern control engineers more and more often focus on mechatronics approaches [1], which synergetically integrate physical systems with information technology and complex-decision making in the design, manufacturing and operation of industrial products and processes.

This dissertation concerns with two interrelated aspects of the mechatronics approaches to improve the tracking performance of servo system with drive trains. The first part of the dissertation presents an online reference trajectory generation scheme for contour following based on vision sensor information. The second part of the dissertation addresses the inertial and vision sensor based end-effector sensing and control for robot manipulators.

1.1 Background and Related Works

1.1.1 Vision Guided Contour Following

Since vision sensors are able to provide a vast amount of information regarding a robot's operating environment, they are essential for robots which are working in unconstructed environments. Even for constructed environments, vision sensors are able to give the flexibility and freedom of creating working environments. Therefore, a considerable amount of research has been carried out to develop vision based robot control systems.

The first vision based robot control system can be dated back to early 1970s [2]. It completed the task of putting a square prism block into a square box with 5 mm clearance. Since then, the development of vision based control has been fairly slow because of the specialized and expensive hardware associated with vision sensing. However, as computer power and CCD/CMOS technology improve, more and more published research can be found in the vision based control area. A comprehensive review on vision based robot control can be found in [3].

There is a part of vision based control area called visual servo. This term was introduced by Hill and Park in 1979 [4] to distinguish from their earlier approach where the system alternated between taking pictures and moving. The task in visual servo is to control the pose of the robot's end-effector by using visual information. It is a fusion of many related areas, such as image processing, real-time computing, robot kinematics, mechanical dynamics and control theory. Visual servo is different from using vision in hierarchical task level control because it actively controls a robot to manipulate its environment rather than just observing the environment [5].

As mentioned in [6], the "low sampling rate" and "noticeable lag" are major obstacles for most of vision sensors. Although there exist high speed vision systems [7], such systems are usually too large and too expensive for industrial applications. The drawbacks of vision sensors make it difficult to stabilize a visual servo system. However, for robot manipulators, it is not necessary to completely rely on vision sensors in feedback control since they also have encoders for accurate position sensing. As an alternative approach, the vision sensor can also be utilized to generate the reference trajectory instead of providing feedback signals for control. The vision sensor is not in the innermost joint loop, and it is not a direct visual servo system. This scheme is similar to the dynamic look-and-move systems based on the taxonomy introduced by [8].

Several authors have developed various schemes for vision based trajectory generation. Namiki [9] and Rouleau [10] among others primarily focused on catching or grasping a moving object. Feddema [11] considered generating a smooth trajectory between actual image features and desired image features for parts placement. For contour following, visual servo in task space was investigated in [12, 13]. Although six degrees of freedom robot path generation method was presented in [14] based on fusing measurements from vision, force and position sensors, they did not mention how to handle the low sampling rate and the large measurement delay introduced by the camera.

The first part of the dissertation concentrates on how to explore image information to generate reference trajectory for joint level controllers [15, 16]. Chapter 3 proposes an online real-time reference trajectory generation scheme based on images obtained from a camera. Since the reference trajectory is generated in task space, the trajectory conversion from task space to joint space by using manipulator Jacobian is also discussed. Chapter 4 evaluates the performance of the proposed scheme by controlling the two link manipulator with generated trajectories to follow straight line, circular arc and rounded square contours.

1.1.2 Multi-Sensor Based End-effector Sensing and Control

Most industrial robot applications require a robot end-effector to either follow a desired trajectory (tracking) or move to a desired point (regulation) with high speed and accuracy. Precise position and velocity information of the end-effector is essential for achieving this goal. Current servo system configurations, however, impose several limitations to achieve this goal. Industrial robots usually have gear mechanisms with a high reduction ratio to increase joint torques. This configuration consequently introduces joint flexibilities to industrial robots. In most cases, only motor side encoders are available and the exact load side information cannot be measured. A robot model can be used to estimate the load side information from the motor side measurements. The performance of this approach, however, is directly related to the accuracy of the model. Unmodeled dynamics, such as kinematic error in the robot links and joint flexibilities, can severely degrade the performance of this approach.

The idea of employing sensors mounted on the end-effector to improve estimation performance was first proposed over twenty years ago [17]. Although this approach has not yet been widely adapted by industry, a considerable amount of research has been carried out in this area [18, 19, 20, 21, 22]. A vision sensor is often considered as a suitable choice for this purpose [23, 24, 25, 26, 27, 28]. But its low sampling rate and high latency makes it very difficult to directly incorporate it into a control loop, which usually runs at a much higher sampling rate. Alternatively, inexpensive sensors such as accelerometers and gyros can be utilized to improve the load side estimation performance. These sensor measurements, however, are usually contaminated by noise. Hence it is not obvious as to how to utilize these sensors.

Among different end-effector sensing techniques, the kinematic Kalman Filter (KKF) is one of the most well developed one. The main idea of the KKF is to apply Kalman filter to a kinematic model and estimate velocity based on position and acceleration measurements.

The idea of employing kinematic model to estimate position and velocity was first proposed by Friedland [29] in 1973. Instead of using an accelerometer, he assumed that the acceleration was a random constant between successive observations. The random constant was assumed to have zero mean and to be uncorrelated with the acceleration at other intervals. He showed that the position errors in a sampled data system can be kept below the inherent sensor errors if the sampling period was below a certain value which related to position and acceleration noise covariances.

KKF that incorporates accelerometer measurements was first proposed by Shim et al. [19] for a linear motor system. The accelerometer and the encoder measurements were integrated for the velocity estimation. It was experimentally verified that KKF produced smoother estimate with less delay than the estimate obtained by a low pass filtered finite difference approach. They also showed that KKF estimate produced very smooth control actions, which may prolong the life of linear motors. This idea was further extended by Lee and Tomizuka [18] to a multi-rate system, where the sampling rates of the encoder and the accelerometer are different. They demonstrated that the multi-rate KKF performed better than the single-rate KKF. As sensor technology improves, fast sampling rate cameras are frequently utilized in industry for motion detection. Soo et al. formulated a multi-dimensional KKF by combining camera, accelerometer and gyroscope measurements to recover the inter-sample values, compensate for the measurement delay of the camera, and provide the state estimate of the end-effector fast and accurately [20]. The derivation in [20] is for three-dimensional case, but they only did experiments for two-dimensional case to validate the effectiveness of their algorithm. The algorithm is not simple and its real-time implementation is not straightforward.

Although there has been a considerable amount of research about KKF, they mainly concerned with integrating different kinds of sensor measurements to estimate the motion of the end-effector. None of them, however, discussed the closed loop stability of the system which included KKF estimate as feedback signals.

The second part of the dissertation presents a simple yet effective motion estimation scheme for the end-effector based on inertial and vision sensors. It focuses on how to identify the covariance ratio of process noise and measurement noise and how to handle the sensors with different sampling rates and large measurement delay [30]. More specifically, open loop frequency responses are measured for the ratio identification. A lifting technique is applied to address the multi-rate problem while a state augmentation technique is used to account for the measurement delay. These techniques will be formulated for one-dimensional case and two-dimensional case separately. More importantly, the closed loop stability of the KKF feedback system is analyzed and the experimental results of the KKF feedback system are discussed.

1.2 Dissertation Contributions

This dissertation makes several contributions in terms of vision based contour following and multi-sensor based end-effector sensing and control for robot manipulators.

- **Real-time online trajectory generation for a two link manipulator:** This dissertation proposes an online reference trajectory generation scheme for a two link manipulator based on the images captured by a camera. The reference trajectory is utilized for contour following control. The contours to be followed are approximated either as a straight line or as a circular arc. The reference trajectories are first generated in task space, and then converted to joint space by the manipulator Jacobian for independent joint loop control. Experimental contour tracking results are presented.

- **Inertia and vision sensor based end-effector sensing and control:** This dissertation proves that the steady state gain of the kinematic Kalman filter depends only on the covariance ratio of process noise and measurement noise. The process of identifying the proper covariance ratio from the open loop frequency responses is also addressed. A multi-rate motion estimation and control scheme for a single joint setup and a two link manipulator based on the kinematic Kalman filter is proposed. For the single joint setup, the scheme assumes that the encoder has a low sampling rate and a large measurement delay to better mimic the sensing capabilities of vision sensors. For the two link manipulator, the position measurement is from a real vision sensor. The effectiveness of the proposed scheme is validated by experiments on the single joint setup and the two link manipulator respectively.

1.3 Dissertation Overview

The remainder of the dissertation is organized as follows:

Chapter 2 introduces a single joint setup and a two link manipulator. More specifically, system identification and friction identification are performed for the single joint setup and system dynamics are given for the two link manipulator.

Chapter 3 proposes an online real-time reference trajectory generation scheme based on images obtained from a camera. Since the reference trajectory is generated in task space, the conversion from task space to joint space by manipulator Jacobian is also addressed.

Chapter 4 evaluates the effectiveness of the proposed trajectory generation scheme by using a two link manipulator to follow different shapes of contours at selected speeds.

Chapter 5 introduces the kinematic Kalman filter and its design. The open loop frequency responses with different noise covariance ratios are measured. The relation of sensor noise covariance ratio to the steady state Kalman filter gain is also discussed.

Chapter 6 presents an end-effector motion estimation scheme based on the measurements of a camera and an accelerometer. Due to the natural drawbacks of the two sensors, a multi-rate kinematic Kalman filter with a large measurement delay is formulated. The proposed scheme is validated on both the single joint setup and the two link manipulator.

Chapter 7 gives the tracking performances of KKF feedback system for both the single joint setup and the two link manipulator. The tracking performance is evaluated by using encoder feedback as a standard benchmark. The closed loop stability of KKF feedback system is also discussed.

Chapter 8 summarizes the dissertation and mention some potential areas of future research.

Chapter 2

NSK Direct Drive Manipulators

In this chapter, a single joint setup and a two link manipulator are introduced. For the single joint setup, an open loop frequency response is measured and it is fitted to a first order model. The friction of the single joint setup is identified by measuring motor torques at different angular velocities. For the two link manipulator, a dynamic model is given and its parameters are lumped into three constants.

2.1 Introduction

Over the years, robot manipulators have been widely used for various industrial applications, such as assembling, packaging, painting, welding, and so forth. They have been recognized as an integral part of automation since 1960[31]. Industrial robots usually consist of multiple drive trains. A typical drive train begins with an electrical motor and ends with a link, which may be connected to another drive train or end-effector. The attached end-effector varies depending on the robot's applications. For assembly machines, the end-effector can potentially be grippers or welding guns. Industrial robot applications usually require the robot end-effector either to follow a desired trajectory (tracking) or to move to a desired point (regulation) with high speed and accuracy.

The drive trains in the robot manipulators can be classified into two categories: direct drive and indirect drive. In direct drive trains, motor shafts are directly connected to links. The motor in direct drive trains usually has high torque capacity in order to drive the link directly. On the other hand, in indirect drive trains, motor shafts and links are connected via gear mechanisms. Due to the joint compliance that results from the gear mechanism, the indirect drive train has two types of performances: motor side and load side. To control the end-effector, the load side information is essential. However, it is difficult to estimate the load side information accurately from the motor encoder because of kinematic errors of the links, joint flexibility of gear mechanisms and so on.

This dissertation will focus on using sensors mounted on the end-effector to estimate the load side information. The performance of this estimation will be evaluated based on the actual load side information. Since the actual load side information is difficult to obtain for indirect drives without the use of specific experimental hardware, this dissertation will utilize a direct drive mechanism to simplify the task of obtaining accurate load side information.

2.2 Single Joint Setup

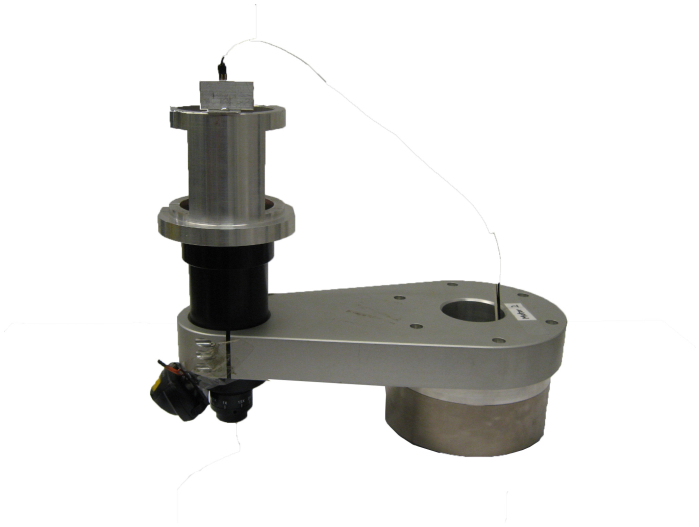

Figure 2.1: Single joint setup

Figure 2.1 shows a single joint direct drive setup. The motor model for this setup is PS3015, which is provided by NSK Ltd. The motor parameters are given in Table 2.1. The encoder resolution may vary depending on the maximum rotational speed. There are inertial and vision sensors installed at the center point of the end-effector.

Table 2.1: Parameters of the motor for single joint setup

Property	Value	Unit
Motor outer diameter	150	mm
Maximum output torque	15	N.m
Maximum rotational speed	1	rev/s
Encoder resolution	3,125,000	count/rev

2.2.1 System Identification

System identification is essential for model based control. Among all the system identification techniques, a sine by sine test is a conceptually simple and effective approach. This technique has been extensively employed in industry to obtain accurate plant models.

If the input is torque and the output is angular velocity, the transfer function of the single joint setup can be written as

$$P(s) = \frac{b}{s+a} \tag{2.1}$$

where a,b are constants to be identified.

Since Coulomb friction effects are not considered in the linear transfer function (2.1), to neglect the effect of Coulomb friction in experiment, the motor has to provide a considerably large torque to overcome Coulomb friction effects. This implies that the motor cannot be run at low speeds for the system identification.

Because the cut-off frequency of the single joint setup is very low, the setup has to run in a very low frequency range to ensure that the cut-off frequency appears in the frequency response if open loop system identification is utilized. It means that the period of the sinusoidal input for that range is very long, and the motor has to travel at a high speed, which may be dangerous. To overcome the drawback of the open loop system identification, a closed loop control system with a proportional gain K is utilized for the system identification as shown in Figure 2.2.

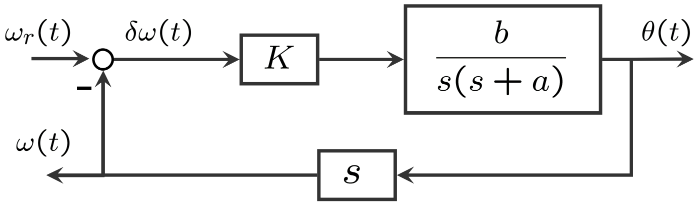

Figure 2.2: Block diagram of closed loop system identification

In Figure 2.2, θ is the joint angle, ω_r is the reference angular velocity and ω is the measured angular velocity. The transfer function from $\omega_r(s)$ to $\omega(s)$ is

$$\frac{\omega(s)}{\omega_r(s)} = \frac{Kb}{s+a+Kb}. \tag{2.2}$$

By increasing K, the bandwidth of the closed loop system is increased.

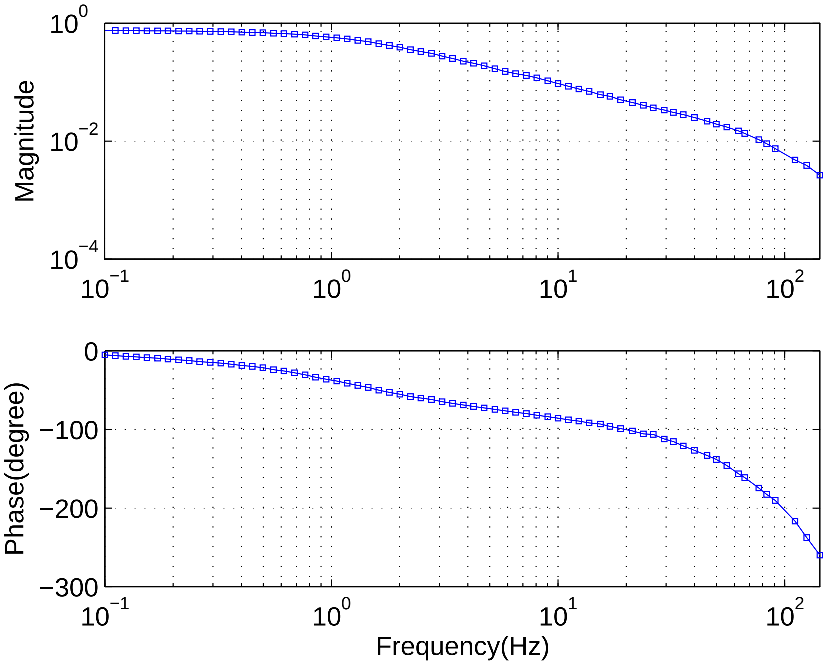

Figure 2.3: Frequency response of single joint setup

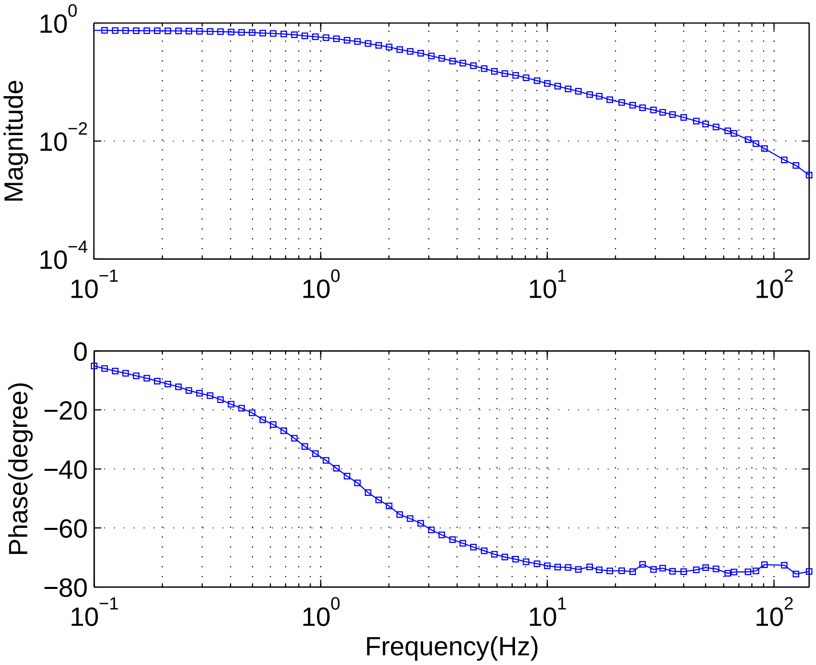

Figure 2.4: Frequency response of single joint setup after delay compensation

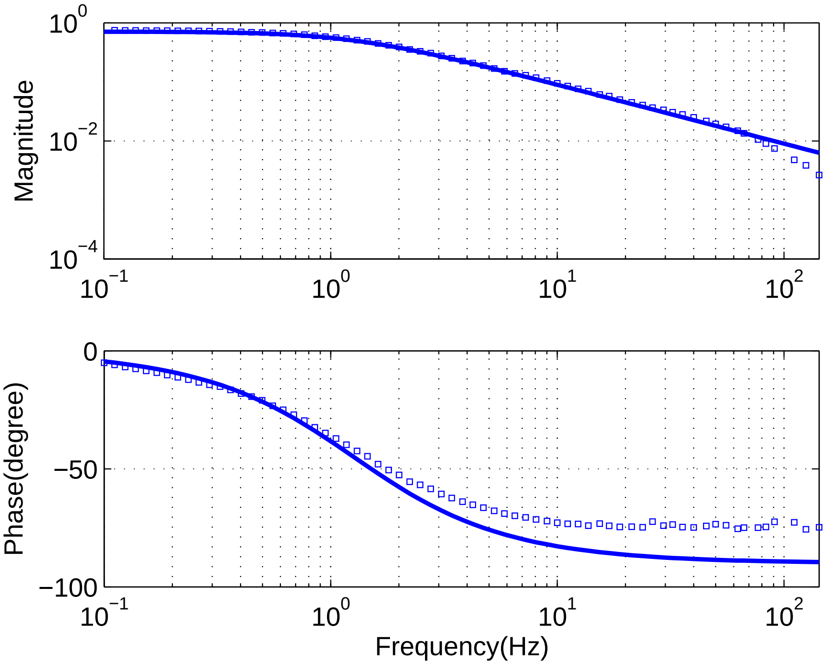

Figure 2.5: Least square curve fitting for frequency response

Figure 2.3 shows the closed loop frequency response of the single joint setup. Here the proportional gain is $K = 5000$. From Figure 2.3 we can see that the bandwidth of the closed loop system is about 1 Hz. The closed loop system is still a first order system, and the phase should be within -90 degree. However, we can see that the phase drops beyond -90 degree for frequencies higher than 10 Hz. Actually, if the frequency axis for the plot is in a linear scale instead of a logarithmic scale, it can be seen that the phase drops linearly as frequency increases. It means that the closed loop system has a noticeable delay.

