Control of a Pendulum using Extended Kalman Filtering

Rajmohan Madhavan
B.E. (First Class)
Anna University India

May 1997

A thesis submitted for the degree of Master of Engineering (by Research)
of The Australian National University

Department of Engineering
Faculty of Engineering and Information Technology
The Australian National University

Declaration

I declare that this thesis is my own work and has not been submitted in any form for another higher degree or diploma at any other University or other Institution. Information derived from published or unpublished work of others has been acknowledged and a list of references is given. The work described in this thesis results from a collaborative effort with my supervisor, Professor Darrell Williamson.

The results of the state estimation approach developed in this thesis have been presented in the following conference:

> R. Madhavan and D. Williamson, **State Estimation of Polynomic Nonlinear Systems**, Proceedings of the Australian CONTROL 97 Conference, Pages 598-603, Sydney, Australia, October 20-22, 1997.

Signed ..M. Rajmohan..

Date ..9 DEC 1997..

Dedicated to my Parents

To whom one always owes more than what is said.

Abstract

This thesis deals with the control of a pendulum using extended Kalman filtering. In particular, it considers the tracking control of the pendulum using estimated states, for which a discrete extended Kalman filtering approach based on the tensor techniques is proposed. Three sensors, an encoder, a rate-gyro and a tilt-sensor are analysed for providing the measurements for the estimation of the states of the pendulum and the feasibility of employing these sensors, subject to various factors such as cost, estimation accuracy etc. is discussed. The estimation approach developed is more numerically efficient than the standard extended Kalman filtering approach. A linear Kalman filtering approach is also developed based on a linearised model of the pendulum and it is shown that the extended Kalman filtering approach is to be preferred both in terms of estimation and control accuracy.

Acknowledgements

So finally I have come to the end of my thesis. I would like to sincerely thank my supervisor, Prof. Darrell Williamson for his invaluable guidance and support, without which, this thesis would not have been possible. It has been a rewarding time of great learning for me, both academically and personally.

Also I would like to thank my co-graduate students, in particular, Geir, Werner and Endra, for being good friends and my room-mates Will and Matt for putting up with me. Thanks to all in the Department of Engineering for making my stay a pleasant one. I have had a wonderful time in Canberra during the entire duration of the Masters program.

Last but not least, I would like to thank my parents for their love and prayers and for being there always for me.

Nomenclature

J	total inertia of the pendulum [Nm]
m	mass of the pendulum bob. [kg]
g	acceleration due to gravity $[\frac{m}{sec^2}]$
ℓ	distance of the bob from the pivot [m]
τ	control torque [Nm]
α_{ts}, β_{ts}	constants associated with the tilt-sensor [sec]
α_f	constant associated with the anti-aliasing filter [sec]
θ	angle measured from the vertical [rad]
$\dot{\theta}$	angular velocity of the pendulum $[\frac{rad}{sec}]$
$\ddot{\theta}$	angular acceleration of the pendulum $[\frac{rad}{sec^2}]$
$\hat{\theta}$	estimate of θ
$\hat{\dot{\theta}}$	estimate of $\dot{\theta}$
T_s	sampling interval [sec]
f_s	sampling frequency [Hz]
ΔT_s	Euler integration step-size [sec]
$\mathbf{x}_k$	state at the discrete time instant kT_s
$\mathbf{z}_k$	measurement made at the discrete time instant kT_s
$\hat{\mathbf{x}}_k(-)$	*a priori* estimate at the discrete time instant kT_s
$\hat{\mathbf{x}}_k(+)$	*a posteriori* estimate at the discrete time instant kT_s using measurements $\mathbf{z}_k$
$\hat{\mathbf{P}}_k(-)$	*a priori* covariance at the discrete time instant kT_s
$\hat{\mathbf{P}}_k(+)$	*a posteriori* covariance at the discrete time instant kT_s

	using measurements $\mathbf{z}_k$
K_k	Kalman filter gain matrix at the discrete time instant kT_s
$\omega(t)$	continuous process noise
$v(t)$	continuous measurement noise
ω_k	sampled process noise
v_k	sampled measurement noise
$\mathbf{Q}$	continuous covariance of the process noise
$\mathbf{R}$	continuous covariance of the measurement noise
$\mathbf{Q}_k$	discrete covariance of the sampled process noise
$\mathbf{R}_k$	discrete covariance of the sampled measurement noise

Abbreviations

KF	Kalman Filter
LKF	Linear (discrete) Kalman Filter
EKF	Extended Kalman Filter
PRBS	Pseudo Random Binary Sequence
ZOH	Zero Order Hold
DAC	Digital to Analog Converter
ADC	Analog to Digital Converter
LS	Least Squares
LSE	Least Squares Estimate

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

The field of robotics combines the traditional engineering disciplines of mechanical engineering, electrical engineering and computer science. It is a highly multidisciplinary application in the areas of controls, computers (both the hardware and software aspects), measurement technology (i.e., sensors), pattern-recognition techniques and hardware (e.g., vision systems) and various aspects of mechanical engineering, including statics, dynamics, kinematics and mechanical design. This follows from the four major components contained in most robotic systems today namely the manipulator, the power conversion unit, the sensory devices and the computer control system [K+89].