Based on the frequency response in high frequency range, the delay time is estimated to be 3.6 ms by a least square curve fit. Figure 2.4 shows the frequency response of the single joint setup after delay compensation.

The frequency response showed in Figure 2.4 is fitted to a first order model by the least square method. Figure 2.5 shows the curve fitting result. The magnitude of the frequency response matches the first order model perfectly below 100 Hz. Although the phase of the frequency response does not match the first order model very well, it is still acceptable.

Two constants in the plant model (2.2) are identified by the curve fitting. Table 2.2 shows the identified parameters of the plant model.

Table 2.2: Coefficients of the transfer function for single joint setup

Constant	Value
a	2.20
b	0.0011

2.2.2 Friction Identification

Friction is presented in all kinds of mechanical systems as long as they have relative moving parts. It is highly nonlinear and may result in steady state errors, limit cycles, and poor performances [32, 33]. Figure 2.6 shows the experimental results of the motor torques for different rotary speeds of the single joint setup. The data were collected for a low velocity range, i.e. ±2 rad/s, by controlling the motor velocity with a PID (proportional plus integral plus derivative) controller. For reference speeds larger than 0.4 rad/s, the sampling interval is 0.1 rad/s. Since the friction will show certain effects in the low speed range, it is reasonable to reduce the sampling interval to 0.03 rad/s for reference speeds between -0.4 rad/s and $+0.4$ rad/s. Motor torques are obtained by taking average of the motor commands for each reference speed.

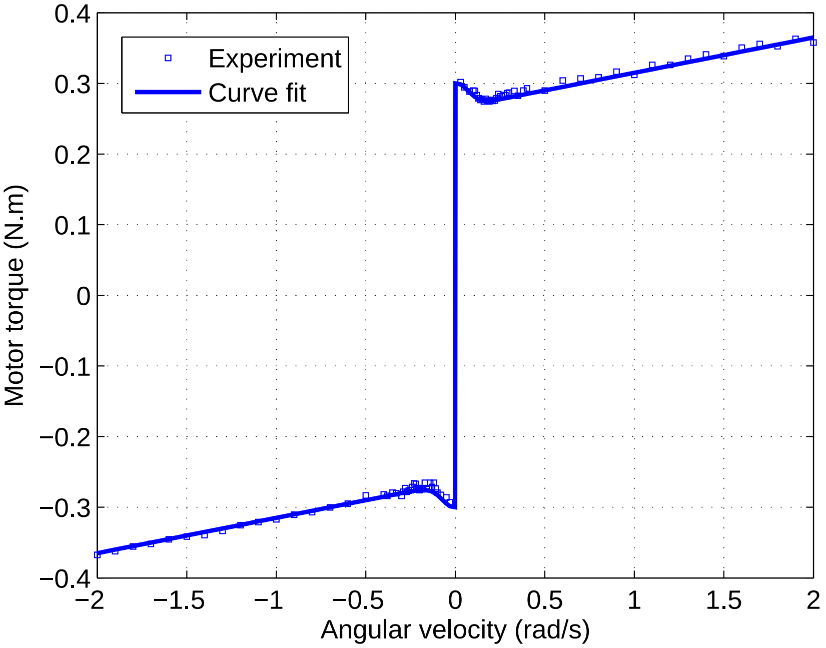

Figure 2.6: Friction for the single joint setup

The curve in Figure 2.6 shows that the setup has static, Coulomb, viscous, and Stribeck friction effects. It can be well fitted by

$$f(\dot{\theta}) = \begin{cases} \left(f_c + (f_s - f_c)\mathbf{e}^{-\dot{\theta}^2/c^2}\right)\mathbf{sgn}(\dot{\theta}) + b\dot{\theta} & \text{if } \dot{\theta} \neq 0 \\ u & \text{if } \dot{\theta} = 0 \text{ and } |u| < f_s \\ f_s\mathbf{sgn}(u) & \text{otherwise} \end{cases} \cdot \tag{2.3}$$

Friction parameters f_c, f_s, b and c can be identified from Figure 2.6. More specifically, the data where the Stribeck effect does not show up, i.e. $|\dot{\theta}| > 0.4$ rad/s, were utilized to identify f_c and b by a least square curve fit. f_s and c can be identified by the data in the relative low speed range, i.e. $|\dot{\theta}| < 0.4$ rad/s. Estimated friction parameters are given in Table 2.3.

Table 2.3: Friction parameters for single joint setup

Parameter	Value	Unit
f_c	0.265	N.m
f_s	0.30	N.m
b	0.05	N.m.s/rad
c	0.11	rad/s

2.3 Two Link Manipulator

Figure 2.7 shows a two link direct drive manipulator. If the first motor and the first link of the manipulator are disassembled, it essentially becomes the single joint setup presented in the previous section. The model of the first motor is PS3090, which is also provided by NSK Ltd. The only difference between PS3090 motor and PS3015 motor is the maximum output torque. The parameters for PS3090 motor are shown in Table 2.4.

Table 2.4: Parameters of the first motor for two link manipulator

Property	Value	Unit
Motor outer diameter	150	mm
Maximum output torque	90	N.m
Maximum rotational speed	1	rev/s
Encoder resolution	3,125,000	count/rev

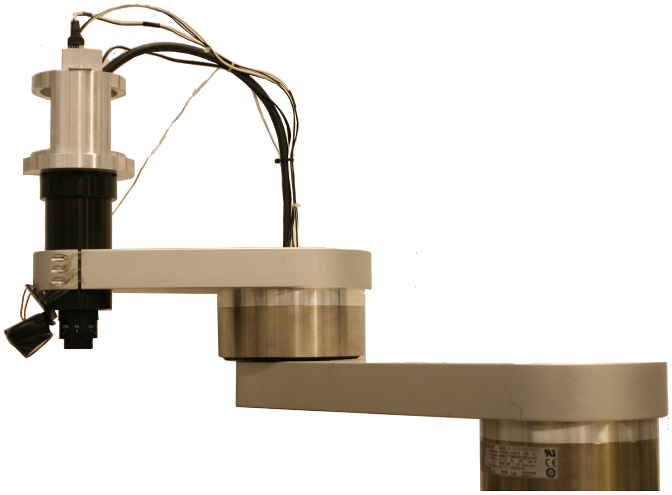

Figure 2.7: Two link manipulator

2.3.1 Manipulator Dynamics

The dynamic mode of the two link manipulator can be described by

$$\begin{aligned} \frac{d}{dt}q(t) &= w(t) && (2.4a) \\ M(q)\frac{d}{dt}w(t) &= \tau(t) - C(q,w) && (2.4b) \end{aligned}$$

where q is a 2×1 vector of joint angular positions; w is a 2×1 vector of joint angular velocities; $M(q)$ is a 2×2 symmetric and positive definite matrix(also called generalized inertia matrix); $\tau(t)$ is a 2×1 vector of joint torques supplied by actuators; $C(q,w)$ is a 2×1 vector of Coriolis and centrifugal forces. Since the manipulator moves only in a horizontal plane, there is no gravity term in the model. The friction term is neglected in the model because it is very small.

$M(q)$ and $C(q,w)$ can be expressed as functions of joint angles, joint angular velocities and lumped constant parameters [34]:

$$\begin{aligned} M(q) &= \begin{bmatrix} \Theta_1 + 2\Theta_3\cos(q_2) & \Theta_2 + \Theta_3\cos(q_2) \\ \Theta_2 + \Theta_3\cos(q_2) & \Theta_2 \end{bmatrix} && (2.5a) \\ C(q,w) &= \begin{bmatrix} -2w_1w_2 + w_2^2 \\ w_1^2 \end{bmatrix}\Theta_3\sin(q_2) && (2.5b) \end{aligned}$$

where

$$\begin{aligned} \Theta_1 &= I_1 + I_2 + (0.25m_1 + m_2)l_1^2 + 0.25m_2l_2^2 + M_2l_1^2 + M_3(l_1^2 + l_2^2) && (2.6a) \\ \Theta_2 &= I_2 + 0.25m_2l_2^2 + M_3l_2^2 && (2.6b) \\ \Theta_3 &= 0.5m_2l_1l_2 + M_3l_1l_2. && (2.6c) \end{aligned}$$

The lumped constant parameters Θ_1, Θ_2 and Θ_3 are functions of the two link manipulator physical parameters as shown in (2.6). Since the moments of inertia of two motor rotors are very small compared with link inertias, it is reasonable to neglect them in (2.5). The mass center points for two links are assumed in the middle of each link for simplicity.

The physical parameters of the two link manipulator are shown in Table 2.5.

Table 2.5: Parameters of the two link manipulator

Notation	Description	Value	Unit
I_1	Moments of inertia of the first link	0.123	Kg.m^2
I_2	Moments of inertia of the second link	0.028	Kg.m^2
l_1	Length of the first link	0.32	m
l_2	Length of the second link	0.215	m
m_1	Mass of the first link	6.83	Kg
m_2	Mass of the second link	3.29	Kg
M_2	Mass of the second motor	5.56	Kg
M_3	Mass of the end-effector	1.05	Kg
Kt_1	Torque constant of the first motor	22.36	N.m/V
Kt_2	Torque constant of the second motor	2.42	N.m/V

2.4 Summary

In this chapter, two setups were introduced. For the single joint setup, system identification by measuring the closed loop frequency response was performed. Based on the frequency response, we found that the closed loop system has a noticeable delay. After delay compensation, a first order plant model was identified for the single joint setup. For this setup, friction identification was also performed by measuring the motor torque at different angular velocities. Based on the velocity-torque plot, the parameters of a friction model were identified. For the two link robot manipulator, the system dynamics was introduced and its associated parameters were given in a table. This information is utilized later in the dissertation when performing model based control.

Part I

Vision Guided Contour Following

Chapter 3

Trajectory Generation Based on Image Information

In this chapter, a reference trajectory generation scheme based on the image obtained from a camera is introduced. Since the field of view (FOV) of the camera is very small, the contour to be followed is approximated as either a straight line or a circular arc. For each case, reference trajectories are generated in task space. In order to control a two link manipulator, the reference trajectory is converted from task space to joint space. A method to achieve this is presented in the last section of this chapter.

3.1 Introduction

Since vision sensors are able to acquire a large amount of information about the environment where robots operate, a considerable amount of research has been carried out on vision based control. Visual servo, where vision sensors provide closed loop position control for a robot end-effector, has been studied. A comprehensive review on visual servo can be found in [5]. A more recent review which focuses on vision based control of robot manipulators can be found in [3].

As we all know, low sampling rate and large measurement delay are major obstacles for most commercially available vision sensors. For robot manipulators, however, it is not necessary to totally rely on vision sensors in feedback control since they have accurate position measurements provided by encoders.

In the following two chapters, we propose a scheme that the vision sensor is utilized to generate the reference trajectory for the position feedback loop to achieve high accuracy contour following, and the joint loop is closed by encoder signals. The block diagram of the scheme is shown in Figure 3.1. The vision sensor measurement is utilized for reference trajectory generation only. The two link manipulator is controlled by a joint loop controller based on the reference trajectory and the encoder measurement. Since the vision sensor is not in the innermost joint loop, it is not a direct visual servo. This scheme is similar to the dynamic look-and-move systems based on the taxonomy introduced by [8].

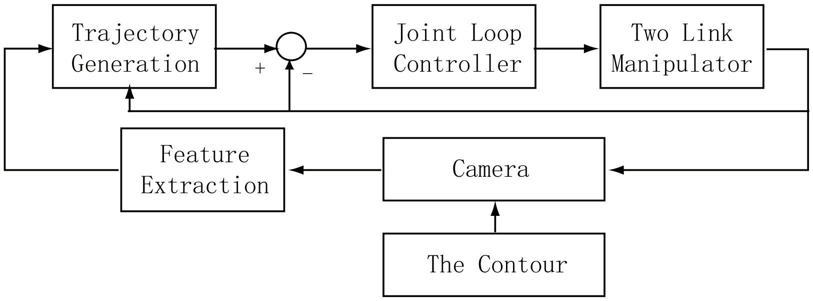

Figure 3.1: Block diagram of vision guided tracking

The formulation of the proposed scheme is based on a two link manipulator. The system configuration is shown in Figure 3.2. The rectangle on the end-effector of the two link manipulator represents the FOV of the camera. The rounded rectangle represents a contour. There are two planes in the figure, one is image plane, shown by $x - y$, which has a left-handed coordinate system for consistency with the pixel information obtained from image processing. The other is task plane, shown by $X - Y$.

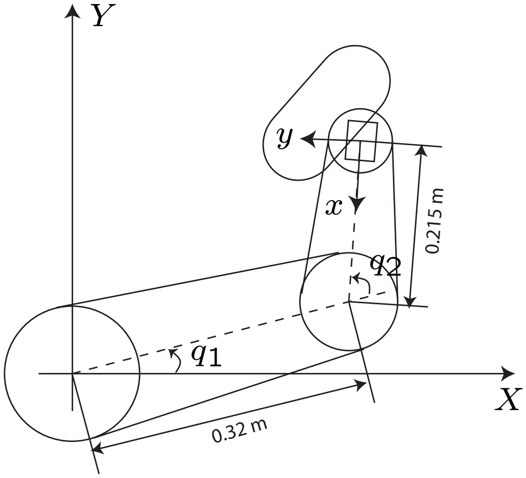

Figure 3.2: System configuration for contour following

Let (x_o, y_o) be the center of the FOV of the camera in the task plane $X - Y$. The robot forward kinematics is given by

$$\begin{bmatrix} x_o \\ y_o \end{bmatrix} = \begin{bmatrix} f_1(q_1, q_2) \\ f_2(q_1, q_2) \end{bmatrix} \tag{3.1}$$

where

$$\begin{aligned} f_1(q_1, q_2) &= l_1 \cos(q_1) + l_2 \cos(q_1 + q_2) \\ f_2(q_1, q_2) &= l_1 \sin(q_1) + l_2 \sin(q_1 + q_2) \end{aligned}$$

where q_1 and q_2 are joint angles, and l_1 and l_2 are lengths of the first link and the second link respectively.

3.2 Trajectory Generation in Task Space

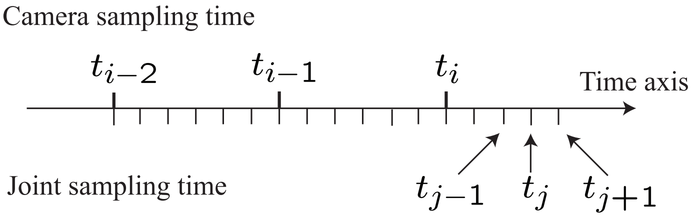

Figure 3.3: Time axis for trajectory generation

As shown in Figure 3.3, the joint sampling time is denoted by t_j and the camera sampling time is denoted by t_i. The joint sampling period $\Delta t_J = t_j - t_{j-1}$ is usually much shorter than the camera sampling period $\Delta t_C = t_i - t_{i-1}$. The sampling period ratio, $\Delta t_C/\Delta t_J$, is assumed to be N. Furthermore, it is assumed that at time t_j, the image information at the most recent camera sampling time t_i is not available due to the measurement delay. Therefore, the image information at time t_{i-1} is utilized to generate the reference trajectory for joint sampling time t_j. Once the image information at time t_i is available, it will be utilized for generating reference trajectories for future joint sampling times.

For high precision motion control, the FOV of a camera has to be very small due to the limited resolution of the image sensor in the camera. The small FOV of the camera and the image delay put a limit on contour tracking speed. The robot cannot move very fast because of the camera delay. Otherwise it may move out of an image when the image information is available. The desired tracking speed V, short side length of the FOV of the camera L, camera sampling period T and delay D have to satisfy the following condition,

$$V \cdot (T + D) < \frac{L}{2}. \tag{3.2}$$

Since the FOV of the camera is very small, it is reasonable to assume that there is no sharp corner. The contour in the FOV of the camera can be approximated as either a straight line or an circular arc based on its curvature. The reference trajectory generation in the task plane is discussed for each type of approximation.

3.2.1 Straight Line Trajectory Generation

For the straight line contour, the partial contour to be followed in the FOV of the camera is shown in Figure 3.4. The dashed rectangle represents the FOV of the camera. The origin, O, the two axes, x and y, of the image plane are defined in the figure. The image feature, $\overrightarrow{OC}$, is defined as the vector starting from the origin O and pointing perpendicularly to the contour AB. The magnitude of the image feature $\overrightarrow{OC}$ in task plane is the tracking error.

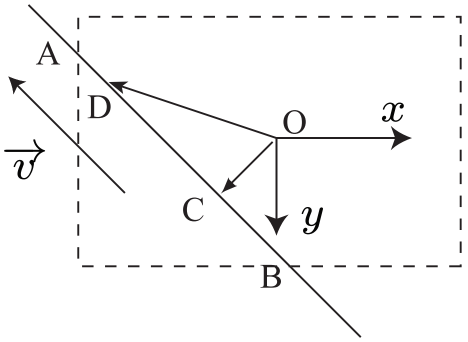

Figure 3.4: Schematic of trajectory generation for straight line contour

Note that the coordinates of $A(x_A, y_A)$, $B(x_B, y_B)$ can be directly obtained from image processing. By analytic plane geometry knowledge, the image feature $\overrightarrow{OC}_i$ in image plane can be expressed as

$$\overrightarrow{OC}_i = \frac{x_B y_A - y_B x_A}{(x_B - x_A)^2 + (y_B - y_A)^2} \begin{bmatrix} -(y_B - y_A) \\ (x_B - x_A) \end{bmatrix} \tag{3.3}$$

where subscript i means that the vector is in the image plane.

Once the corresponding image feature in the image plane $\overrightarrow{OC}_i$ is known, the image feature in the task plane can be obtained by

$$\overrightarrow{OC} = kR(q)\overrightarrow{OC}_i \tag{3.4}$$

where $k \in \mathbb{R}$ is a scaling factor from the image plane to the task plane, $R(q) \in \mathbb{R}^{2\times 2}$ is a rotation and flipping matrix given by

$$R(q) = \begin{bmatrix} -\cos(q_1 + q_2) & -\sin(q_1 + q_2) \\ -\sin(q_1 + q_2) & \cos(q_1 + q_2) \end{bmatrix} \tag{3.5}$$

where q_1 and q_2 are joint angles of the first link and the second link respectively.

For straight line tracking, the reference velocity in the task plane is a constant within one camera sampling period. It can be written as

$$\overrightarrow{v}_r = V \cdot R(q) \frac{\overrightarrow{AB}}{|AB|} \tag{3.6}$$

where $V \in \mathbb{R}$ is the desired tracking speed, $|AB|$ is Euclidean norm of $\overrightarrow{AB}$.

The schematic of trajectory generation for straight line contours is shown in Figure 3.4. The end-effector position is located at the origin O at time t_{i-1}. $\overrightarrow{OC}$ is the image feature. D is the desired end-effector position for time $t_j = t_i + l \cdot \Delta t_J$, where $l = 0, \cdots, N-1$. Therefore,

$$\overrightarrow{CD} = \overrightarrow{v}_r(\Delta t_C + l \cdot \Delta t_J), \quad l = 0, \cdots, N-1. \tag{3.7}$$

$\overrightarrow{OD}$ is the vector which will be utilized to generate reference positions for successive joint sampling times, and it can be obtained by

$$\overrightarrow{OD} = \overrightarrow{OC} + \overrightarrow{CD}. \tag{3.8}$$

For the straight line contour following, the reference acceleration in task plane is zero, i.e.

$$\overrightarrow{a}_r = 0. \tag{3.9}$$

3.2.2 Circular Arc Trajectory Generation

For the contour in the FOV of a camera that can be approximated as a circular arc, the circle center point coordinates and the radius are essential information for trajectory generation. This information can be obtained by curve fitting on a group of sample points on the circular arc. The curve fitting can be formulated as a least square problem.

The standard equation for a circle is

$$(x - x_C)^2 + (y - y_C)^2 = r^2 \tag{3.10}$$

where (x_C, y_C) is the center, r is the radius. It can be represented as

$$x^2 + y^2 + ax + by + c = 0 \tag{3.11}$$

where $a = -2x_C$, $b = -2y_C$ and $c = x_C^2 + y_C^2 - r^2$.

For a group of sample points (x_1, y_1),(x_2, y_2),$\cdots$,(x_n, y_n) the observation model is

$$\underbrace{\begin{bmatrix} x_1^2 + y_1^2 \\ x_2^2 + y_2^2 \\ \vdots \\ x_n^2 + y_n^2 \end{bmatrix}}_{Y} = \underbrace{\begin{bmatrix} -x_1 & -y_1 & -1 \\ -x_2 & -y_2 & -1 \\ \vdots & \vdots & \vdots \\ -x_n & -y_n & -1 \end{bmatrix}}_{M} \underbrace{\begin{bmatrix} a \\ b \\ c \end{bmatrix}}_{\theta} + \underbrace{\begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}}_{e} \tag{3.12}$$

which can be written as

$$Y = M \cdot \theta + e. \tag{3.13}$$

The least square estimation for θ is

$$\hat{\theta}^{LS} = (M^T M)^{-1} M^T Y. \tag{3.14}$$

Therefore, the estimated circle center coordinates and radius are

$$\hat{x}_C = -\hat{a}/2 \tag{3.15a}$$
$$\hat{y}_C = -\hat{b}/2 \tag{3.15b}$$
$$\hat{r} = \sqrt{\hat{x}_C^2 + \hat{y}_C^2 - \hat{c}}. \tag{3.15c}$$

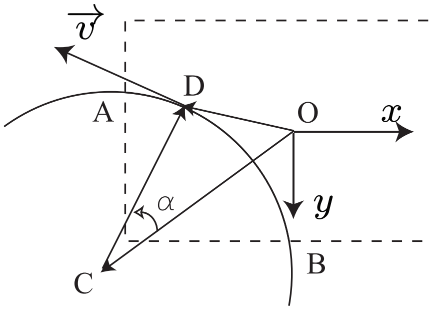

Figure 3.5: Schematic of trajectory generation for circular arc contour

The schematic of trajectory generation for the circular arc is shown in Figure 3.5. The image feature $\overrightarrow{OC}$ is defined as the vector originating at origin O and ending at the circle center point. The tracking error is defined as the difference between the circle radius and magnitude of the image feature in the task plane.

Suppose an image is taken at time t_{i-1} and we want to generate reference trajectory for time t_j. D is the desired end-effector position for time t_j. The angle from $\overrightarrow{CO}$ to $\overrightarrow{CD}$ is α, which can be obtained by

$$\alpha = \frac{V}{r}(\Delta t_C + l \cdot \Delta t_J), \quad l = 0, \cdots, N-1. \tag{3.16}$$

$\overrightarrow{CD}$ can be written as

$$\overrightarrow{CD} = -r \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \frac{\overrightarrow{OC}}{|OC|}. \tag{3.17}$$

$\overrightarrow{OD}$ is the vector which will be utilized to generate reference positions for following successive joint sampling times, and it can be obtained by

$$\overrightarrow{OD} = \overrightarrow{OC} + \overrightarrow{CD}. \tag{3.18}$$

The reference velocity and acceleration for time t_j in the task plane are given by

$$\begin{aligned} \overrightarrow{v}_r(t_j) &= \frac{V}{r} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \overrightarrow{CD} & \text{(3.19a)} \\ \overrightarrow{a}_r(t_j) &= -\frac{V^2}{r^2} \overrightarrow{CD}. & \text{(3.19b)} \end{aligned}$$

3.3 Trajectory Generation in Joint Space

In the previous section, the reference trajectory was generated in the task plane. For joint level control, it has to be converted to the joint space for independent joint loop control. Suppose that $J_f(q)$ is the two link manipulator Jacobian,

$$J_f(q) = \begin{bmatrix} \frac{\partial}{\partial q_1} f_1(q_1, q_2) & \frac{\partial}{\partial q_2} f_1(q_1, q_2) \\ \frac{\partial}{\partial q_1} f_2(q_1, q_2) & \frac{\partial}{\partial q_2} f_2(q_1, q_2) \end{bmatrix} \tag{3.20}$$

which can also be expressed as

$$J_f(q) = \begin{bmatrix} -l_1 \sin(q_1) - l_2 \sin(q_1 + q_2) & -l_2 \sin(q_1 + q_2) \\ l_1 \cos(q_1) + l_2 \cos(q_1 + q_2) & l_2 \cos(q_1 + q_2) \end{bmatrix}. \tag{3.21}$$

The manipulator Jacobian $J_f(q)$ is essential for trajectory conversion. It is easy to check that $J_f(q)$ is singular if and only if $q_2 = n\pi, n \in \mathbb{N}$, If the contour is within the reachable area of the robot arm, the reference joint angle and angular velocity for time t_j are given by

$$\begin{aligned} q_r(t_j) &= q(t_{i-1}) + J_f^{-1}\left(q(t_{i-1})\right) \cdot \overrightarrow{OD} & \text{(3.22a)} \\ \dot{q}_r(t_j) &= J_f^{-1}\left(q(t_j)\right) \cdot \overrightarrow{v}_r(t_j). & \text{(3.22b)} \end{aligned}$$

In order to avoid taking derivative of the inverse manipulator Jacobian, which requires a considerable amount of computation, the reference joint angular acceleration is calculated by using $q_r(t_j)$ and $\dot{q}_r(t_j)$. By taking derivative of $\overrightarrow{v}_r(t_j) = J_f(q(t_j))\dot{q}_r(t_j)$, we can get

$$\overrightarrow{a}_r(t_j) = \dot{J}_f(q(t_j))\dot{q}_r(t_j) + J_f(q(t_j))\ddot{q}_r(t_j). \tag{3.23}$$

Therefore, the reference joint angular acceleration is given by

$$\ddot{q}_r(t_j) = J_f^{-1}(q(t_j))(\overrightarrow{a}_r(t_j) - \dot{J}_f(q(t_j))\dot{q}_r(t_j)). \tag{3.24}$$

3.4 Summary

This chapter proposed a reference trajectory generation scheme by utilizing information from joint encoders and a vision camera. Since the FOV of the camera was very small, the contours to be followed were either approximated as a straight line or as a circular arc. Based on different approximations, the reference trajectory was first generated in the task space, then it was converted to the joint space by using the manipulator Jacobian. Due to different sampling rates of the joint loop controller and the camera, one image was utilized to generate reference trajectories for several joint sampling times. An important characteristic of the proposed scheme was that it allowed the vision sensor input at any time. Although the formulation of the scheme was for two dimensional contour following, it can be easily extended to three dimensional case. In the next chapter, we will present real-time implementations of the proposed scheme on the two link manipulator.

Chapter 4

Contour Following Based on Generated Trajectory

In this chapter, experimental results for following different shape contours based on the trajectory generated in the previous chapter are discussed. Since experiments are performed on a two link manipulator, a model based independent joint loop control scheme for the two link manipulator is presented and the experimental setups are introduced.

4.1 Introduction

Contour following has a broad range of industrial applications such as welding, metal cutting, painting, and inspection and so on. A traditional approach for such applications is generating reference trajectories for robot joints offline based on contour information, which is assumed to be known in advance. The obvious disadvantage of this approach is the assumption. In practical applications, more often than not, the contour to be followed is known only approximately because it depends on the positioning and orientation errors of parts, among other factors.

With the help of a vision camera, a robot manipulator can obtain position information regarding the contour in real time. Therefore, it is not necessary to know the absolute position of the contour in advance.

In the previous chapter, we proposed a reference trajectory generation scheme for robot manipulators based on contour information captured by a camera. This chapter presents experimental results of controlling the two link manipulator to follow contours in real-time by the generated reference trajectory.

4.2 Joint Space Control

The block diagram of vision guided tracking is shown in Figure 3.1. Based on the reference trajectories generated in joint space (Chapter 3), the two link manipulator is controlled by a joint level controller as shown in Figure 4.1.

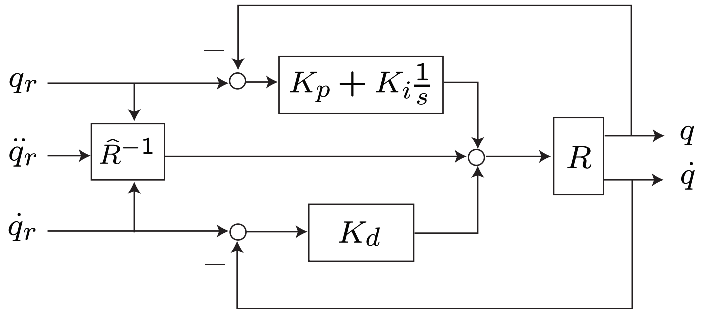

Figure 4.1: Joint loop controller for two link manipulator

Joint level control is based on a feedforward controller plus a PD controller [35]. A weak integral controller (i.e. a small I-gain) is added to the controller to minimize the tracking error caused by Coulomb friction. As shown in Figure 4.1, R is the two link manipulator dynamics. $\hat{R}^{-1}$ is the identified model of inverse two link manipulator dynamics. K_p and K_d are proportional and derivative gains respectively. The reference acceleration is utilized for feedforward control while the reference joint angle and angular velocity are utilized for both feedforward and feedback control.

The control law is given by

$$\begin{aligned} \tau &= \hat{R}^{-1}(q_r, \dot{q}_r, \ddot{q}_r) + K_p(q_r - q) \\ &\quad + K_i \int (q_r - q)dt + K_d(\dot{q}_r - \dot{q}). \end{aligned} \tag{4.1}$$

4.3 Experimental Setup

To show the effectiveness of the proposed trajectory generation scheme presented in the previous chapter, experiments were performed on a two link direct drive manipulator. Figure 4.2 shows the experimental setup. A CMOS camera (Photonfocus CMOS monochrome camera model MV-D640(c)) of focal length $f = 6$ cm is mounted on the end-effector of the two link manipulator. Full 640×480-pixel images can be acquired at frame rate 100 Hz under external trigger condition, and each pixel is quantized into 256 gray levels. The model of the lens is Edmund VZM 300. The FOV of the camera is 8 mm×6 mm when the magnification is 0.75X. The data and images

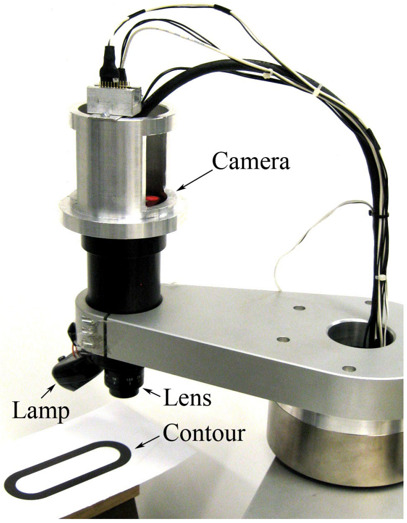

Figure 4.2: Experimental setup for contour following

were collected by the National Instruments data acquisition board PCI-7831R and image acquisition board PCIe-1429, respectively. The real-time system is National Instruments LabVIEW ETS real-time module 8.2.

The camera sampling period was 10 ms, the joint loop sampling period was 1 ms, and the image delay was within 10 ms. The bandwidth of first link and second link were set at 10 Hz and 15 Hz respectively. The contour to be followed was printed on a piece of white photo paper with a high resolution printer. Experiments of three kinds of contours at two different tracking speeds were performed to demonstrate the feasibility of the proposed scheme. For each case, tracking error and tracking speed profiles were plotted. Tracking error was directly calculated from the image feature, so its sampling frequency was the same as camera's, which was 100 Hz, while tracking speed was calculated by the manipulator Jacobian and its sampling frequency was 1 kHz.

4.4 Experimental Results

4.4.1 Straight Line Tracking

The error and speed profiles for tracking a straight line at the speed of 5 cm/s are shown in Figure 4.3. The negative sign means that the robot end-effector is on the other side of the straight line. From the error profile, it can be seen that the tracking error is bounded within 25 μm, except for the starting error due to a sudden jump of

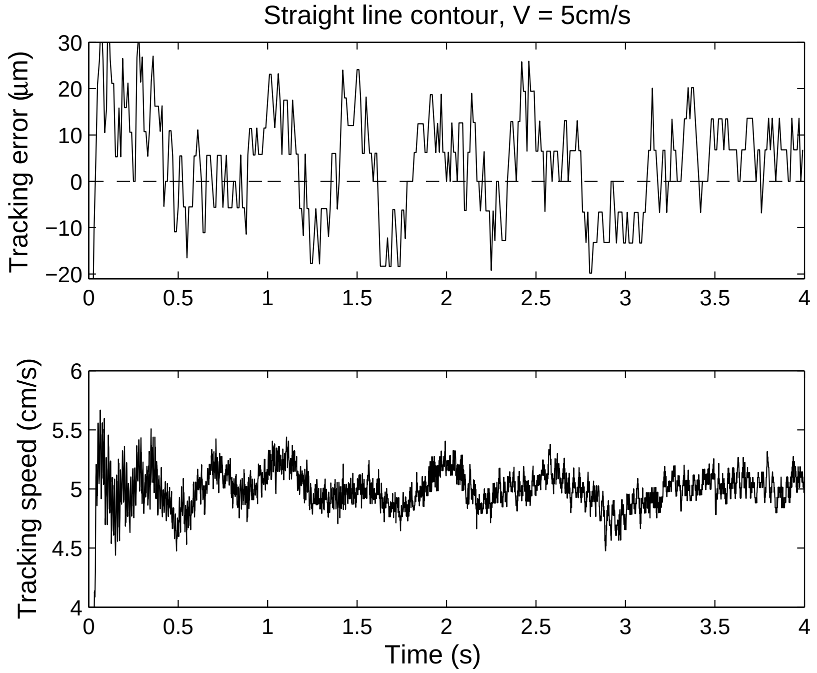

Figure 4.3: Straight line contour tracking error and speed profiles at V = 5 cm/s

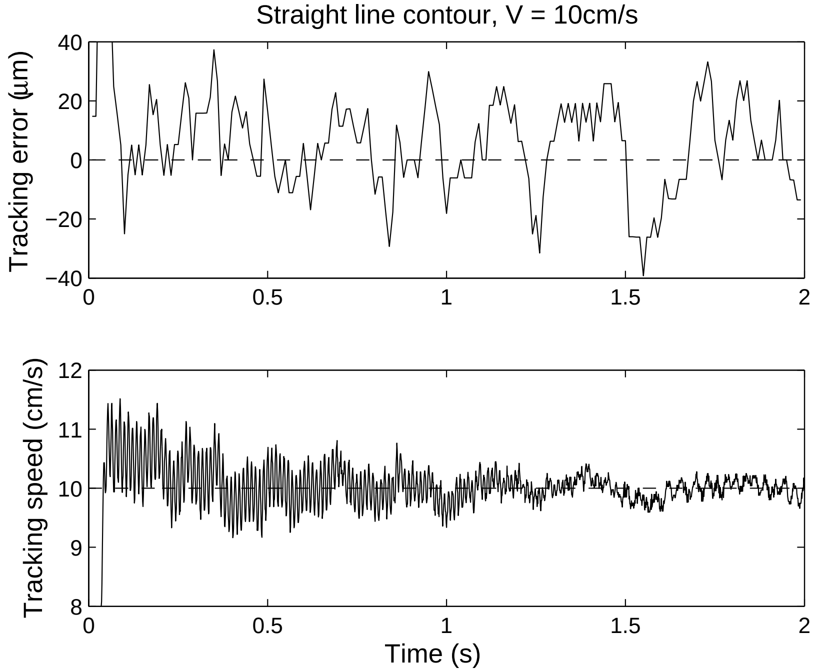

Figure 4.4: Straight line contour tracking error and speed profiles at V = 10 cm/s

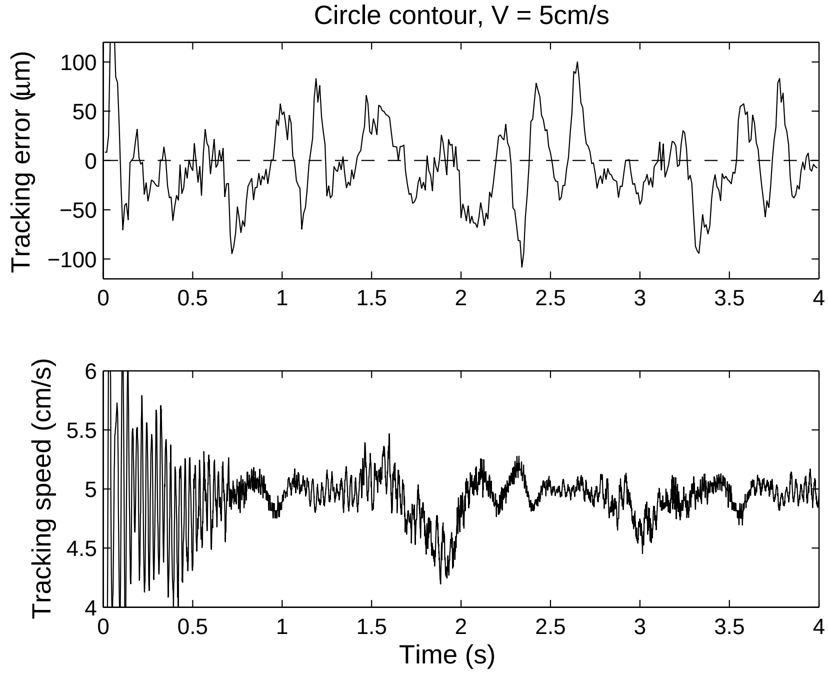

Figure 4.5: Circle contour tracking error and speed profiles at V = 5 cm/s

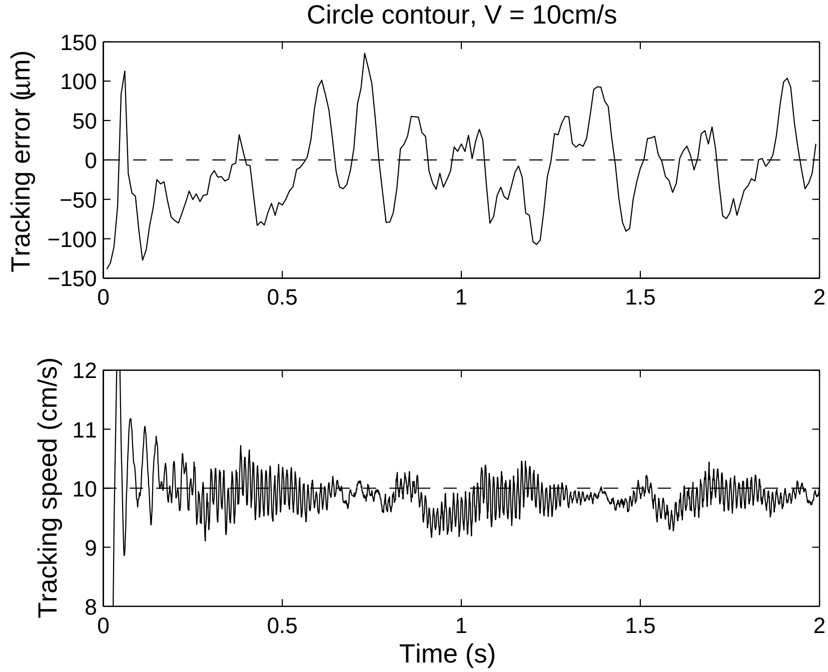

Figure 4.6: Circle contour tracking error and speed profiles at V = 10 cm/s

the expected speed from 0 cm/s to 5 cm/s at the beginning. The speed profile shows that the tracking speed error is within 0.5 cm/s.