The present-day performance and stability requirements on robotic systems has necessitated the design of control methods that take into account the full nonlinear model dynamics. In the early days, the mechanical construction in robotic systems was overdesigned in the sense that heavy, and consequently, rigid links together with high-gear transmission mechanisms were employed. These old-fashioned robot systems do not satisfy the requirements of today for two main reasons: First, to achieve reasonable velocities, these heavy systems need high power actuation which is unattractive both from the economic and environmental point of view. Second, the need for high-gear

transmissions introduces many parasitic effects such as flexibility, friction and backlash, which impose severe limitations on performance and accuracy among other factors. To overcome these deficiencies, there has been a need to develop light-weight robot constructions actuated by direct-drive motors. The dynamics characterising such systems are, however, highly nonlinear. So for good motion control over the complete operation space, one has to take into account the nonlinear effects that are either directly, or consequentially associated with the dynamics of the system under consideration.

In order to determine whether or not a system is performing properly, and ultimately to control the system performance, the behaviour of the system at any given instant of time must be known, or in other words, the *states* of the system must be known. For example, in robotics, the state usually consists of position and velocity of the robotic arm under consideration. In order to determine the state or states of a system, a measuring device or devices called sensors which collect *measurements* (*observations*) of the system are required. Sensory devices may monitor position, speed, acceleration or torque. Typically, the sensor is connected to the actuator shaft. However, it could also be coupled to the output of the transmission. Sensory devices are elements that inform the robot controller about the status of the manipulator. This can be done continuously or only at the end of the desired motion. For instance, in some robots, the sensors provide instantaneous position, velocity and possibly acceleration information about the individual links that can be fed back to the control unit to produce the proper control of the mechanical system. More simply, the controller can be informed only when the individual links of the manipulator have reached the desired end positions. These sensory devices are normally corrupted by noise by the electronic and mechanical components associated with the measuring device. The problem of determining the states of a system from noisy observations is called *state estimation*. It is of great importance in engineering since the states are not always directly accessible. So with the available states, it may be possible to reconstruct the missing states. This is the case in most of the industrial applications today, though theoretically it is possible to assign sensors to measure a particular state but it may not be a viable option in the

sense that, it might increase the cost and time constraints by a significant factor. So estimation of states is almost imperative both from a practical and a theoretical point of view (in that it might not be feasible to assign sensors for measuring all the states).

The correct execution of the end effector motion is entrusted to the control system which will provide the joint actuators of the manipulator with the commands consistent with the desired motion trajectory. Control of end effector motion depends on an accurate analysis of the characteristics of the mechanical structure, actuators and sensors. The control methods may be classified according to the objective that is defined for the end-effector of the robot. One frequently encountered objective in robot control is point-point-control, also known as regulation, though this objective is rather restrictive. For this reason, the trajectory tracking or motion control objective for robots has become increasingly popular, since it significantly extends the application area of robots [Ber94]. For instance, current applications like welding, painting and grinding as well as future applications like sea-mineral processing and space station repair require motion control. For the proper control of the robot, either the states or at least the estimated states should be available for designing the proper control scheme that will help in achieving the desired goal. Since the dynamics of a robot are highly nonlinear, the estimation approach is not as simple as in the linear case. Though linearisation of the nonlinear system for a particular operating point is acceptable, no linearisation that is valid for all regions can be found. Therefore it is important that if the nonlinearity can be included in the system model, and the approach that is employed for the state estimation takes into consideration this nonlinearity.

Motivated by these considerations, this thesis deals with the control of a pendulum which takes into consideration the nonlinearity in the model when estimating the states. Tracking control of the pendulum is then accomplished using the estimated states.

1.2 Problem Formulation

This thesis concentrates on one specific but important class of nonlinear systems that arises when modelling a pendulum. A large number of mechanical systems approximately fall into this category. For example, a single-link of a robot can be modelled as a pendulum with the mass concentrated at a point. The pendulum is considered in the down position (as opposed to the upright position whose equilibrium point is unstable) corresponding to its stable equilibrium position. The pendulum system can be equipped with different sensors for the position and velocity (angular) measurements. Particular sensors that can be used for this purpose are the encoder, the rate-gyro and the tilt-sensor. Five distinct combinations of sensors are considered as follows:

i. Pendulum model with measurements from an encoder,
ii. Pendulum model with measurements from a rate-gyro,
iii. Pendulum model with measurements from an encoder and a rate-gyro,
iv. Pendulum model with measurements from a tilt-sensor and
v. Pendulum model with measurements from a rate-gyro and a tilt-sensor.

The best choice of sensor configuration will depend on various considerations such as cost, estimation accuracy and feasibility of employing a particular sensor with respect to a particular application. Based on the foregoing considerations, the objectives of this thesis are threefold:

Firstly, the pendulum (one link robot manipulator) is to be modelled as a mechanism that is instrumented with a sensor (either the encoder or the tilt-sensor) to measure the joint angle or the joint velocity (rate-gyro) or both (either the encoder and rate-gyro combination or the tilt-sensor and rate-gyro combination) and an actuator. Models that provide an accurate representation of the sensors are essential if the information provided by the sensors is to be used effectively. To make best use of the sensing system, it is important to have a detailed understanding of both the physics of the devices and the sources of error associated with the operation of the sensors. The values of the parameters associated with

the pendulum and the combined pendulum and tilt-sensor models are also usually unavailable, and therefore have to be determined.

Secondly, having adequately modelled the pendulum and having determined the factors that influence the operation of the sensors and the actuator, the problem of estimating the states of the pendulum with the measurements provided by each of the possible sensor combination will be considered and the necessary estimation approaches will be developed for each case. As indicated earlier, nonlinear estimation strategies are more complex and involved than in the linear case. The extended Kalman filtering extends the principles for the design of a linear Kalman filter to a nonlinear problem. In particular, two types of filtering, the standard extended Kalman filtering and the developed discrete extended Kalman filtering will be compared with respect to the numerical efficiency of the approaches. Also a linear Kalman filtering approach will be considered based on a linearised pendulum model. The two types of extended Kalman filters will then be compared with the linear Kalman filter in order to quantify the improvement in estimation accuracy at higher swing-angles.

Finally, it will be shown how the estimates that are obtained using the various estimation schemes discussed, can be employed in an effective way to achieve desired robotic control. In particular, it will be shown that the discrete-estimator based controller using a nonlinear pendulum model will not require any hardware modifications nor significant increase in computational power.

1.3 Contributions of the Thesis

The Thesis deals with the modelling, estimation and control of a simple pendulum about its stable equilibrium position. The state estimation and control are achieved and demonstrated via a simulation study. The states are estimated by employing a linear Kalman filter and three extended Kalman filters using measurements obtained from various sensor inputs. The sensors considered for this purpose are the encoder,

the rate-gyro and the tilt-sensor. The nonlinearity of the pendulum is modelled by using the Volterra series approximations. Control of the pendulum is then effected by using the internal model principle and the feedback linearisation technique, where the estimated states obtained by the proposed methods are employed. The whole discrete estimator based controller has the advantage of increased computational speed and does not demand any additional hardware configurations.

The Volterra functional representation has very often been used in the identification of nonlinear systems. The class of systems to which this has been successfully applied is one in which an analytical nonlinearity is interposed between two linear systems that are connected in cascade, and is referred to as the *extended generalised Wiener model structure*. However, to the author's knowledge, though the Volterra series and the tensor theory have been utilised in the analysis and identification of systems arising in communication and control, no significant attempt has yet been made to use them in conjunction with the discrete extended Kalman filtering algorithm for the state estimation of a nonlinear time-invariant system with emphasis on reducing the computational complexities.

This thesis demonstrates the versatility of the tensor techniques when combined with the Volterra series and the discrete extended Kalman filtering approach in estimating the states of a nonlinear model. The method developed is a *general* method applicable to models of any order and it uses the state space representation and thus can be easily implemented on a digital computer. The pendulum and the combined pendulum and tilt-sensor models serve as good examples demonstrating the advantages of the developed approach. This approach will be shown to be more numerically efficient than the standard extended Kalman filtering approach by a significant factor.

1.4 Organisation of the Thesis

Chapter 2 considers the modelling aspects. The equation describing the motion of the pendulum and the physics of the sensors and the actuator are discussed. Reasons for employing nonlinear analysis instead of the linearisation techniques are discussed, and it is shown how the inclusion of the nonlinearity of the pendulum is important for higher-swing angles of the pendulum. A parameter estimation algorithm is developed for identifying the pendulum and the tilt-sensor parameters. The concept of vector homogenous forms is introduced next and the ZOH equivalence of polynomic nonlinear systems is presented. The Volterra series representations of both the pendulum model and the combined pendulum and tilt-sensor models are derived. The experiments with the sensors and the actuator are also presented.