If the desired tracking speed is increased to 10 cm/s, the tracking error will increase to 40 μm, as shown in Figure 4.4. However, if we compare the speed profile in Figure 4.3 with that in Figure 4.4, we can see that the steady state speed tracking errors are almost the same.

4.4.2 Circle Tracking

Figure 4.5 shows the tracking error and speed profile of following a circle contour at a speed of 5 cm/s. The radius of the circle contour is 20 mm. From the figure we can see that the tracking error is bounded within 100 μm except at the beginning. We notice that the tracking error for the circle contour is larger than that for the straight line. It is mainly attributed to Coulomb friction of the two link manipulator. That the angular velocities of the two joints both change signs during tracking a circle results in large errors. This is confirmed by the periodic peaks shown in the tracking error profile.

For the same circle contour, increasing desired tracking speed results in larger tracking errors. It can be seen by comparing Figure 4.5 with Figure 4.6. As the same in straight line tracking case, the steady state speed tracking error does not increase as the tracking speed increases.

4.4.3 Rounded Rectangle Tracking

When there is a combination of straight lines and circular arcs in a contour, special attention should be paid in the trajectory generation. Once a new image is available, a decision has to be made to determine whether a curve in the FOV of the camera can be approximated as a straight line or a circular arc. After the decision, the routines for straight line and circular arc trajectory generation can be applied. A simple way to make such a decision is to sample three points (two end points and one middle point) on the curve and check whether they are in a straight line.

A rounded rectangle contour is shown in Figure 4.2. The radius of the rounded corner is 20 mm. The tacking error and the tracking speed profiles are shown in Figure 4.7. The tacking error reaches its maximum when the robot end-effector passes junctions of circular arc and straight line.

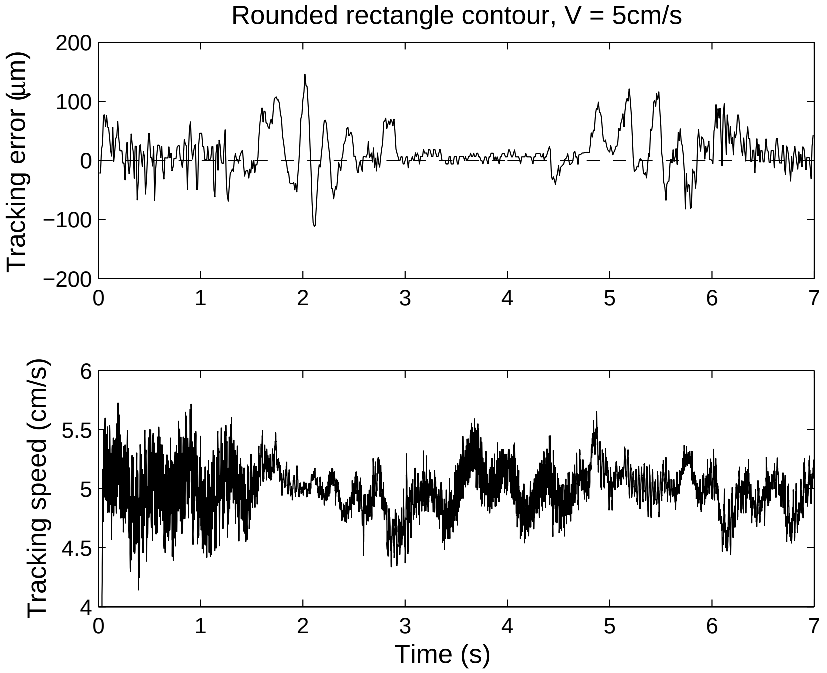

Figure 4.7: Rounded rectangle contour tracking error and speed profiles at V = 5 cm/s

4.5 Summary

The proposed real-time online reference trajectory generation scheme based on the image information for contour following was validated by experiments of following a variety of contours at difference speeds. The contours that were followed included straight lines, circles, and rounded rectangles. The experimental results showed that the two link manipulator can follow different shapes of contours with high accuracy based on the proposed reference trajectory generation scheme.

Part II

Multi-Sensor Based End-effector Sensing and Control

Chapter 5

Kinematic Kalman Filter Design

In this chapter, a one-dimensional kinematic Kalman filter is studied. A naive kinematic Kalman filter design is presented first. Although this filter produces accurate estimates, the closed loop system using the estimates for feedback control is not stable. Stability analysis based on the open loop frequency response is then performed. Finally, a Kalman filter design based on the return difference equality is discussed.

5.1 Introduction

Accurate velocity information is essential for motion control, the significance of which can be seen in proportional-integral-derivative (PID) control systems, model based control systems, adaptive control systems, as well as for friction compensation. Hence, velocity estimation has been a popular research topic for many years. In conventional approaches, velocity is estimated either by numerically differentiating the encoder counts or by a model based approach. For the first approach, the estimation accuracy depends not only on resolution of the encoder, but also on the motion speed. Consequently, it has an unavoidable delay, especially for slow motions. Note that in order to obtain a smooth velocity profile for slow motions, a large amount of encoder counts have to be averaged. As a result, the time delay ends up being half of the time span of the averaged encoder counts. On the other hand, the second approach [36, 37] can provide robust estimation by using a disturbance observer. A disadvantage, however, is that accurate parameters for inertia and friction are required.

As an alternative approach, the Kalman filter may be utilized in the velocity estimation. The idea was first mentioned by Friedland [29]. It is noted that he did not use any accelerometer and his purpose was not for velocity estimation. As MEMS technology made accelerometers more readily available, researchers began to incorporate accelerometers for velocity estimation. It was experimentally verified that the proposed scheme produced smoother velocity estimate with less delay than the estimates obtained by the low pass filtered finite difference approach. Their research, however, was limited to estimation, and they did not use the estimate as feedback signal for closed loop control. This chapter will discuss how to design KKF to ensure closed loop stability of the feedback system with KKF in the loop.

5.2 Kinematic Kalman Filter

5.2.1 Kinematic Model

In a kinematic model, the position measurement is taken as the output while the acceleration measurement is treated as the fictitious input. Based on the position and acceleration measurements, a Kalman filter can be formulated to estimate the velocity. The Kalman filter based on the kinematic model is also known as the kinematic Kalman filter (KKF) [38]. Since the kinematic model does not depend on any unknown system parameters, KKF is immune to modeling errors. Consider the following one-dimensional second order kinematic model:

$$\begin{aligned} \ddot{x}(t) &= a(t) + n_a && (5.1a) \\ y(t) &= x(t) + n_y && (5.1b) \end{aligned}$$

where x and a are the position and the acceleration respectively. n_a and n_y are measurement noises for the position and the acceleration respectively. n_a and n_y are assume to be zero mean Gaussian noises with covariance W_a and V_p, respectively. If zero-order hold is applied to discretize (5.1), $a(t)$ is assumed to be constant in between each sampling period T_s and is denoted by $a(k)$, i.e.

$$a(t) = a(k) \quad \text{for} \quad kT_s \leq t < (k+1)T_s$$

The discrete time state space model for (5.1) is given by

$$\begin{aligned} x_1(k+1) &= x_1(k) + T_s x_2(k) + \frac{1}{2}T_s^2(a(k) + n_a(k)) && (5.2a) \\ x_2(k+1) &= x_2(k) + T_s(a(k) + n_a(k)) && (5.2b) \\ y(k) &= x_1(k) + n_y(k) && (5.2c) \end{aligned}$$

where x_1 is position, x_2 is velocity, and

$$\begin{aligned} n_a(k) &= \frac{1}{T_s}\int_{kT_s}^{(k+1)T_s} n_a(t)dt \\ n_y(k) &= \frac{1}{T_s}\int_{kT_s}^{(k+1)T_s} n_y(t)dt. \end{aligned}$$

Note that $n_a(k)$ and $n_y(k)$ are zero mean Gaussian noises with covariance $W = \frac{1}{T_s}W_a$ and $V = \frac{1}{T_s}V_p$, respectively.

The matrix form of (5.2) can be written as

$$\begin{aligned}
\underbrace{\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix}}_{x(k+1)} &= \underbrace{\begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}}_{x(k)} + \underbrace{\begin{bmatrix} \frac{T_s^2}{2} \\ T_s \end{bmatrix}}_{B} (a(k) + n_a(k)) && \text{(5.3a)} \\
y(k) &= \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{C} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + n_y(k). && \text{(5.3b)}
\end{aligned}$$

In (5.3), the input is the accelerometer measurement and the output is the encoder measurement. There is no model parameter or uncertainty involved in the equation. Therefore, it is robust in the sense of the model uncertainties and disturbances.

5.2.2 Kinematic Kalman Filter Equations

In order to estimate velocity, a standard Kalman filter is applied to the discrete time kinematic model (5.3). It is reasonable to assume that the accelerometer noise and encoder noise are independent white sense stationary Gaussian noises, i.e.

$$\begin{aligned}
E[n_a(k+l)n_a^T(k)] &= W\delta(l) \\
E[n_y(k+l)n_y^T(k)] &= V\delta(l) \\
E[n_a(k+l)n_y^T(k)] &= 0.
\end{aligned}$$

By applying the Kalman filter theory [39], a posteriori state estimator can be written as

$$\begin{aligned}
\hat{x}^o(k+1) &= A\hat{x}(k) + Ba(k) && \text{(5.4a)} \\
\tilde{y}^o(k) &= y(k) - C\hat{x}^o(k) && \text{(5.4b)} \\
F(k) &= M(k)C^T[CM(k)C^T + V]^{-1} && \text{(5.4c)} \\
\hat{x}(k) &= \hat{x}^o(k) + F(k)\tilde{y}^o(k) && \text{(5.4d)}
\end{aligned}$$

where $\hat{x}^o$ is the a-priori state estimate, $\hat{x}$ is the a-posteriori state estimate, $\tilde{y}^o$ is the estimation error of measurement y, F is the Kalman filter gain, and M is the a-priori state estimation error covariance.

The block diagram of (5.4) is shown in Figure 5.1. The dashed box is the kinematic model. Two inputs are acceleration measurements $\bar{a}(k)$ and position measurements $\bar{y}(k)$, two outputs are states of the system. By using the Kalman filter, not only the velocity can be estimated, but also the position measurements can be refined beyond the resolution of position sensors[19, 40].

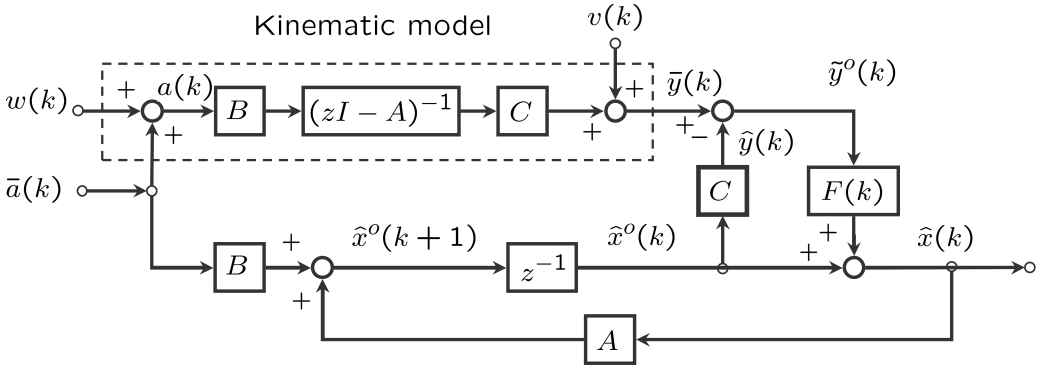

Figure 5.1: Block diagram of kinematic Kalman filter(KKF)

The a-posteriori estimation error covariance matrix $Z(k)$ and a-priori estimation error covariance matrix $M(k)$ are updated by

$$\begin{aligned} Z(k) &= M(k) - F(k)CM(k) && (5.5a)\\ M(k) &= AZ(k)A^T + BWB^T. && (5.5b)\end{aligned}$$

(5.5) are only utilized for calculating the Kalman filter gains. If the pair $[A, C]$ is observable (or detectable), $M(k)$ will converge to a constant matrix M after several iterations. If that is the case, M can be directly obtained as the positive solution of the following Riccati equation:

$$AMA^T - M = -BWB^T + AMC^T[CMC^T + V]^{-1}CMA^T. \quad (5.6)$$

Then, $F(k)$ will be a constant gain, denoted as F. The Kalman filter with a constant gain is called the steady state Kalman filter.

5.3 Kinematic Kalman Filter Design

From (5.4) we know that the implementation of Kalman filter requires the a-priori information of $\hat{x}(0)$, $M(0)$, the exact knowledge of the process noise covariance W and the measurement noise covariance V. Since $\hat{x}(0)$ and $M(0)$ only affect first several estimates, the values of them can be roughly approximated. However, it is crucial to obtain accurate values for W and V since they affect the Kalman filter gains, and the estimation results. These values are usually difficult to identify. Therefore identification of W and V is often done by trial and error [41].

In Section 5.3.1, the single joint setup is controlled by a PID controller to follow a reference trajectory for different W and V. The estimation results for different W and V will be compared and discussed. In Section 5.3.2, an explanation of the conclusion drawn in the previous section will be given.

5.3.1 Estimation Performance for Different W and V

In this section, the single joint setup will be controlled by a simple PID controller, as shown in Figure 5.2.

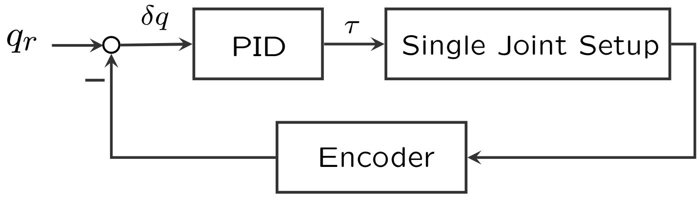

Figure 5.2: Block diagram of encoder feedback system for single joint setup

Feedback signal is from the encoder. The angular position is directly measured with encoder counts while the angular velocity is obtained by one step backward difference of the angular positions, i.e.

$$\omega(k) = \frac{q(k) - q(k-1)}{T_s} \tag{5.7}$$

where ω is the angular velocity, q is the angular position and T_s is the sampling period.

The PID control law is given by

$$\begin{aligned} \tau(k) &= K_p(q_r(k) - q(k)) \\ &\quad + K_i \sum_{i=1}^{k} (q_r(i) - q(i)) + K_d(\omega_r(k) - \omega(k)) \end{aligned} \tag{5.8}$$

where τ is the torque command, q_r and w_r are the reference joint angle and angular velocity respectively. q and ω are measured joint angle and angular velocity respectively. $K_p = 360$, $K_i = 1$ and $K_d = 10$ are the proportional, integral and derivative gains respectively.

The single joint setup for the experiment is introduced in Section 2.2. A dual axis MEMS accelerometer (ADXL202E) manufactured by Analog Devices Inc. is used to measure the acceleration. The parameters of the accelerometer are shown in Table 5.1.

Table 5.1: Parameters of ADXL202E accelerometer

Specifications	Values	Units
Range	±2	g
Noise Variance	0.029	$(\mathrm{m/s^2})^2$
3dB Bandwidth	6	kHz

The two-dimensional accelerometer is mounted on the center of the end-effector. This allows the accelerometer to measure acceleration in the tangent direction(y axis) and radial direction(x axis). However, only the y axis measurements are used to calculate the angular acceleration by

$$\bar{a} = \bar{a}_y / L \tag{5.9}$$

where L is the link length of the single joint setup.

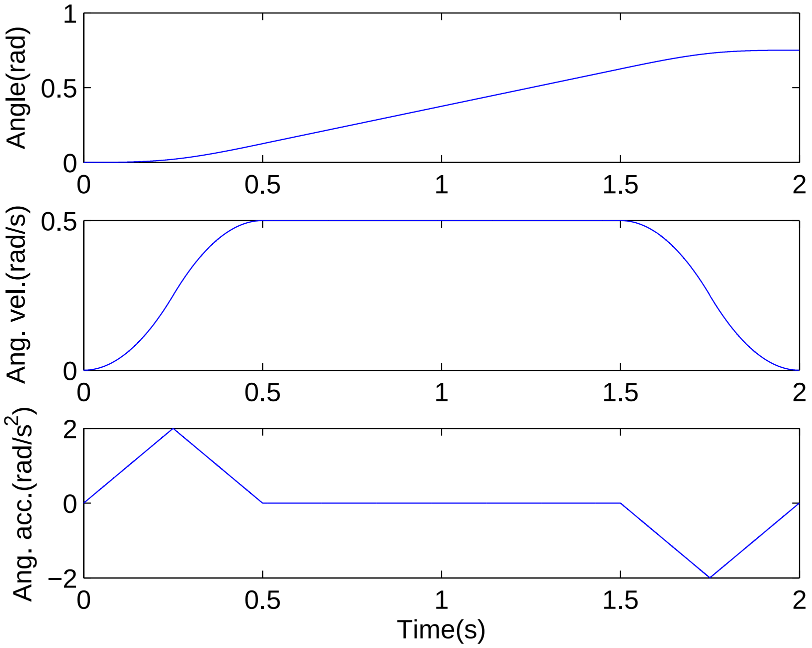

Figure 5.3: Reference trajectory for single joint setup

The single joint setup is controlled to trace a circular arc within a 2 second period. The reference trajectory for the circular arc in the joint space is shown in Figure 5.3.

The total angle of the circular arc is 0.7506 rad, which is about 43 degree. The radius of the circular arc is the same as the link length L. It has a smooth velocity profile with maximum speed 0.5 rad/s and a continuous acceleration profile with maximum acceleration 2 rad/s^2.

As is well known, the gain matrix of a Kalman filter for single-input, single-output systems depends on the ratio between the covariance of the process noise and that of the measurement noise, W/V. The experimental result in Figure 5.4 was obtained for $W/V = 1E8$. Note that the experiment is intended to test the open loop estimation performance of the Kalman filter, and the estimate is not used for feedback control. Blue lines are measurements from the encoder, red lines are the reference trajectory, green and cyan lines are Kalman filter estimates. The measured angular velocities are obtained by (5.7), so it is very noisy. In order to see the estimation more clearly, Figure 5.4 plots a short segment of the estimation process. Figure 5.4 shows that the Kalman filter estimate tracks the real measurement very closely. What is more, the estimated angular velocities are much smoother than the encoder measurements, which is a benefit of the kinematic Kalman filter. Although there are two Kalman filter estimates, the green and cyan lines overlap in Figure 5.4.

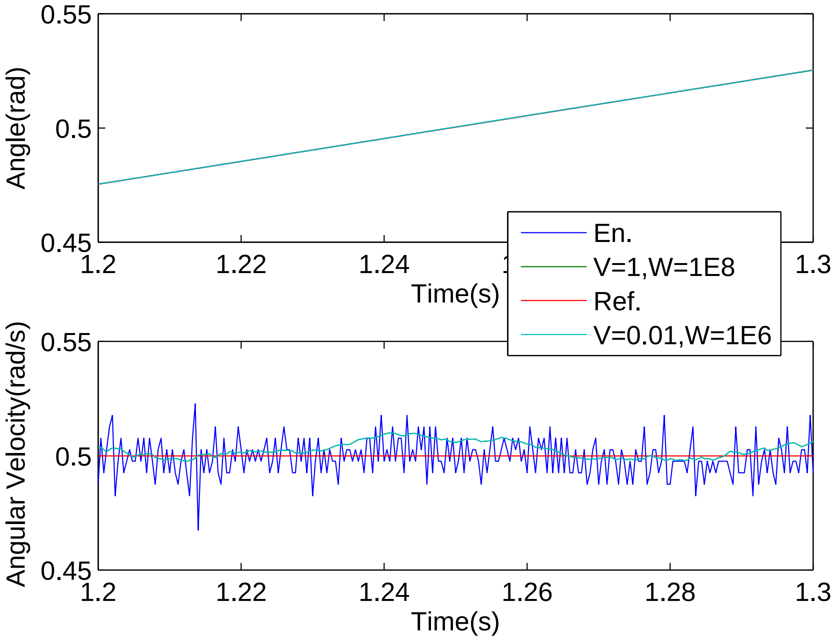

Figure 5.4: KKF estimations for the same W/V ratio

Figure 5.5 shows the KKF performance for different W/V ratios. The green line is the KKF estimate for $W/V = 1E6$, while the cyan line is the KKF estimate for $W/V = 1E8$. From Figure 5.5 we can see that the KKF performance for different

W/V ratios are different. Apparently, the estimation error for $W/V = 1E8$ is smaller than that for $W/V = 1E6$.