Chapter 3 presents the state estimation problem. It starts with a general introduction to Kalman filtering. A discrete-time linear Kalman filter, LKF, is employed for the state estimation of the pendulum and the results corresponding to the linear filtering case are presented. A general introduction to nonlinear filtering using the standard extended Kalman filtering approach and the difficulties associated with the nonlinear filtering are discussed next. An extended Kalman filter, EKF1, is then used for the state estimation where the involved computations in the implementation of this approach and the associated computational burdens are discussed. This is followed by the development of input-output relationships for polynomic nonlinear systems at discrete-time instants and the development of discrete extended Kalman filtering (EKF2 and EKF3) approaches based on such input-output relationships. It is then demonstrated how this approach can reduce the computational burden and so result in an increase in the computational speed. The results are presented for each sensor case. When the tilt-sensor is included, the combined model becomes a third-order model because of the tilt-sensor dynamics. The EKF2 and EKF3 approaches are then extended to the third-order case and thus it is shown that this approach can be extended to models of any order with the appropriate inclusions. A thorough treatment of the computations involved in the

implementation of the EKF2 and EKF3 approaches are presented and the results are compared with the standard EKF1 approach. The corresponding simulation and the estimation error variance results are also included.

Chapter 4 deals with the tracking control of the pendulum. Using the estimates obtained by the techniques developed in Chapter 3, the output of the pendulum model is made to track a pre-specified, time-varying reference trajectory. The control strategy (as opposed to the strategy for state estimation) is based on the feedback linearisation technique which attempts to cancel the nonlinearity of the model using the estimates obtained by the state estimator. An application of the internal model principle for linear control design provides the structure for the generation of the reference signals. The corresponding simulation results and the tracking error variances are also provided.

Chapter 5 summarises the results along with directions for further research.

1.5 Notations

The following notations are used in this thesis.

Bold-faced quantities

Vectors and matrices are denoted by small and capital bold-faced letters respectively. For example, $\mathbf{x}$ denotes a vector whereas $\mathbf{X}$ denotes a matrix.

Matrix notations

`Transpose`

$\mathbf{x}^T$ and $\mathbf{X}^T$ denote the transpose of the vector $\mathbf{x}$ and the matrix $\mathbf{X}$ respectively.

`Inverse`

$\mathbf{X}^{-1}$ indicates the inverse of the non-singular square matrix $\mathbf{X}$.

`Identity matrix`

$\mathbf{I}_n$ denotes an identity matrix of order n.

For example, the identity matrix of order 2 is given by

$$\mathbf{I}_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Mathematical notations

`Dot notation for derivatives with respect to time`

$\dot{f}(t)$ and $\ddot{f}(t)$ are used to denote the first two derivatives of f with respect to time, t. Alternatively, the first derivative is also denoted by $\frac{d}{dt}(f)$.

`Partial derivative`

$\frac{\partial \mathbf{x}}{\partial \mathbf{x_1}}$ indicates the partial derivative of $\mathbf{x}$ with respect to $\mathbf{x_1}$.

`Approximation`

$\approx$ means *approximately equal to.*

`Not equal to`

$\neq$ means *not equal to.*

`Equivalent to`

$\equiv$ means *equivalent to.*

`Mathematical expectation`

ξ indicates the mathematical expectation operator.

`Random variable notation`

The notation $\mathbf{x} \sim N(\mathbf{m}, \mathbf{P})$ indicates that $\mathbf{x}$ is a Gaussian (normal) random vector with mean $\mathbf{m}$ and covariance $\mathbf{P}$. For example, a one-dimensional random process x with mean m and standard deviation σ is denoted by $x \sim N(m, \sigma^2)$.

`For all`

$\forall$ means *for all.* For example, $x_j^k = 3 \ \forall \ j$ and k means $x_j^k = 3$ for all values of j and k.

`If and only if`

iff indicates *if and only if.*

`Norm`

$||\mathbf{x}||$ denotes the *2-norm* of $\mathbf{x}$.

`Vector homogenous form`

$\mathbf{x}^{[p]}$ represents a vector homogenous form of degree p such that $||\mathbf{x}^{[p]}||_2^2 = ||\mathbf{x}||_2^{2p}$.

For example, if $\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$, then $\mathbf{x}^{[3]} = \begin{pmatrix} x_1^3 \\ \sqrt{2}x_1^2x_2 \\ \sqrt{2}x_1x_2^2 \\ x_2^3 \end{pmatrix}$

Kalman filter abbreviations

`LKF`

This is a linear discrete-time Kalman filter. The plant model is continuous and the measurement model is discrete.

`EKF1`

This is an extended Kalman filter with a continuous-time plant model and a discrete-time measurement model where the nonlinearity in the plant model is a sinusoid. Both the state estimate and error covariance propagations are done once in every ΔT_s instants.

`EKF2`

This is an extended Kalman filter with a continuous-time plant model and a discrete-time measurement model where the nonlinearity in the plant model is approximated. Both the state estimate and error covariance extrapolations are done once in every sampling interval.

`EKF3`

This EKF is an approximated version of EKF2.

Chapter 2

Modelling

A mathematical description that provides the system's dynamic characteristics is called a mathematical model. By applying physical laws to a specific system, it is possible to develop a mathematical model that describes the dynamics of the system. Such a model may include unknown parameters, which must be then be evaluated by some procedure. In such a procedure, the system is subjected to a set of known inputs and from the corresponding outputs, these unknown parameters are obtained. The objective of modelling mechanical systems is to provide a way to mathematically express the relationships between input variables (such as applied torque) and output variables (such as the physical positions of various components). A well developed mathematical representation of a model provides the basis for further development and analysis.

The successful control of most robots depends on being able to obtain information about the joint(s) and/or end effector. It is therefore necessary to have devices (transducers) that provide such information which can be readily utilised. In particular, position, velocity and/or acceleration (or at least analog or digital representations of these quantities) must be measured to ensure that the robotic manipulator moves in a desired manner (e.g., a straight line) with little or no oscillation (i.e., overshoot) at the final position. These so-called internal state sensors must not only permit the required

degree of accuracy to be achieved, but they must also be cost-effective since each of the robot's axes will normally utilise such devices. As a consequence, the sensor selection and the decision to place it either on the load side or on the output of the joint actuator itself is influenced by such factors as overall sensor cost, power needs for a particular joint, maximum permissible size of the actuator, sensor resolution, the factors that influence the resolution and the need to monitor directly the actions of the joint. The data gathered by these sensors can then be utilised by the robot's controller to modify or adapt to a given situation. Of course, the particular application will largely influence the type, construction and cost of such sensors and actuators. These ideas form the backbone of this chapter along with the operation of the sensors and actuator themselves as applied to the control of the pendulum.

The physics of the sensors and the actuator considered are discussed briefly. The sensors that are considered are the encoder, the rate-gyro and the tilt-sensor. The operation of the sensors, their specifications and other significant points are presented. Next, the equation of motion of the pendulum is derived and the pendulum model with different combinations of sensors is introduced where it is shown how each sensor can be employed for measuring the states of the pendulum. In addition to the three cases pertaining to the measurements provided by the individual sensors, two additional cases with the combination of encoder and rate-gyro and tilt-sensor and rate-gyro are also considered. A parameter estimation algorithm is developed for estimating the parameters of the linearised pendulum model and the simulation results are given. Then the ZOH equivalence of polynomic nonlinear systems is presented where the concept of vector homogenous forms is introduced along with the relevant properties. The Volterra series representation of the pendulum and the combined pendulum and tilt-sensor models are derived next. The results of the experiments are also provided for each of the above sensor and the actuator.

2.1 Sensors and Actuator

2.1.1 Encoder

One of the most commonly encountered position sensors is the optical encoder. This device has a resolution that is usually more than adequate for robotic applications. Accurate digital encoders have to be attached to the motor shafts to detect the position of the pendulum-arm. The explicit encoders considered were the Hewlett Packard HEDS-series encoders. The HEDS-series [Pac95] is a high performance, low cost, optical encoder module. When operated in conjunction with a code wheel, this module detects the rotary position. The module consists of a lensed Light Emitting Diode (LED) source and a detector IC enclosed in a small C-shaped plastic package. Due to a highly collimated light source and a unique photo-detector array, the module is extremely tolerant to mounting misalignment.

Operation of the encoder

The HEDS-series encoder is a C-shaped emitter/detector module. Coupled with a code-wheel, it translates the rotary motion of a shaft into a two-channel digital output. The module contains a single LED as its light source. The light is collimated into a parallel beam by means of a single lens located directly over the LED. Opposite the emitter is the integrated detector circuit. This IC contains of multiple sets of photo-detectors and the signal processing circuitry necessary to produce the digital waveforms.