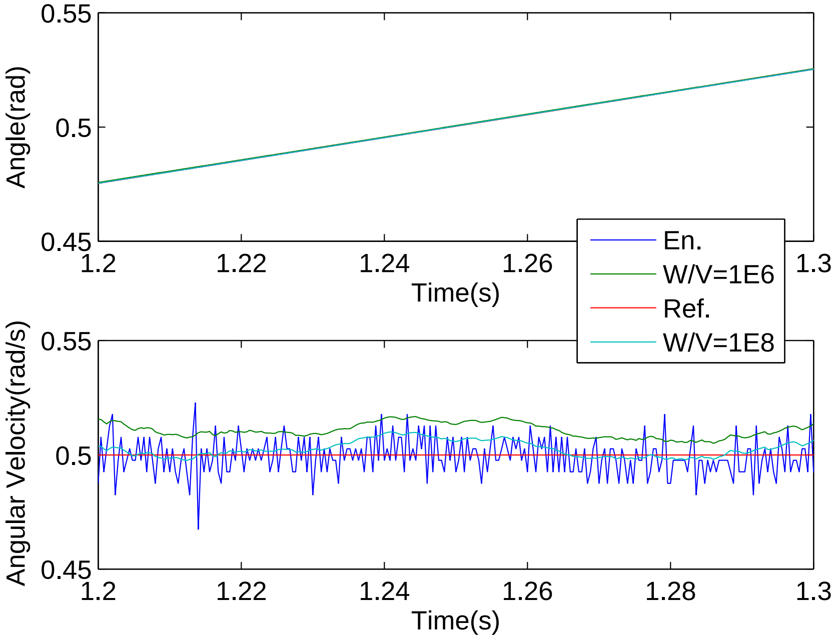

Figure 5.5: KKF estimations for different W/V ratios

The above two experiments shows the steady state KKF performance is only affected by the W/V ratio; as is well known it is irrelevant to specific values of W and V. We can see this point more clearly if the Kalman filter gains are calculated. As long as the W/V ratio are the same, the steady state Kalman filter gain is always the same. Different W/V ratios will result in different Kalman filter gains. In the following section, an explanation will be given.

5.3.2 Relations Between KKF Gain and W/V

In this section, we make use of the return difference equality for Kalman filter to show that the steady state Kalman filter gain depends only on the ratio of W and V [39].

Figure 5.6 shows block diagram of Kalman filter as a whitening filter. Notations follow Section 5.2, while $\Phi(z) = (zI - A)^{-1}$ and $L = AF$. The plant turns white noise to colored noise, while Kalman filter turns colored noise back to the white noise.

The return difference equality for the Kalman filter is given by

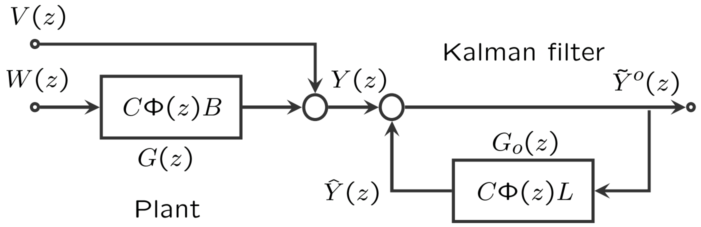

Figure 5.6: Block diagram of plant and Kalman filter

$$[1 + G_o(z)][1 + G_o(z^{-1})] = \gamma \left[1 + \frac{W}{V} G(z)G(z^{-1})\right] \tag{5.10}$$

where

$$\gamma = \frac{V}{CMC^T + V}.$$

Assume $Q = CMC^T + V$ and arrange (5.10), we can get

$$[1 + G_o(z)]Q[1 + G_o(z^{-1})] = V + G(z)WG(z^{-1}). \tag{5.11}$$

The return difference equality for the discrete time regulator is derived in Anderson and Moore [42]. The return difference equality for the Kalman filter can be derived in a similar manner. The proof in Appendix A actually verifies the return difference equality (5.11).

Based on the return difference equality (5.10), root locus of the closed loop Kalman filter can be drawn. It can be seen that the closed loop poles of the steady state Kalman filter, which are eigenvalues of $A - LC$, depend only on W/V. The single joint setup is an observable single output system. For such systems, the estimator gain and the closed loop eigenvalues are uniquely related. Therefore, the steady state Kalman filter gain L depends only on W/V.

5.3.3 Proper W/V Identification

In the previous section we showed that the steady state Kalman filter gain only depends on W/V. However, the proper value of W/V still has to be identified for estimation purposes.

One index to evaluate the goodness of a certain W/V is the estimation error. From Figure 5.5 we can see that the estimation error for $W/V = 1E6$ is larger than that for $W/V = 1E8$. This is because the resolution of encoder is very high while the measurement of acceleration is very noisy. High W/V value means that we trust the encoder measurement more than the acceleration measurement. As W/V value is increased, however, the KKF position estimate becomes closer to encoder measurement, which means that the estimated velocity profile becomes more jagged. Since the real velocity profile should be smooth, having the KKF output close to the encoder values does not necessarily mean that the estimated values are accurate. The accurate velocity estimate should be a smooth profile at about the center line of the encoder measurement profile.

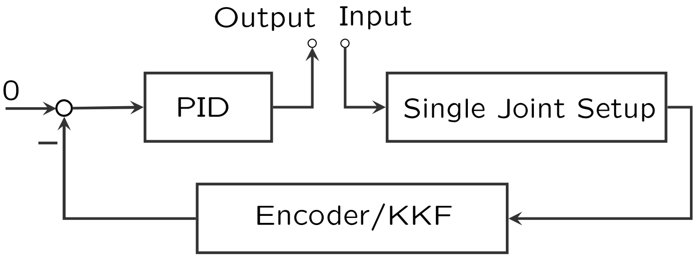

Figure 5.7: Block diagram of open loop frequency response measurement

The other index to evaluate the desirability of a certain W/V is the closed loop stability of the system when KKF estimates are used for feedback control. The robustness of the KKF feedback system varies as W/V value changes. In order to to see this clearly, we obtained open loop frequency responses with different W/V values. As shown in Figure 5.7, the sine-by-sine torque input was injected to the single joint setup and the controller output was measured. In the experiment, the frequency range for the test was 0.5 Hz - 150 Hz. The frequency responses of the encoder feedback system and the KKF feedback system with different W/V values are shown in Figure 5.8.

In Figure 5.8, blue line indicates the frequency response for encoder feedback control, all other color lines indicate the KKF feedback control with different W/V values. In order to see the robustness of KKF feedback system with different W/V values, phase margins were calculated as shown in Table 5.2.

From Table 5.2 we can see that the robustness of the KKF feedback system decreases as W/V value increases. However, low W/V value indicates large estimation errors. Therefore there is a tradeoff between estimation error and system robustness. We can also see that by properly choosing the W/V value, the KKF feedback system may be more robust than the encoder feedback system.

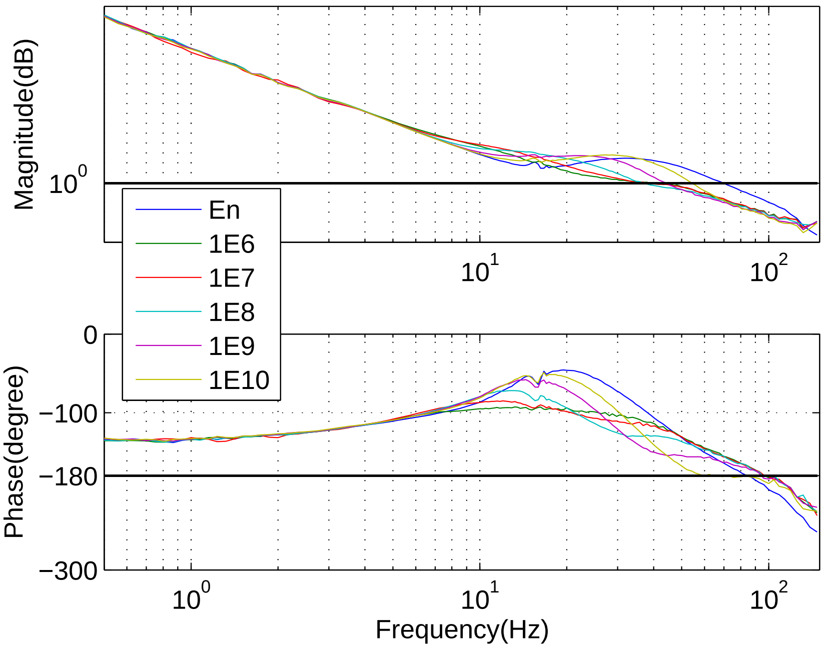

Figure 5.8: Open loop frequency responses for different W/V ratios

Table 5.2: Phase margins for encoder and KKF feedback control

W/V	Gain Crossover Frequency(Hz)	Phase Margins(degree)
Encoder	69	16.5
W/V=1E6	32	74.6
W/V=1E7	38	65.7
W/V=1E8	37	49.8
W/V=1E9	44	25.8
W/V=1E10	54	5.5

5.4 Summary

In this chapter, a kinematic model based Kalman filter was introduced. Based on the return difference equality, it was shown that the steady state Kalman filter gain depends only on the ratio of the process noise covariance and the measurement noise covariance. The ratio must be selected by examining the estimation error and the closed loop stability robustness of the feedback loop, where the feedback controller utilizes the KKF state estimate.

Chapter 6

Kinematic Kalman Filter Based Sensing

In this chapter, kinematic Kalman filters with multiple sampling rates and large measurement delay are discussed. More specifically, a lifting technique is utilized to handle the multi-rate issue while the state augmentation technique is utilized to handle the delay issue. These techniques are experimentally verified for one-dimensional case and two-dimensional case respectively.

6.1 Introduction

Mounting sensors on the end-effector to improve estimation performance was first proposed by Sweet and Good in 1985 [17]. Although this approach has not yet been widely adapted in industry, a considerable amount of research has been carried out in this area. A vision sensor is often considered as a suitable choice for this purpose. Its low sampling rate and high latency, however, makes it very difficult to directly incorporate it into a control loop, which usually runs at a much higher sampling rate. Although high speed vision systems exist, they are too expensive for industrial applications. Alternatively, inexpensive sensors such as accelerometers and gyros can be utilized to improve the load side estimation performance. However, these sensors measurements are usually contaminated by noise. Hence a noise filter is essential for utilizing these sensors.

The Kinematic Kalman Filter (KKF) is one of the well developed end-effector sensing techniques. The main idea of KKF is applying the Kalman filter to a kinematic model with practical acceleration measurements. It was first applied to a one-dimensional linear motor, and the position sensor is an encoder [19]. This idea was further extended by Jeon et al. to a three-dimensional case with a vision sensor as the position measurement [20]. The algorithm in [20], however, is complicated for real-time implementation.

In this chapter, we will formulate a KKF with the position signal from a vision sensor. Since the vision sensor has a low sampling rate and large measurement delay, we will propose a simple yet effective scheme to address multi-rate and measurement delay issues. More specifically, for multi-rate issue, the lifting technique is applied, and for measurement delay issue, the state augmentation technique is utilized. Two techniques will be addressed for one-dimensional and two-dimensional cases respectively.

6.2 One-dimensional Estimation

Consider the one-dimensional discrete time kinematic model introduced in Section 5.2.

$$\underbrace{\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix}}_{x(k+1)} = \underbrace{\begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}}_{x(k)} + \underbrace{\begin{bmatrix} \frac{T_s^2}{2} \\ T_s \end{bmatrix}}_{B} (a(k) + n_a(k)) \tag{6.1a}$$

$$y(k) = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{C} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + n_y(k) \tag{6.1b}$$

where x_1 and x_2 are the position and the velocity respectively, a is the acceleration, and T_s is the sampling period. n_a and n_y are noises for acceleration and position measurements, respectively. They are assumed to be zero mean Gaussian noises with covariance W and V respectively.

6.2.1 Lifting Technique

Different sensors and actuators may have different sampling rates. When these devices are combined together, a multi-rate state estimation may provide better results than a single-rate state estimation[18]. Consider the case where a slow vision sensor is used for position measurement while a fast accelerometer for acceleration measurement in a kinematic model. Suppose that the sampling period of the vision sensor is N times longer than that of the accelerometer. We define the time index for the vision sensor sampling time as m. m is related to the time index for the accelerometer k by $k = mN$. To express the overall system at the low sampling rate, we apply the lifting technique to (6.1), we can get

$$\begin{aligned} x\left((m+1)N\right) &= A^N x(mN) \\ &+ \left[\begin{array}{cccc} A^{N-1}B & \cdots & AB & B \end{array}\right] \left(U_a(mN) + N_a(mN)\right) \quad (6.2a) \\ y(mN) &= Cx(mN) + n_y(mN) \quad (6.2b) \end{aligned}$$

where

$$U_a(mN) = \left[\begin{array}{c} a(mN) \\ a(mN+1) \\ \vdots \\ a(mN+N-1) \end{array}\right], \quad N_a(mN) = \left[\begin{array}{c} n_a(mN) \\ n_a(mN+1) \\ \vdots \\ n_a(mN+N-1) \end{array}\right]. \quad (6.3)$$

The covariance of measurement noise n_y is V while the covariance of process noise N_a is $W \cdot I^{N\times N}$. (6.2) shows that the lifting technique is the same as writing a serial set of state space equations into a compact form. This form is computationally efficient and is also convenient for applying the Kalman filter. See [43] for the detailed discussion of the lifting technique.

For a multi-rate system, it is always true that the system can be run in single rate by downsampling the fast device. By doing this, however, some information provided by the fast device is lost. The lifting technique is designed for taking full advantage of the fast device.

Figure 6.1 shows that the KKF utilizing the lifted model produce better results than that utilizing the downsampled model. The experiment was performed for $T_s = 0.4$ ms, $N = 20$, $W/V = 1E8$, and the setup is the same as in Section 5.3.1. Green lines show the KKF estimates from (6.2). For intersample estimation for i's at $mN \leqslant i < (m+1)N$, simple open loop integration of accelerometer measurements is applied, i.e.

$$\hat{x}(mN+i+1) = A\hat{x}(mN+i) + B\bar{a}(mN+i). \quad (6.4)$$

Cyan lines show the KKF estimates for model (6.1) where $T_s = N \times 0.4$ ms $= 8$ ms.

As shown in the lower subplot of Figure 6.1, velocity estimate from the lifted model (green line) has less error than that from the downsampled model (cyan line). However it is hard to tell the position estimation error from the upper subplot of Figure 6.1. In order to see the position estimation error more clearly, a small time range of Figure 6.1 is shown in Figure 6.2. From Figure 6.2 we can see that there is a constant position estimation error for the KKF estimate with the downsampled model. The KKF position estimation with the lifted model does not have such error. Therefore, the lifting technique can significantly improve the KKF estimation performance.

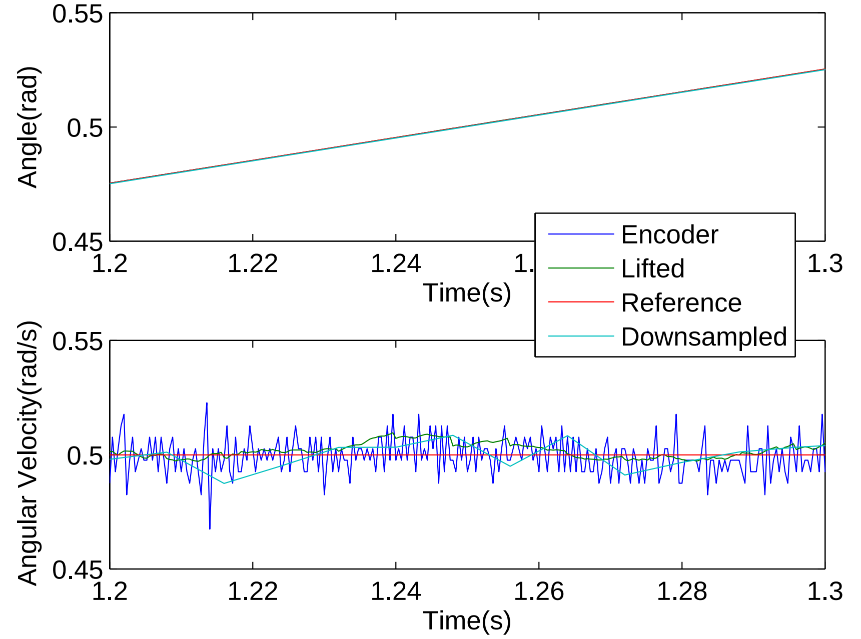

Figure 6.1: KKF estimates for lifted and downsampled model

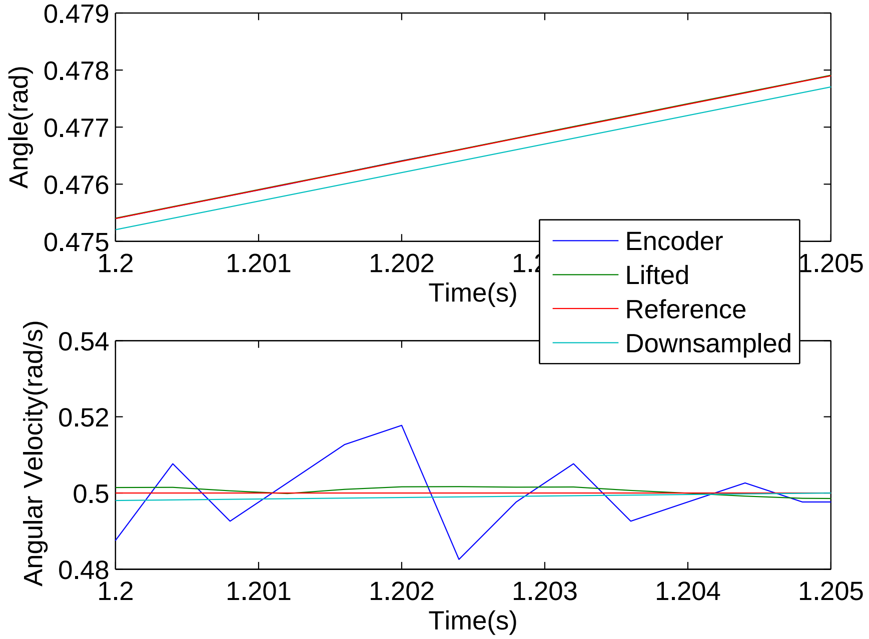

Figure 6.2: KKF estimates for lifted and downsampled model (zoom in)

6.2.2 State Augmentation Technique

There are a variety of techniques to treat measurement delay in control systems. These methods include Dahlin's Algorithm [44] and Smith Predictor Algorithm [45]. Most of these methods, however, highly depend on the plant model. Modeling error for the plant can cause the performance of these algorithms to deteriorate. A simple yet effective method which does not depend on modeling errors is the state augmentation technique. The main idea of this technique is to add additional states to the original system such that the states of the previous step are preserved for the next step. This method is effective as long as the delay time is smaller than one sampling period.