The code-wheel rotates between the emitter and the detector, causing the light beam to be interrupted by the pattern of spaces and bars on the code-wheel. The photo-diodes which detect these interruptions are arranged in a pattern that corresponds to the radius and design of the code-wheel. These detectors are also spaced such that a light period on one pair of detectors corresponds to a dark period on the adjacent pair of detectors. The photo-diode outputs are then fed through the signal processing circuitry resulting in signals A and $\bar{A}$ and B and $\bar{B}$. Two comparators receive these signals and produce final outputs for channels A and B. Due to this integrated phasing

technique, the digital output of channel A is in quadrature with that of channel B (90^o out of phase). The encoder output can be used to find the direction of rotation, position and to count the number of rotations.

Specifications

Some of the important specifications of the HEDS-9100 encoder are presented in Table 2.1. The HEDS-series encoders provide sophisticated motion detection at a low cost, making them ideal for high volume applications. Typical applications include printers, plotters, tape drives and factory automation equipment.

Table 2.1: Specifications of the encoder

Supply voltage	5 V $\pm$ 10%
Rise time	200 ns
Fall time	50 ns
Number of channels	2
Counts per turn	500
Operating temperature	-20 to $+100$°C
MI of code wheel	$\leq 0.6\ gm^2$
Max. acceleration	250,000 $\frac{rad}{s^2}$
Max. output current per channel	5 mA
Phase shift	$90°e \pm 45°e$
Dimensions	41 mm $\times$ 30 mm $\times$ 18.3 mm
Weight	20 g

2.1.2 Rate-gyro

Operation of the muRata rate-gyro

The particular model considered was the muRata gyrostar ENV-05S. As illustrated in Figure 2.1, this device has a triangular metal prism fixed at two points. This prism is forced to vibrate by the piezoelectric ceramic at 7 kHz [Wei93]. With no rotation around **z** axis, each of the other two piezoelectric ceramics detect a large equal signal. When the prism is turned, it gets twisted such that the detectors receive different

signals. This signal difference is examined by internal analog circuits and brought out as a voltage proportional to the angular velocity ($\dot{\theta}$).

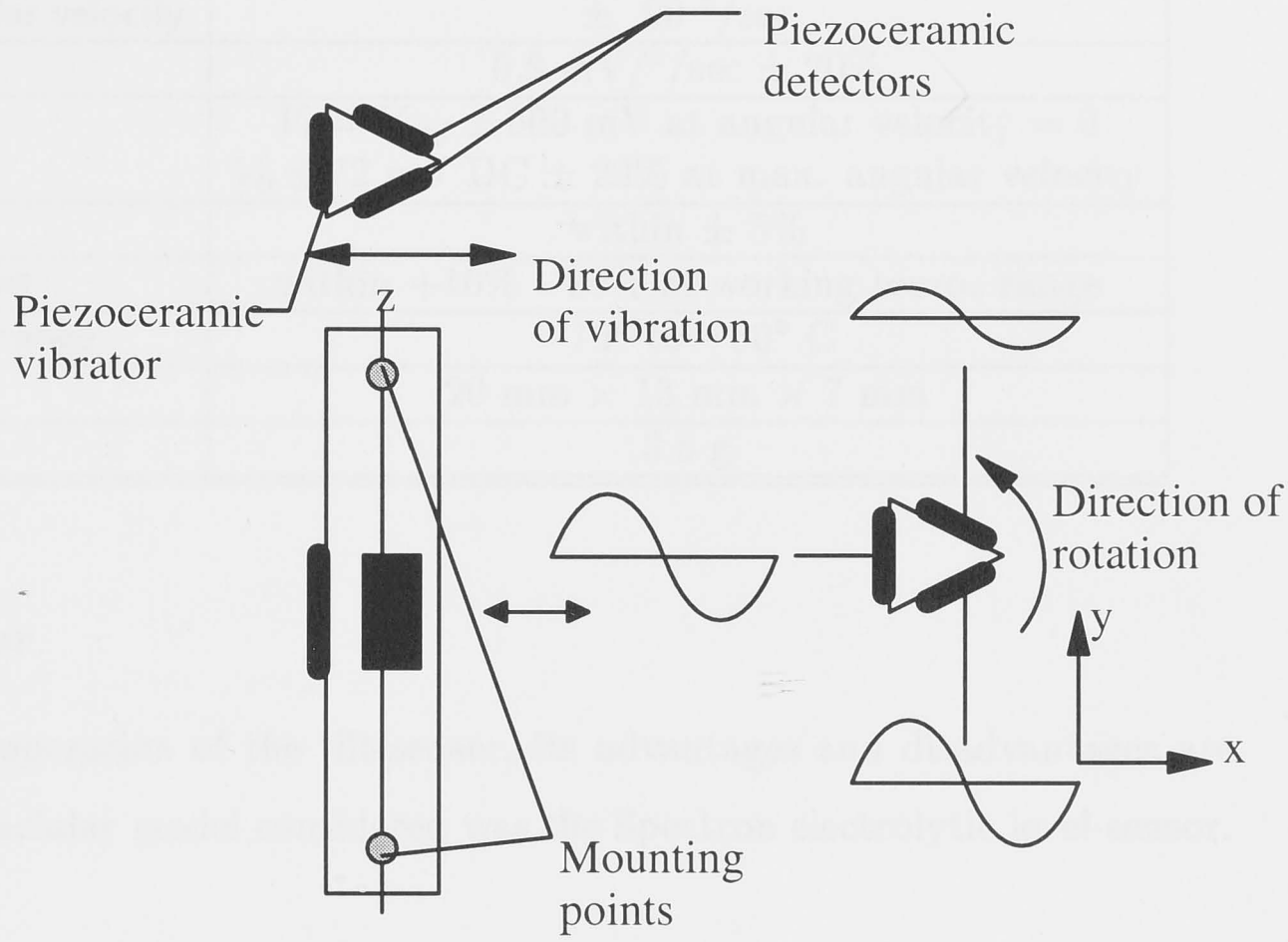


Figure 2.1: Rate-gyro

Specifications

The specifications of the rate-gyro are shown in Table 2.2. Some tests and compensation schemes have to be done for the successful use of this device. The following points are worth noting [Wei93]

i. Noise due to its own vibration at 7 kHz can be suppressed (see section 2.5.2).

ii. Over time, self-heating induces error in offset drift.

iii. Device is highly linear. At $21°C$, deviations are $< 0.05°/sec$ from ideal line.

iv. Scale factor is 22 mV / $°/sec$ $\pm$ 0.15% and offset is $0.5°/sec/\ °C$.

Table 2.2: Specifications of the rate-gyro

Supply voltage	5.0 V DC, 2.5 V DC (V_{ref}.)
Supply current	7 mA
Maximum angular velocity	$\pm$ 90 °/sec
Scale factor	0.8 mV/°/sec $\pm$ 20%
Output	$V_0 = V_{ref} \pm 500$ mV at angular velocity = 0 $V_0 \pm 72$ mV DC $\pm$ 20% at max. angular velocity
Linearity	within $\pm$ 5%
Scale factor offset	within +10% −20% at working temp. range
Working temp. range	−10° to +60° C
Dimensions	20 mm × 13 mm × 7 mm
Weight	3.5 g

2.1.3 Tilt-sensor

In this section, the operation of the tilt-sensor, its advantages and disadvantages are discussed. The partuclular model considered was the Spectron electrolytic level-sensor. (model# L211U)

Operation of the Spectron tilt-sensor

These tilt-sensors can be broadly classified into two major categories namely

i. High sensitivity sensor and
ii. Wide angle sensor.

A high sensitivity sensor resembles the glass vial in a carpenter's level. A wide angle sensor resembles a thick glass lens filled with a diameter of approximately one centimetre. They are both hermetically sealed, glass vials with three electrodes. Each glass vial is partially filled with an electrically-conductive fluid. When a sensor is at its zero (or null) position, the electrical resistance of the fluid from the centre electrode to each outside electrode is equal. Tilting the sensor disturbs this balanced condition, and the resistances change, in proportion to the angle of tilt. It should be recognised that the centre of gravity of the volume of fluid remains fixed, and the glass vial, with electrodes,

is being moved about it.

In electrical terms, the electrolytic tilt-sensor is a liquid potentiometer, with the wiper arm at the mid-point, when in level (horizontal) position. For the system designer, the electrolytic tilt-sensor may be termed as a single-axis sensor that indicates the direction of the total acceleration vector. A sensor will always respond to the acceleration of the local gravity vector and, in addition, respond to sustained and low-frequency ambient accelerations. In general, all respond to low-frequency inputs, some up to 10 Hz. Response to high-frequency inputs (vibration) is controlled by use of a unique, inverted T, internal cross-section. Spectron vertical sensing electrolytic potentiometer L-211U [STI93] with housing 557A is a single axis unit providing linear voltage output when the unit is tilted about the horizontal axis.

Specifications

Some of the important characteristics of the Spectron L-211U tilt-sensor are given in Table 2.3.