Let us take the system (6.1) as an example. If the measurement has one step delay, to apply state augmentation technique, an additional state $x_3(k+1)$ has to be added to the original system to preserve the state of the previous step $x_1(k)$. The output is the new state $x_3(k+1)$. The modified system is

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \end{bmatrix} = \begin{bmatrix} 1 & T_s & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} + \begin{bmatrix} \frac{T_s^2}{2} \\ T_s \\ 0 \end{bmatrix} (a(k) + n_a(k)) \quad (6.5a)$$

$$y(k) = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} + n_y(k). \quad (6.5b)$$

Figure 6.3 shows that KKF estimates over the augmented model (6.5) is superior to the estimates over the delayed model (6.6). The experiment was performed for $T_s = 0.4$ ms, $W/V = 1E8$ and the setup is the same as in Section (5.3.1). In order to see the difference clearly, only estimated tracking errors are plotted. Blue lines show the KKF estimated tracking error using model (6.5) where delay is handled by the state augmentation technique. Green lines show the KKF estimated tracking error using model (6.1) which does not have any measurement delay. Cyan lines show the KKF estimated tracking error using model (6.6) which has one step measurement delay.

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} \frac{T_s^2}{2} \\ T_s \end{bmatrix} (a(k) + n_a(k)) \quad (6.6a)$$

$$y(k) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1(k-1) \\ x_2(k-1) \end{bmatrix} + n_y(k). \quad (6.6b)$$

As shown in Figure 6.3, KKF estimation performance with the augmented model is almost the same as that with the model which does not have any measurement delay. However, KKF estimation performance with model (6.6) is worse than that with the augmented model, especially for position estimation. This justifies the state augmentation technique.

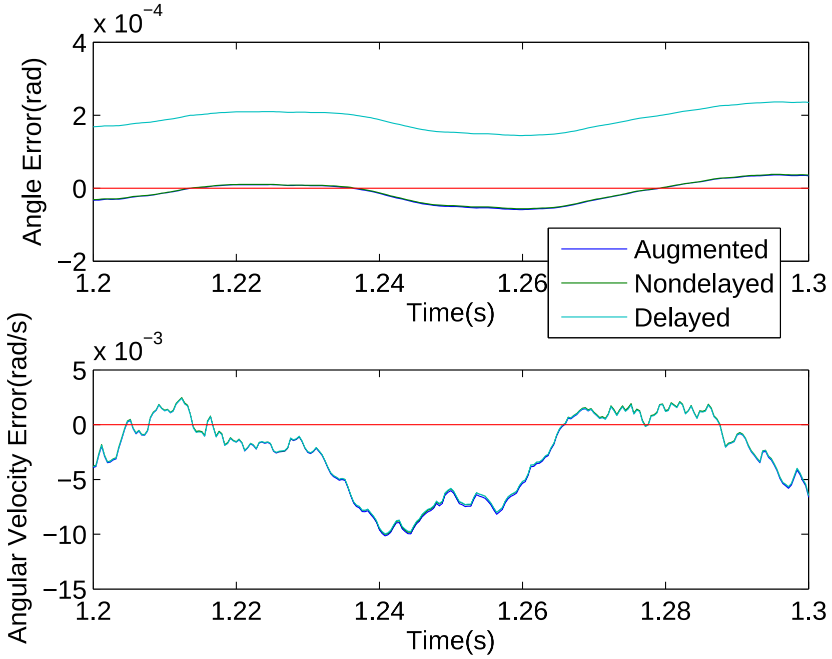

Figure 6.3: KKF estimated tracking error for augmented model

For the system that has an N-step delay, the lifting technique has to be applied first before applying state augmentation technique. Taking system (6.2) as an example, the augmented system is given by

$$\begin{aligned} \begin{bmatrix} x((m+1)N) \\ z((m+1)N) \end{bmatrix} &= \begin{bmatrix} A^N & 0_{2\times1} \\ \begin{bmatrix} 1 & 0 \end{bmatrix} & 0 \end{bmatrix} \begin{bmatrix} x(mN) \\ z(mN) \end{bmatrix} \\ &+ \begin{bmatrix} A^{N-1}B & \cdots & AB & B \\ & 0_{1\times N} & & \end{bmatrix} (U_a(mN)+N_a(mN)) \quad (6.7a) \\ y(mN) &= \begin{bmatrix} 0_{1\times2} & 1 \end{bmatrix} \begin{bmatrix} x(mN) \\ z(mN) \end{bmatrix} + n_y(mN) \quad (6.7b) \end{aligned}$$

where $z(m+1)N$ is the added state to preserve $x_1(mN)$ which is the state of N steps before, and $U_a(mN)$, $N_a(mN)$ are given by (6.3).

6.2.3 One-dimensional Multi-rate KKF with Large Measurement Delay

If N is dropped for convenience, (6.7) can be written as

$$\begin{aligned} x_e(m+1) &= A_e(m) + B_e(U_a(m) + N_a(m)) & (6.8a) \\ y_e(m) &= C_e(m) + N_y(m) & (6.8b) \end{aligned}$$

where

$$\begin{aligned} A_e &= \begin{bmatrix} A^N & 0_{2\times 1} \\ \begin{bmatrix} 1 & 0 \end{bmatrix} & 0 \end{bmatrix}, \\ B_e &= \begin{bmatrix} A^{N-1}B & \cdots & AB & B \\ & 0_{1\times N} & & \end{bmatrix}, \\ C_e &= \begin{bmatrix} 0_{1\times 2} & 1 \end{bmatrix}. \end{aligned}$$

$N_a(m)$ and $N_y(m) = n_y(mN)$ are accelerometer and vision sensor noises. It is reasonable to assume that they are independent white noise with covariance $W_e(m)$ and $V_e(m)$, respectively. As a result, $N_a(m)$ and $N_y(m)$ satisfy

$$\begin{aligned} E\{N_a(m+l)N_a^T(m)\} &= W_e(m)\delta(l) \\ E\{N_y(m+l)N_y(m)\} &= V_e(m)\delta(l) \\ E\{N_a(m+l)N_y(m)\} &= 0. \end{aligned}$$

By applying the Kalman filter theory, a-posteriori state estimator can be written as

$$\begin{aligned} \hat{x}_e^o(m+1) &= A_e\hat{x}_e(m) + B_eU_a(m) & (6.9a) \\ \tilde{y}_e^o(m) &= y_e(m) - C_e\hat{x}_e^o(m) & (6.9b) \\ F(m) &= M_e(m)C_e^T[C_eM_e(m)C_e^T + V_e(m)]^{-1} & (6.9c) \\ \hat{x}_e(m) &= \hat{x}_e^o(m) + F(m)\tilde{y}_e^o(m). & (6.9d) \end{aligned}$$

A-posteriori estimation error covariance matrix $Z_e(m)$ and a-priori estimation error covariance matrix $M_e(m)$ are updated by

$$\begin{aligned} Z_e(m) &= M_e(m) - F(m)C_eM_e(m) & (6.10a) \\ M_e(m) &= A_eZ_e(m)A_e^T + B_eW_e(m)B_e^T. & (6.10b) \end{aligned}$$

(6.8 - 6.10) run every N steps to correct the current estimate by the delayed position measurement obtained N steps before. For the intersample estimation, a simple open loop integration of accelerometer measurements is employed. In order to evaluate the performance of the proposed sensor fusion algorithm, it is necessary to implement it to a practical system.

6.2.4 Experimental Results

The experimental setup utilized to evaluate the estimation scheme is described in Section 2.2. The accelerometer is described in Section 5.3.1. The National Instruments PCI-7831R data acquisition board was used to collect accelerometer and encoder measurements and also to send control commands to the single joint setup. The real-time system is National Instruments LabVIEW ETS real-time module 8.5.1.

Since a vision sensor functions like a position sensor with a low sampling rate and large time delay, it is reasonable to mimic the vision sensor measurements with modified encoder measurements. This is the approach taken in this experiment. In order to resemble the vision sensor measurements, the encoder measurements have to be delayed and its sampling rate has to be reduced. For the experiments, the sampling rates of the controller and the accelerometer are set to 2.5 kHz, while that of the encoder measurement is set to 125 Hz. The reference trajectory and the control law are the same as in Section 5.3.1.

Figure 6.4 shows the estimation performance. The red line is the reference trajectory, the blue line is the encoder measurement, and the green line is the KKF estimates. Figure 6.5 shows the tracking error profile, the blue line is the real tracking error from the encoder, while the green line is the estimated tracking error from KKF. It shows that the estimated tracking error is very close to the real tracking error. Experimental results showed the effectiveness of the proposed multi-rate KKF with large measurement delay scheme for the one-dimensional case.

6.3 Two-dimensional Estimation

Consider the two link direct drive manipulator mentioned in Section 2.3. The schematic is shown in Figure 6.6. $X-Y$ is task space coordinate system, and $x-y$ is the end-effector coordinate system associated with the accelerometer and the camera. The rectangle on the end-effector represents the camera. Let (p_X, p_Y) be the center of the camera in the task plane. The accelerometer is mounted on the center of the camera. q_1 and q_2 are joint angles of the first and the second link respectively.

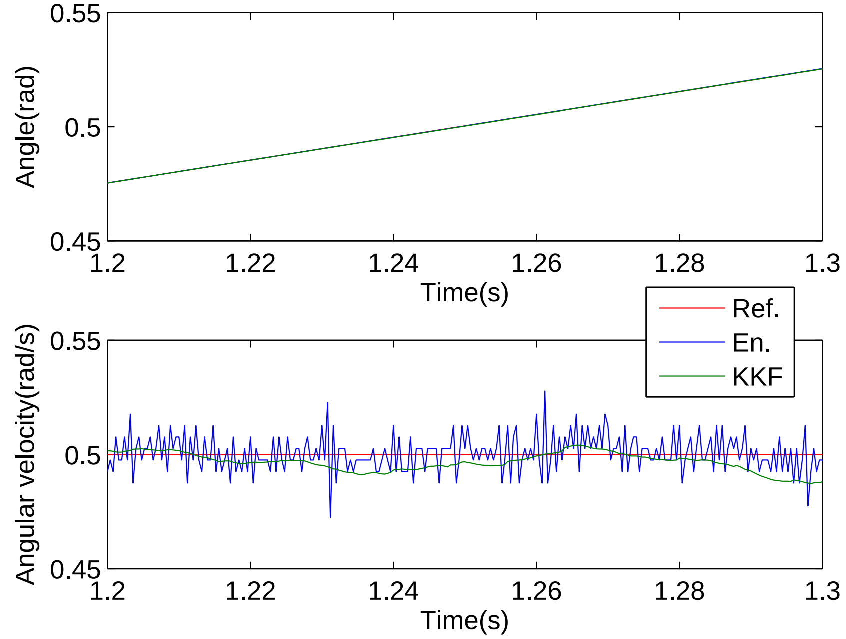

Figure 6.4: KKF estimates of angle and angular velocity

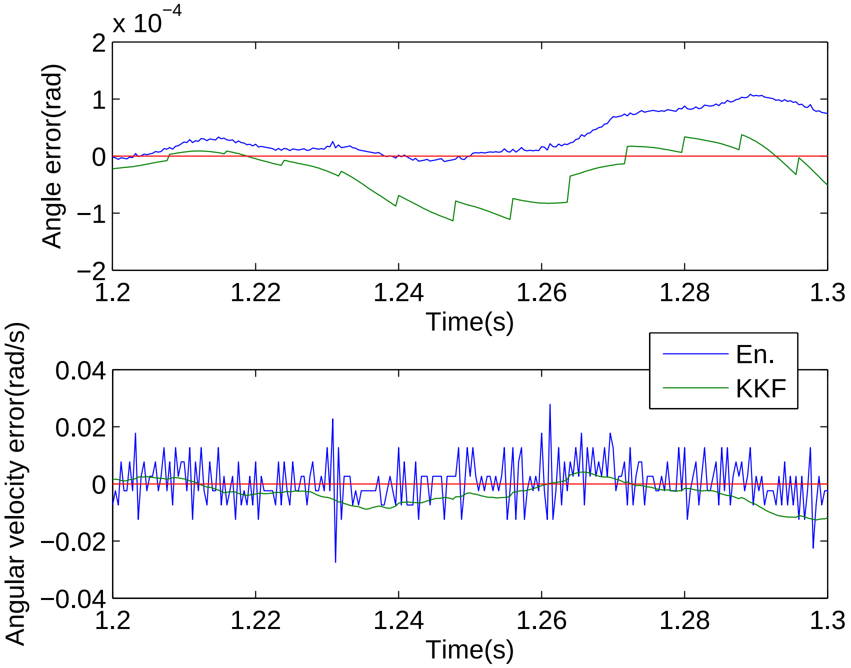

Figure 6.5: Estimated tracking error for angle and angular velocity

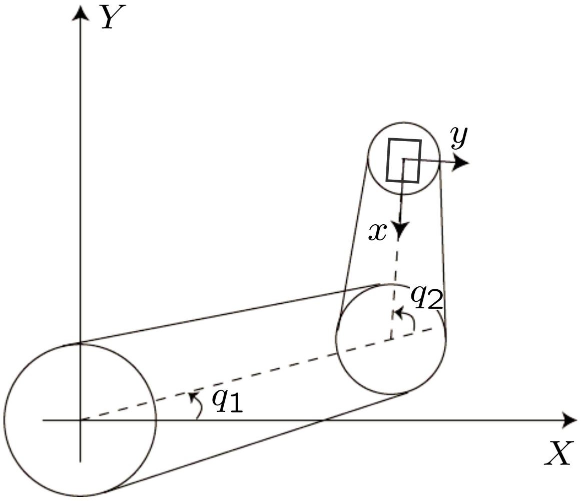

Figure 6.6: Schematic of two link manipulator (overview)

The accelerometer measurements are in end-effector coordinate system. Thus, it is necessary to convert them to the task space coordinate system for estimation purpose. Suppose the measurements of acceleration in x and y axis are a_x and a_y respectively. Then, the acceleration in task space a_X and a_Y can be obtained by

$$\begin{bmatrix} a_X \\ a_Y \end{bmatrix} = \underbrace{\begin{bmatrix} -\cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & -\cos(\alpha) \end{bmatrix}}_{R(\alpha)} \begin{bmatrix} a_x \\ a_y \end{bmatrix} \tag{6.11}$$

where $\alpha = q_1 + q_2$.

The end-effector position, which is the same as the center of the camera, is obtained by the following forward kinematics:

$$\begin{bmatrix} p_X \\ p_Y \end{bmatrix} = \begin{bmatrix} l_1 \cos(q_1) + l_2 \cos(\alpha) \\ l_1 \sin(q_1) + l_2 \sin(\alpha) \end{bmatrix} \tag{6.12}$$

where l_1 and l_2 are the lengths of the first link and the second link respectively.

Two-dimensional discrete time kinematic model is

$$\begin{aligned}
\underbrace{\begin{bmatrix} p_X(k+1) \\ p_Y(k+1) \\ v_X(k+1) \\ v_Y(k+1) \end{bmatrix}}_{x(k+1)} &= \underbrace{\begin{bmatrix} 1 & 0 & T_s & 0 \\ 0 & 1 & 0 & T_s \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} p_X(k) \\ p_Y(k) \\ v_X(k) \\ v_Y(k) \end{bmatrix}}_{x(k)} \\
&+ \underbrace{\begin{bmatrix} \frac{T_s^2}{2} & 0 \\ 0 & \frac{T_s^2}{2} \\ T_s & 0 \\ 0 & T_s \end{bmatrix}}_{B} (a(k) + n_a(k)) \qquad (6.13a) \\
y(k) &= \underbrace{\begin{bmatrix} I_{2\times2} & 0_{2\times2} \end{bmatrix}}_{C} x(k) + n_p(k) \qquad (6.13b)
\end{aligned}$$

where

$$a(k) = \begin{bmatrix} a_X(k) \\ a_Y(k) \end{bmatrix}, \quad n_a(k) = \begin{bmatrix} n_{aX}(k) \\ n_{aY}(k) \end{bmatrix}, \quad n_p(k) = \begin{bmatrix} n_{pX}(k) \\ n_{pY}(k) \end{bmatrix}.$$

T_s is the sampling period, and v_X, v_Y are the end-effector velocity in X axis and Y axis respectively. n_a is the acceleration measurement noise vector and n_p is the camera measurement noise vector. n_a and n_p are white sense stationary Gaussian noises with covariance matrix $W \cdot I_{2\times2}$ and $V \cdot I_{2\times2}$ respectively, where W and V are scalars.

6.3.1 Two-dimensional Multi-rate KKF with Large Measurement Delay

Suppose that the sampling rate of the accelerometer is N times faster than that of the vision sensor and that the measurement delay of the vision sensor is N times the sampling period of the accelerometer. By applying lifting and state augmentation techniques to the model (6.13), we obtain

$$\begin{aligned}
\begin{bmatrix} x((m+1)N) \\ z((m+1)N) \end{bmatrix} &= \begin{bmatrix} A^N & 0_{4\times2} \\ \begin{bmatrix} I_{2\times2} & 0_{2\times2} \end{bmatrix} & 0_{2\times2} \end{bmatrix} \begin{bmatrix} x(mN) \\ z(mN) \end{bmatrix} \qquad (6.14a) \\
&+ \begin{bmatrix} A^{N-1}B & \cdots & AB & B \\ & 0_{2\times2N} & & \end{bmatrix} (U_a(mN) + N_a(mN)) \\
y(mN) &= \begin{bmatrix} 0_{2\times4} & I_{2\times2} \end{bmatrix} \begin{bmatrix} x(mN) \\ z(mN) \end{bmatrix} + n_p(mN) \qquad (6.14b)
\end{aligned}$$

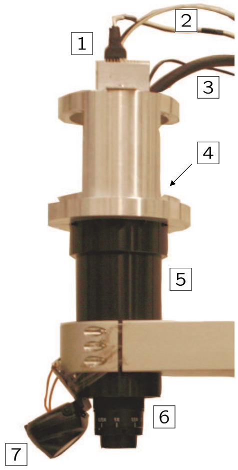

Figure 6.7: Sensors mounted on the end-effector of the two link manipulator

where

$$U_a(mN) = \begin{bmatrix} a(mN) \\ a(mN+1) \\ \vdots \\ a(mN+N-1) \end{bmatrix}, \quad N_a(mN) = \begin{bmatrix} n_a(mN) \\ n_a(mN+1) \\ \vdots \\ n_a(mN+N-1) \end{bmatrix}. \quad (6.15)$$

Since $N_a(mN)$,$n_p(mN)$ are independent white sense stationary Gaussian noises, a Kalman filter can be applied to estimate the states in a similar fashion to Section 6.2.3. For state estimation between two successive camera measurements, a sample open loop integration of accelerometer measurements is utilized, i.e.

$$\hat{x}(mN+i+1) = A\hat{x}(mN+i) + B\bar{U}_a(mN+i) \quad \text{for} \quad mN \leqslant i < (m+1)N. \quad (6.16)$$

The open loop integration will have a drifting error and will be corrected by the camera measurement every N steps.

6.3.2 Experimental Results

The two link manipulator shown in Figure 2.7 is utilized to evaluate the two-dimensional estimation performance. The end-effector is shown in Figure 6.7. $\boxed{1}$ is the accelerometer, $\boxed{2}$ is the cable for the accelerometer, $\boxed{3}$ is the camera link cable, $\boxed{4}$ is the camera, $\boxed{5}$ is the holder for camera and lens, $\boxed{6}$ is the lens, and $\boxed{7}$ is the illumination device.

The camera is a CMOS monochrome camera manufactured by Photonfocus, Model MV-D640(c). The lens is manufactured by Edmund, Model VZM300. The parameters for the camera and lens are shown in Table 6.1. The data and image acquisition boards, PCI-7831R and PCIe-1429 respectively, were provided by National Instruments. The real-time system is National Instruments LabVIEW ETS real-time module 8.5.

Table 6.1: Parameters of camera and lens

Specifications	Values	Units
Frame Rate	50	Hz
Delay Time	20	ms
Field of View	6×8	mm^2
Resolution	640×480	pixel
Focal Length	6	cm

The reference trajectory in joint space is generated based on a 10 cm marker. As shown in Figure 6.8, the marker is composed by 21 even distributed different size black spots. The two link manipulator is controlled by an independent joint loop PID controller using encoder feedback to follow the marker in 2 s. At the same time, the camera is used to produce position measurements for the KKF by observing the marker.

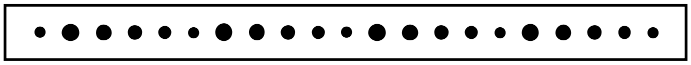

Figure 6.8: Marker to identify the position of the end-effector

The control system for the two link manipulator is shown in Figure 6.9. For the independent joint loop PID controller, the control law for each joint is the same as in Section 5.3.1. The sampling period for joint controller and accelerometer is 1 ms, while the sampling period for camera is 20 ms.

Figure 6.10 shows the position estimate by fusing accelerometer and camera measurements. They are compared with the encoder measurements. The position from the encoder is calculated by (6.12). In order to see the difference between encoder measurement and KKF estimation, only a small time range is plotted. Blue lines show the encoder measurement while green lines show the KKF estimates. The upper figure shows the movement in X axis while the lower figure shows the movement in Y axis.

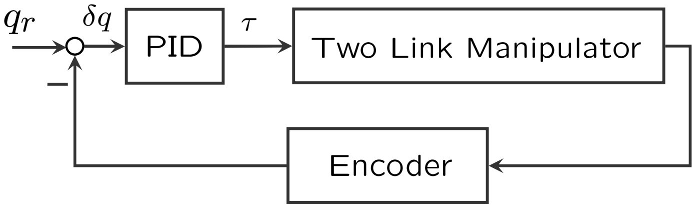

Figure 6.9: Block diagram of encoder feedback system for two link manipulator

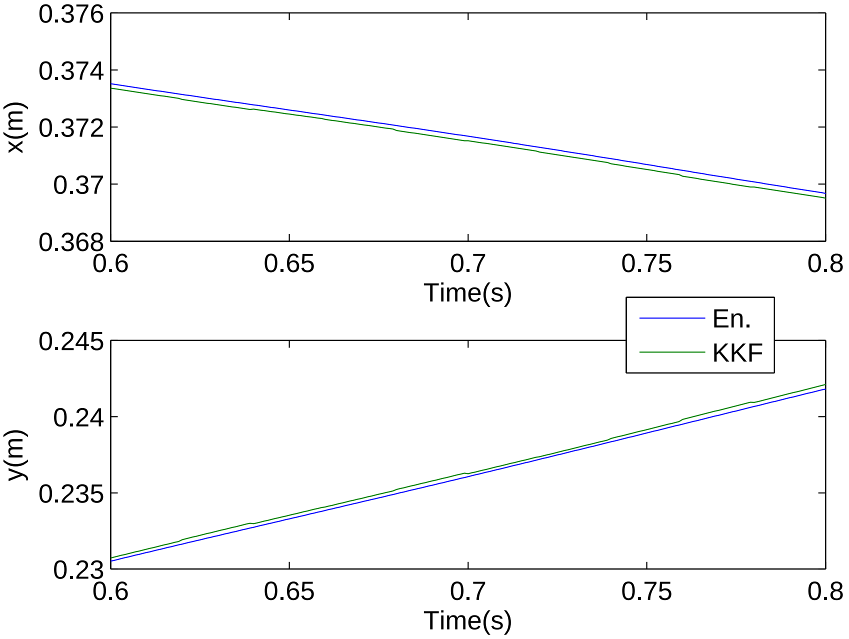

Figure 6.10: KKF estimates of position

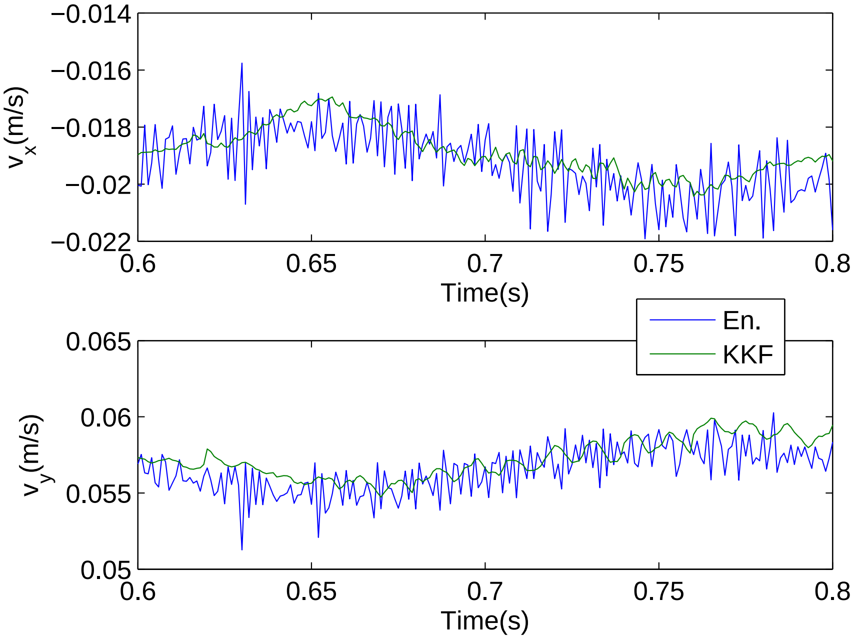

Figure 6.11: KKF estimates of velocity

We can clearly see the vision correction every 20 ms in the the lower figure, but for the upper figure, it can barely be seen. It is because the movement in Y axis is much larger than that in X axis. Figure 6.10 shows that the position estimation error is less than 1 mm. This means that the end-effector position can be estimated by the KKF pretty accurately.

Figure 6.11 shows the velocity estimate by fusing accelerometer and camera measurements. They are also compared with the encoder measurements. The velocity from the encoder is obtained by a one step backward difference of the encoder measurements, and it is very noisy. However, the KKF gives a much smoother velocity profile for both X and Y axes.

6.4 Summary

In this chapter, end-effector motion estimation based on vision and acceleration sensors was discussed. Since the vision sensor had a low sampling rate and a large measurement delay, a lifting technique was utilized to handle multi-rate issue while a state augmentation technique was utilized to handle the large measurement delay issue. The benefits of the two techniques were demonstrated by experiments. The proposed end-effector sensing scheme was experimentally verified for one-dimensional and two-dimensional cases respectively.

Chapter 7

Kinematic Kalman Filter Based Control

In this chapter, the KKF estimates obtained in the previous chapter are utilized as feedback signals to control the single joint setup and the two link manipulator. Closed loop stability is the main concern for each setup. The tracking performance for the two setups using KKF feedback is discussed and the results are compared with those of encoder feedback systems.

7.1 Introduction

The Kinematic Kalman filter(KKF) as an end-effector sensing technology has been studied for a long time. At the beginning, the accelerometer and the encoder measurements were integrated for velocity estimation. As sensor technology continues to improve, cameras with a fast sampling rate are becoming more frequently utilized in industry for motion detection. Although there is a considerable amount of research done about KKF, they are mainly concerned with integrating different sensor measurements to estimate the motion of the end-effector. None of them discussed the closed loop stability of the whole system when utilizing the KKF estimates as feedback signals. In this chapter, we focus on using KKF estimates as feedback signals for the single joint setup and the two link manipulator.

7.2 Control of Single Joint Setup

Even though Chapter 6 shows that the KKF estimates are very close to the encoder values, it does not guarantee that the closed loop performance with encoder feedback and that with KKF feedback are the same. Once the estimates are utilized as feedback signals, the system may become unstable. In Chapter 5, we saw that the return difference equality indicates that the steady state Kalman filter gain L depends only on the ratio of process noise and measurement noise covariances W/V. Open loop frequency responses with KKF feedback for different W/V were measured. It is shown that by properly choosing the ratio, the KKF feedback control system may be robustly stabilized.

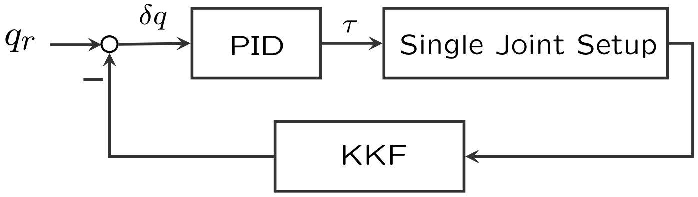

Figure 7.1: Block diagram of KKF feedback system for single joint setup

Figure 7.1 shows the block diagram of KKF feedback control system. In the experiments, the reference trajectory, the controller and the setup are all the same as in Section 6.2.4. The only difference is that the feedback signals are KKF estimates.

Figure 7.2 shows the tracking performance of KKF feedback control system for $T_s =$ 0.4 ms and $N = 20$. The red line is the reference trajectory, the blue line is the encoder feedback result, and the green line is the KKF feedback result. In order to see the difference between encoder feedback and KKF feedback, the tracking errors are plotted in Figure 7.3. The blue line is the tracking error for encoder feedback and the green line is the tracking error for KKF feedback. From the two figures, we can see that the tracking performance of the KKF feedback control system is almost the same as that of the encoder feedback control system.

In order to see the performance of KKF feedback control system for different sampling rates, another experiment was performed for $T_s = 1$ ms and $N = 20$. The only difference compared with the previous experiment is the sampling period. Figure 7.4 shows the tracking performance, and Figure 7.5 shows the tracking errors. Legends are exactly the same as the previous figures. The two figures show the experimental results between 1.2 ms and 1.3 ms. Comparing with the previous experiment, reducing sampling rate does not have much effect on the tracking performance. This conclusion is very important if the position sensor is the camera, which has a 20 ms sampling period and a 20 ms measurement delay.

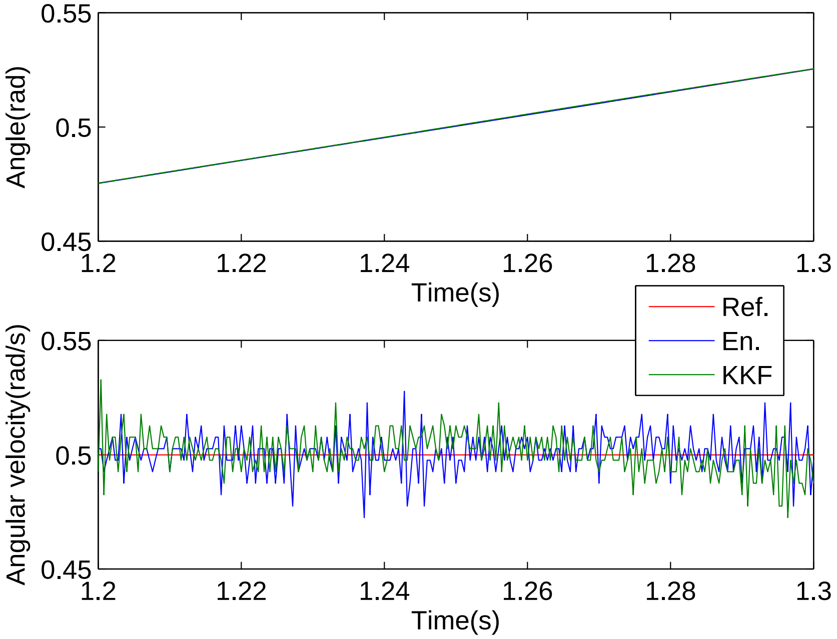

Figure 7.2: Tracking performance for single joint setup at $T_s = 0.4$ ms

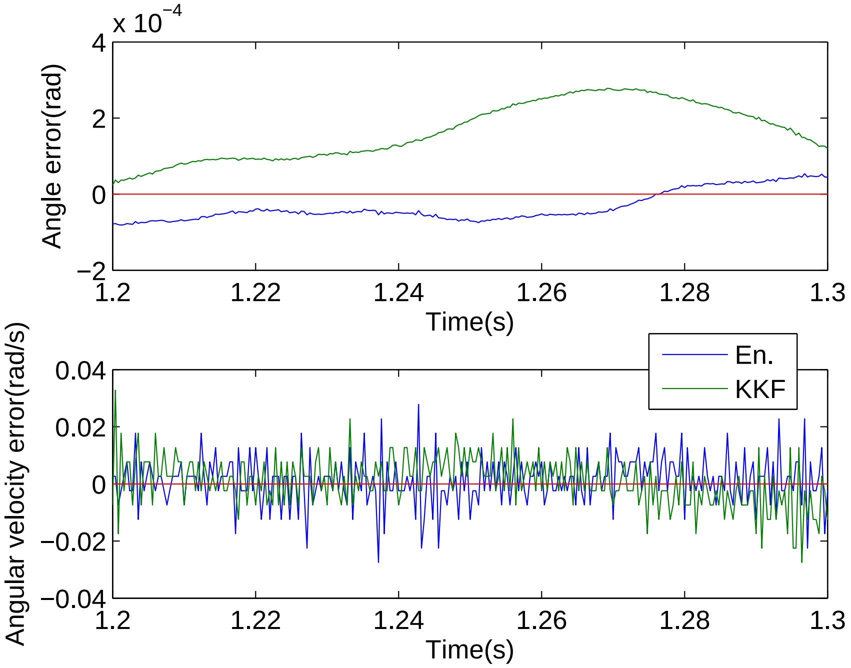

Figure 7.3: Position and velocity tracking error for single joint setup at $T_s = 0.4$ ms

The two experiments show that the tracking error for KKF feedback system is almost the same as that for encoder feedback system. This implies that the proposed estimation scheme may perform as good as installing a load side encoder on the end-effector.

7.3 Control of Two Link Manipulator

From Section 6.3 we know that the KKF estimation for the two link manipulator is performed in task space. Therefore it is convenient to control the two link manipulator by using task space control [31]. The forward kinematics of the two link manipulator is given by (6.12), denoted it as $f(q)$, i.e.

$$f(q) \triangleq \begin{bmatrix} l_1 \cos(q_1) + l_2 \cos(\alpha) \\ l_1 \sin(q_1) + l_2 \sin(\alpha) \end{bmatrix} \tag{7.1}$$

where $\alpha = q_1 + q_2$, l_1 and l_2 are the lengths of the first link and the second link respectively.

The manipulator Jacobian, which maps velocities from joint space to task space, is given by

$$J(q) \triangleq \frac{\partial f(q)}{\partial q} = \begin{bmatrix} -l_1 \sin q_1 - l_2 \sin(\alpha) & -l_2 \sin(\alpha) \\ l_1 \cos q_1 + l_2 \cos(\alpha) & l_2 \cos(\alpha) \end{bmatrix}. \tag{7.2}$$

For the two link manipulator, singularities of the Jacobian occur when $q_2 = n\pi, n \in \mathbb{N}$. Those points should be avoided in trajectory design.

The block diagram of the KKF feedback system for the two link manipulator is shown in Figure 7.6. Since the reference trajectory and the KKF are both in task space, the tracking error signals are also in task space. It is necessary to convert the error signals from task space to joint space by using Jacobian inverse and Jacobian derivative (7.3) for joint level control. Notice that the Jacobian inverse and Jacobian derivative need the information of joint angle and angular velocity, which are approximated by the reference trajectory, i.e.

$$\delta q = J^{-1}(q_r)\delta p \tag{7.3a}$$

$$\delta\omega = J^{-1}(q_r)(\delta v - \dot{J}(q_r, \omega_r)\delta q) \tag{7.3b}$$

In experiments, the reference trajectory is the task space version of the trajectory mentioned in Section 6.3. The sampling rate of the joint controller and the accelerometer is 1 kHz, while the sampling rate of the camera is 50 Hz. If the position

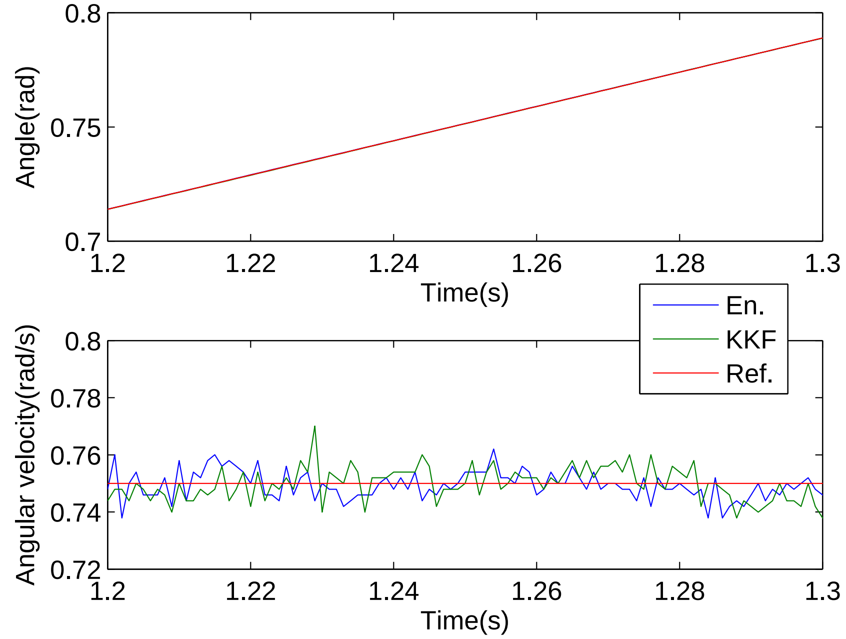

Figure 7.4: Tracking performance for single joint setup at $T_s = 1$ ms

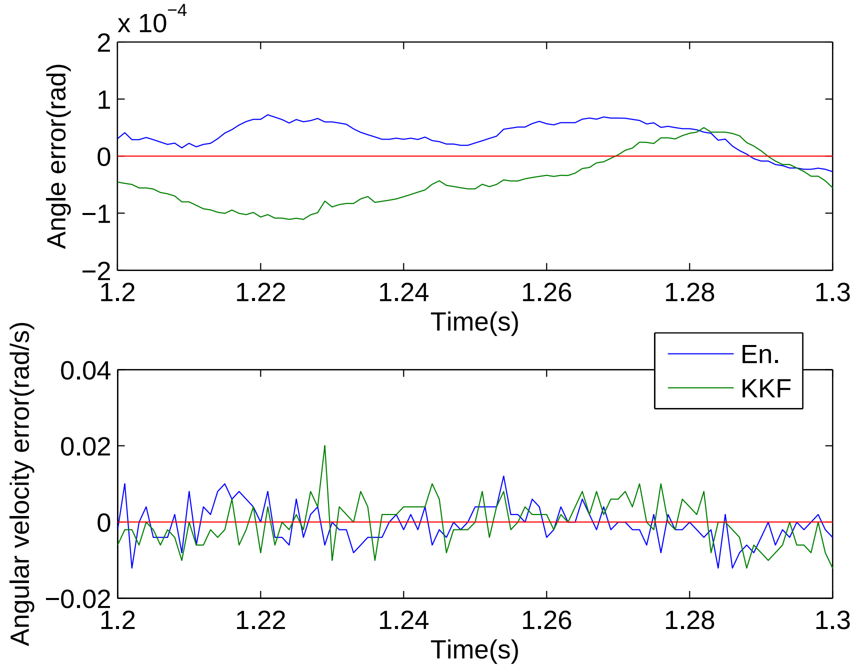

Figure 7.5: Position and velocity tracking error for single joint setup at $T_s = 1$ ms

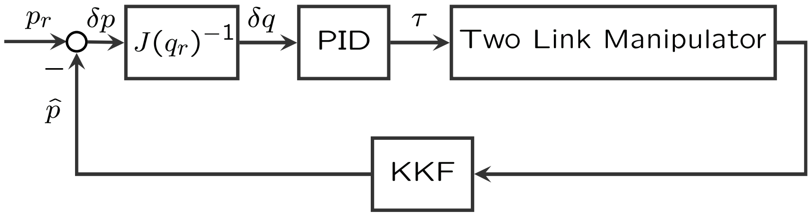

Figure 7.6: Block diagram of KKF feedback system for two link manipulator

and velocity feedback signals are both from KKF estimates, the experimental system becomes unstable. However, if the position feedback signal is from KKF and the velocity signal is from encoder, the system remains stable. It turns out that the velocity signals have a significant effect on the closed loop stability. Experimental results shown in this chapter are the stable cases.

Figure 7.7 compares the position tracking performance of the KKF feedback system with that of the encoder feedback system. Blue lines are the position measurements when the loop is closed with the encoder signal while the green lines are the position measurements when the loop is closed with the KKF estimate. The position is calculated from the encoder measurements by the forward kinematics. Red lines are the reference position signals. For clarity purposes, only a small time range is showed.

Figure 7.8 compares the velocity tracking performance of the KKF feedback system with that of the encoder feedback system. Blue lines are the velocity measurements when the loop is closed with the encoder signal while the green lines are the velocity measurements when the loop is closed with the KKF estimate. They are calculated from the one step backward difference of the position signal in Figure 7.7. Red lines are the reference velocity signals.

From these two figures, we can see that although the encoder feedback system performs slightly better than the KKF feedback system, the encoder feedback system requires a high resolution load side encoder, which may not be available for industrial applications. However, the KKF feedback system requires only several low cost sensors mounted on the end-effector.

7.4 Summary

In this chapter, experiments using KKF estimates as feedback signals for the single joint setup and the two link manipulator were performed. The tracking performance of the KKF feedback system was compared with that of the encoder feedback system for both the single joint setup and the two link manipulator. It was shown that the KKF feedback system performs as good as the encoder feedback system.

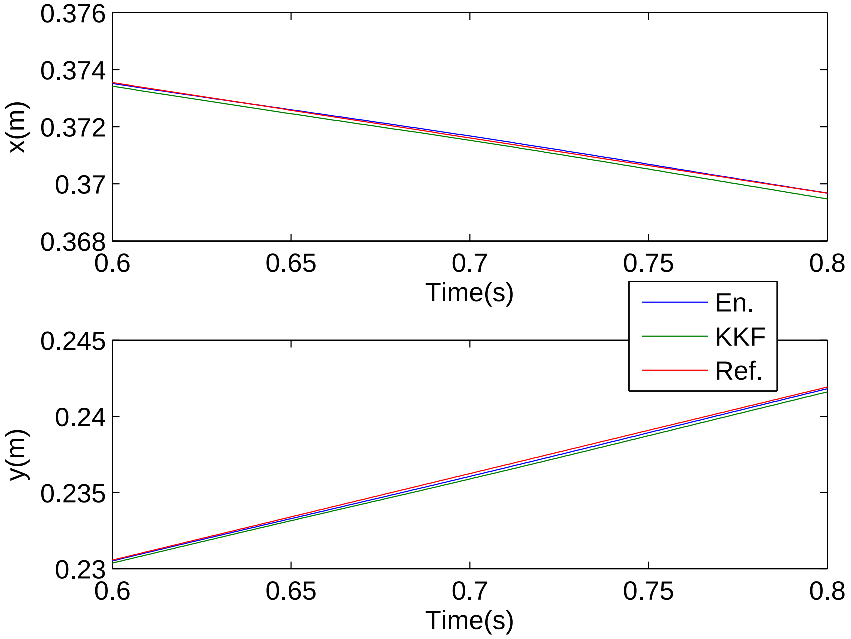

Figure 7.7: Position comparisons between encoder and KKF feedback systems

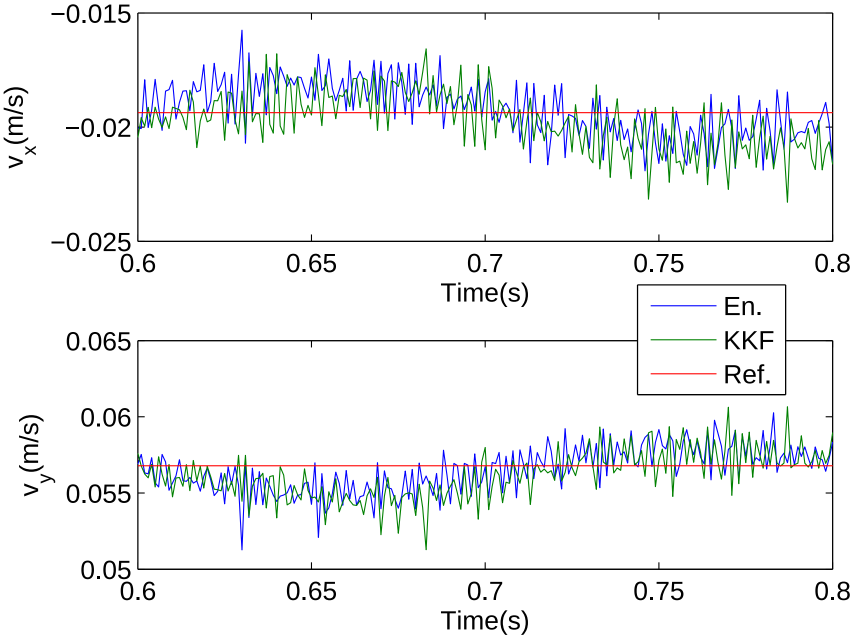

Figure 7.8: Velocity comparisons between encoder and KKF feedback systems

Chapter 8

Conclusions and Future Research

8.1 Conclusions

Drive trains are essential for a variety of positioning systems used in factory automation. The signal for feedback control is often obtained from the motor only, which sets the performance limitation in terms of positioning and tracking accuracy of the end-effector. To overcome such limitations, this dissertation suggested two approaches that utilize additional sensors.

At the beginning of this dissertation, a single joint setup and a two link robot manipulator were introduced. More specifically, the system identification and friction identification were performed for the single joint setup, while the dynamic model and of its associated parameters were given for the two link manipulator.

In the first part of the dissertation, an online reference trajectory generation scheme based on vision information was presented. Based on the contours captured by a camera, a reference trajectory for a two link manipulator was generated to follow the contour at selected speeds. Since the field of view of the camera is very small, the contour was approximated either as a straight line or as a circular arc. The reference trajectory was first generated in task space, then it was converted to joint space for joint level control. The experimental results of the two link manipulator following different contours at selected speeds demonstrated the effectiveness of the proposed reference trajectory generation scheme.

The second part of the dissertation addressed inertial and vision sensor based end-effector sensing and control for the single joint setup and the two link manipulator. Since this part is based on the kinematic Kalman filter(KKF), the KKF design was discussed at the beginning. It was emphasized that the steady state KKF gain depended only on the ratio of the process noise and measurement noise covariances, and this point was confirmed by the symmetric root locus method. It was noted that the ratio must be selected by examining the estimation accuracy as well as the closed loop robustness of KKF feedback control. A multi-rate KKF with large measurement delay was proposed for both the single joint setup and the two link manipulator. The estimation results were utilized as feedback signals to control the two setups. Experimental results showed the effectiveness of the proposed end-effector motion estimation scheme.

8.2 Future Research

For future work, since all the parameters for the two link manipulator may be lumped into three independent constants, adaptive system identification based on [34, 46] can be performed. This scheme can be incorporated for real-time adjustment of the feedback control gains. In addition, adaptive control of the two link manipulator can also be investigated [47]. The robust state estimation by KKF may be an advantage in the design of these adaptive systems.

For the work regarding vision based contour following, we did not consider the case where the contour contains sharp corners. If the contour has sharp corners and we still want to keep a constant speed, as the end-effector passes the corner, the vision sensor should be able to detect the corner and generate the initial trajectory to lead into the corner and an additional trajectory to reorient the end effector such that the end-effector can exit the corner without any sudden jerks or accelerations. In other words, the additional trajectory should be designed to smoothly connect the two sides of the corner. If the end-effector is performing any functionality on the contour, it should be turned off on the extra trajectory. The contour following work can also be extended to the three-dimensional case. In that case, more vision sensors are required and methods of interpreting the captured images to obtain accurate end-effector position and orientation information may become substantially more complicated. Other sensors, such as additional accelerometers, gyros, may be incorporated as well.

For the work of inertial and vision sensor based end-effector sensing and control, the motivation is end-effector motion estimation for indirect drive trains. However, limited to the experimental setup, we only verified the proposed scheme on direct drive trains. One possible future direction is the implementation of the scheme to indirect drive trains to show its effectiveness. In addition, it is essential to extend the proposed scheme to the three dimensional case. In that case, gyros and more cameras are required. How to combine these sensor measurements to obtain end-effector motion information promptly and accurately will be a challenging problem. The validity and effectiveness of the proposed sensor fusion scheme should eventually be demonstrated for articulated six axes industrial robots.

Bibliography

[1] M. Tomizuka, “Mechatronics: from the 20th to 21st century,” *Control Engineering Practice*, vol. 10, August 2002.

[2] Y. Shirai and H. Inoue, “Guiding a robot by visual feedback in assembling tasks,” *Pattern Recognition*, vol. 5, pp. 99–108, 1973.

[3] K. Hashimoto, “A review on vision-based control of robot manipulators,” *Advanced Robotics*, vol. 17, no. 10, pp. 969–991, 2003.

[4] J. Hill and W. T. Park, “Real time control of a robot with a mobile camera,” in *9th International Symposium on Industrial Robots*, March 1979.

[5] S. Hutchinson, G. D. Hager, and P. I. Corke, “A tutorial on visual servo control,” *Trans. on Robotics and Automation*, vol. 12, no. 5, pp. 651–670, 1996.

[6] P. I. Corke, “Dynamic issues in robot visual-servo systems,” in *International Symposium on Robotics Research*, pp. 488–498, Springer, 1995.

[7] Y. Nakabo, M. Ishikawa, H.Toyoda, and S. Mizuno, “1ms column parallel vision system and it’s application of high speed target tracking,” in *Proceedings of the 2000 IEEE International Conference on Robotics & Automation*, 2000.

[8] A. Sanderson and L. Weiss, “Image-based visual servo control using relational graph error signals,” in *Proceedings of the International Conference on Cybernetics and Society*, pp. 1074–7, 1980.

[9] A. Namiki and M. Ishikawa, “Vision-based online trajectory generation and its application to catching,” *Control Problems in Robotics*, pp. 249–264, 2002.

[10] G. Rouleau, S. Verma, and I. Sharf, “Vision-based tracking and trajectory generation for robotic capture of objects in space,” in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2005.

[11] J. T. Feddema and O. R. Mitchell, “Vision guided servoing with feature based

trajectory generation," *IEEE Trans. on Robotics and Automation*, vol. 5, no. 5, 1989.

[12] E. Coste-Manière, P. Couvignou, and P. K. Khosla, "Visual servoing in the task-function framework: A contour following task," *Journal of Intelligent and Robotic Systems*, vol. 12, 1995.

[13] J. Lin, H. T. Chen, P. Jiang, Y. J. Wang, and P. Y. Woo, "Curve tracking and reproduction by a robot with a vision system," *Journal of Robotic Systems*, vol. 16, 1999.

[14] H. Zhang, H. Chen, N. Xi, G. Zhang, and J. He, "On-line path generation for robotic deburring of cast aluminum wheels," in *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2400–2405, 2006.

[15] H. Cheng and M. Tomizuka, "Vision-based online trajectory generation for contour following," in *Proceedings of 2008 ISFA (2008 International Symposium on Flexible Automation)*, June 2008.

[16] H. Cheng, "Contour following with and without vision sensor," Master's thesis, University of California at Berkeley, 2008.

[17] L. M. Sweet and M. C. Good, "Redefinition of the robot motion-control problem," *IEEE Control Systems Magazine*, vol. 5, no. 3, pp. 18–25, 1985.

[18] D. Lee and M. Tomizuka, "Multirate optimal state estimation with sensor fusion," in *Proceedings of the American Control Conference*, pp. 2887–2892, 2003.

[19] H. Shim, M. Kochem, and M. Tomizuka, "Use of accelerometer for precision motion control of linear motor driven positioning system," in *Proceedings of the American Control Conference*, 1998.

[20] S. Jeon, M. Tomizuka, and T. Katou, "Kinematic kalman filter for robot end-effector sensing," *ASME Journal of Dynamic Systems, Measurement and Control*, vol. 131, no. 2, 2009.

[21] C. D. Wann and J. H. Gao, "Sensor motion tracking by imm-based extended kalman filters," in *Proceedings of the 2009 Fourth International Conference on Innovative Computing, Information and Control*, pp. 1377–1380, 2009.

[22] H. Yung, G. Chesi, and Y. Hung, "Position and motion estimation for visual robot control with planar targets," in *Proceedings of the 7th Asian Control Conference*, August 2009.

[23] Y. Nam and S. L. Dickerson, "Position estimation with accelerometer and vision

measurement," in *Proceedings of the American Society of Mechanical Engineers, Dynamic Systems and Control Conference*, pp. 1–8, 1991.

[24] A. Huster and S. M. Rock, "Relative position estimation for manipulation tasks by fusing vision and inertial measurements," in *Proceedings of the Oceans Conference*, pp. 1025–1031, 2001.

[25] D. Strelow and S. Singh, "Motion estimation from image and inertial measurements," *International Journal of Robotics Research*, vol. 23, no. 12, pp. 1157–1195, 2004.

[26] P. Gemeiner, P. Einramhof, and M. Vincze, "Simultaneous motion and structure estimation by fusion of inertial and vision data," *The International Journal of Robotics Research*, vol. 26, June 2007.

[27] L. Armesto, J. Tornero, and M. Vincze, "Fast ego-motion estimation with multi-rate fusion of inertial and vision," *The International Journal of Robotics Research*, vol. 26, no. 6, 2007.

[28] Z.-H. Jiang and T. Eguchi, "Vision feedback based end-effector motion control of a flexible robot arm," in *2007 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2007*, October 2007.

[29] B. Friedland, "Optimum steady-state position and velocity estimation using noisy sampled position data," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-9, November 1973.

[30] H. Cheng and M. Tomizuka, "Inertial and vision sensor based end-effector sensing and control for robot manipulators," in *Proceedings of 2010 DSCC (2010 ASME Dynamic Systems and Control Conference)*, September 2010.

[31] J. J. Craig, *Introduction to Robotics: mechanics and control.* Electrical and Computer Engineering: Control Engineering, Reading, Massachusetts: Addison-Wesley, 1989.

[32] H. Olsson, K. J. Åström, C. C. de Wit, M. Gafvert, and P. Lischinsky, "Friction models and friction compensation," *European Journal of Control*, vol. 4, no. 3, 1998.

[33] B. Armstrong-Hélouvry, *Control of Machines with Friction.* The Kluwer international series in engineering and computer science. Robotics, Norwell, Massachusetts: Kluwer Academic Publishers, 1991.

[34] N. Sadegh and R. Horowitz, "Stability and robustness analysis of a class of adaptive controllers for robotic manipulators," *The International Journal of Robotics Research*, vol. 9, June 1990.

[35] C. H. An, C. G. Atkeson, and J. M. Hollerbach, *Model-Based Control of A Robot Manipulator.* The MIT press, 1988.

[36] H.-W. Kim and S.-K. Sul, "A new motor speed estimator using kalman filter in low-speed range," *IEEE Transactions on Industrial Electronics*, vol. 43, August 1996.

[37] S.-H. Song and S.-K. Sul, "An instantaneous speed observer for low speed control of ac machine," in *Proceedings of Applied Power Electronics Conference and Exposition*, vol. 2, February 1998.

[38] D. Lee and M. Tomizuka, "State / parameter / disturbance estimation with an accelerometer in precision motion control of a linear motor," in *Proceedings of ASME IMECE*, 2001.

[39] R. Horowitz and M. Tomizuka, "Class notes of Advanced Control System II."

[40] S. Jeon and M. Tomizuka, "Benefits of acceleration measurement in velocity estimation and motion control," *Control Engineering Practice*, vol. 15, pp. 325–332, 2007.

[41] M. Nakamura, "Relationship between steady state kalman filter gain and noise variances," *International Journal of Systems Science*, vol. 13, May 1982.

[42] B. D. O. Anderson and J. B. Moore, *Optimal Control: Linear Quadratic Methods.* Information and System Sciences Series, Englewood Cliffs, New Jersey: Prentice-Hall, 1989.

[43] M. Tomizuka, "Multi-rate control for motion control applications," in *8th IEEE International Workshop on Advanced Motion Control*, pp. 21–29, 2004.

[44] E. B. Dahlin, "Designing and tuning digital controllers," *Instruments and Control Systems*, vol. 41, pp. 77–83, 1968.

[45] K. Warwick and D. Rees, *Industrial digital control systems.* London, United Kindom: Peter Peregrinus Ltd, 1988.

[46] N. Sadegh and R. Horowitz, "Stability analysis of an adaptive controller for robotic manipulators," in *Proceedings of International Conference on Robotics and Automation*, March 1987.

[47] J.-J. E. Slotine and W. Li, "On the adaptive control of robot manipulators," *The International Journal of Robotics Research*, vol. 6, no. 3, 1987.

Appendix A

Proof of (5.11)

$$[1+G_o(z)]Q[1+G_o(z^{-1})] = V + G(z)WG(z^{-1}). \tag{A.1}$$

Proof. Since

$$\begin{aligned} L &= AF \\ &= AMC^T(CMC^T+V)^{-1} \\ &= AMC^TQ^{-1}, \end{aligned}$$

the left side of (A.1)

$$\begin{aligned} \text{L.H.S} &= [1+C\Phi(z)L]Q[1+L^T\Phi(z^{-1})^TC^T] \\ &= Q + C\Phi(z)LQ + QL^T\Phi(z^{-1})^TC^T + C\Phi(z)LQL^T\Phi(z^{-1})^TC^T \\ &= Q + QL^T\Phi(z^{-1})^TC^T + C\Phi(z)LQ \\ &\quad + C\Phi(z)(AMC^TQ^{-1}CMA^T)\Phi(z^{-1})^TC^T \\ &= V + CMC^T + CM^TA^T\Phi(z^{-1})^TC^T + C\Phi(z)AMC^T \\ &\quad + C\Phi(z)(AMC^TQ^{-1}CMA^T)\Phi(z^{-1})^TC^T. \end{aligned}$$

From the Riccati equation (5.6) we know

$$BWB^T = M - AMA^T + AMC^TQ^{-1}CMA^T$$

and

$$\begin{aligned} M - AMA^T &= M - Az^{-1}M + AMz^{-1} - AMA^T \\ &= (zI-A)z^{-1}M + AM(z^{-1}I - A^T) \\ &= \Phi(z)^{-1}z^{-1}M + AM(\Phi(z^{-1})^{-1})^T. \end{aligned}$$

Thus, the right side of (A.1)

$$\begin{aligned}
\text{R.H.S.} &= V + C\Phi(z)BWB^T\Phi(z^{-1})^TC^T \\
&= V + C\Phi(z)(M - AMA^T + AMC^TQ^{-1}CMA^T)\Phi(z^{-1})^TC^T \\
&= V + C\Phi(z)(\Phi(z)^{-1}z^{-1}M + AM(\Phi(z^{-1})^{-1})^T)\Phi(z^{-1})^TC^T \\
&\quad + C\Phi(z)(AMC^TQ^{-1}CMA^T)\Phi(z^{-1})^TC^T \\
&= V + Cz^{-1}M\Phi(z^{-1})^TC^T + C\Phi(z)AMC^T \\
&\quad + C\Phi(z)(AMC^TQ^{-1}CMA^T)\Phi(z^{-1})^TC^T.
\end{aligned}$$

Compare L.H.S with R.H.S, we only need to show

$$CMC^T + CM^TA^T\Phi(z^{-1})^TC^T = Cz^{-1}M\Phi(z^{-1})^TC^T.$$

Since M is symmetric, $M = M^T$,

$$\begin{aligned}
& Cz^{-1}M\Phi(z^{-1})^TC^T - CM^TA^T\Phi(z^{-1})^TC^T \\
=\ & C(M^Tz^{-1}\Phi(z^{-1})^T - M^TA^T\Phi(z^{-1})^T)C^T \\
=\ & C(M^T(Iz^{-1} - A^T)\Phi(z^{-1})^T)C^T \\
=\ & CM^TC^T \\
=\ & CMC^T.
\end{aligned}$$

Therefore, (A.1) holds. □