Table 2.3: Specifications of the Spectron electrolytic tilt-sensor

Tilt angle range	$\pm$ 60°
Output	7.2 $\pm$ 20% mV/°/volt excitation
Total null	2 mV at 3 V 400 Hz
Repeatability at any angle	0.03°
Null repeatability	0.03
Linearity	5% of full scale
Time constant	0.1 $\pm$ 25% seconds
Excitation voltage	0.5 to 5 V AC (20 to 20,000 Hz)
Operating temp. range	$-40°$ to $+80°$ C
Dimensions	14 mm $\times$ 13 mm $\times$ 13 mm
Weight	2 g

Tilt-sensor signal-conditioner

The tilt-sensor signal-conditioner is required for

i. Temperature compensation,

ii. Output gain adjustment for desired voltage,

iii. Sensor offset correction,

iv. Symmetry adjustment,

v. To develop the required AC sensor excitation.

Specifications

Some of the important specifications of this signal-conditioner (SA40012) are given in Table 2.4 [TA94].

Table 2.4: Specifications of the tilt-sensor signal-conditioner

Power requirement	5 V DC to 15 V DC
Supply current	0.5 mA at 5 V DC; 0.9 mA at 15 V DC
Sensor excitation	380 mV peak-peak square wave
Input impedance	5 Megohm.
Output signals	50 mV $< V_{out} < V_{DD}$ −1.5 V DC
Time constant	30 ms
Offset adjustment	± 45 mV DC
Output ripples	0.35% of output voltage
Temperature coefficient	0.2/°C at 1.0 V output
Operating temp. range	−25° to +70°C
Resistor gain adjustment	RG 10k $\rightarrow \infty$

2.1.4 Actuator

Brushed Motors

The motor considered for analysis was the Precision DC motor (model# RS718-981) [Mot92]. These motors are based on an iron-less rotor, combined with a commutation system using precious metals for the brush gear. The active rotor part simply consists of a cylindrical skew winding, requiring no support. Because of the absence of an iron core, rotor inertia is very low and there is no cogging. The rotor will stop in any position. The running speed is not limited by iron losses but depends only on the supply

voltage and load torque. The stator part consists of a cylindrical two-pole permanent magnet that fits inside a steel tube closing the magnetic circuit. This construction leads to distinct advantages in numerous applications, where high performance drive and servo systems are required.

Minimum friction and very small dynamic losses result in very high efficiency. For the electrical-to-mechanical energy transformation, only I^2R losses have to be considered. At breakaway, the only obstacle to overcome is friction of the brush-gear and the bearings. Due to the small dimensions of these parts, the rotor already starts moving with only fractions of a volt at its terminals, a major advantage in a position servo loop.

Advantages

i. High acceleration, excellent resistance to shocks and vibration and short mechanical time constant.
ii. Very high efficiency. The low losses (both electrical and mechanical) do not require extra power to be dissipated and do not increase temperature.
iii. Very low torque ripple, excellent dynamic balance and stability.
iv. Smooth and regular movement even at very low speed. No saturation, high peak torques without any risk of demagnetising the stator magnet.
v. Very low speed-dependent losses, high peak speeds, very low starting voltage and very little electrical noise during commutation.

Specifications

The important specifications of the motor (RS718-981) are shown in Table 2.5. For further details see [Mot92].

Table 2.5: Specifications of the actuator

Measuring voltage	36 V
Average no-load current	3 mA
No-load speed	7200 rpm
Back-emf constant	$4.9\ \frac{V}{1000}$ rpm
Rotor inductance	7 mH
Torque constant	$46.8\ \frac{mNm}{A}$
Rotor inertia	2×10^{-7} kgm^2
Mechanical time constant	20 ms

2.2 System Description

In this section, the equation of motion of the pendulum which incorporates the different sensor combinations is derived using Newton's second law of motion and the importance of the nonlinear analysis as opposed to linearisation is discussed.

2.2.1 Pendulum model

The dynamic equation of a pendulum relates forces and torques to positions, velocities and accelerations. That is, given forces and torques as input, this equation specifies the resulting motion of the pendulum. A single-link of a robot can be modelled as a pendulum with all the mass concentrated at a point. The torque applied to the shaft of the joint is a function of the inertia, the joint friction and gravity.

Derivation of the equation of motion

Consider the pendulum shown in Figure 2.2 which rotates about a pivot at one end where ℓ indicates the length of the massless rod and m the mass of the bob. Let θ denote the angle subtended by the rod and the vertical axis through the pivot point. The pendulum is free to swing in the vertical plane. The bob of the pendulum moves in a circle of radius ℓ. The pendulum is considered in the down position (corresponding to the equilibrium position). Figure 2.3 shows the stable and unstable equilibrium positions.

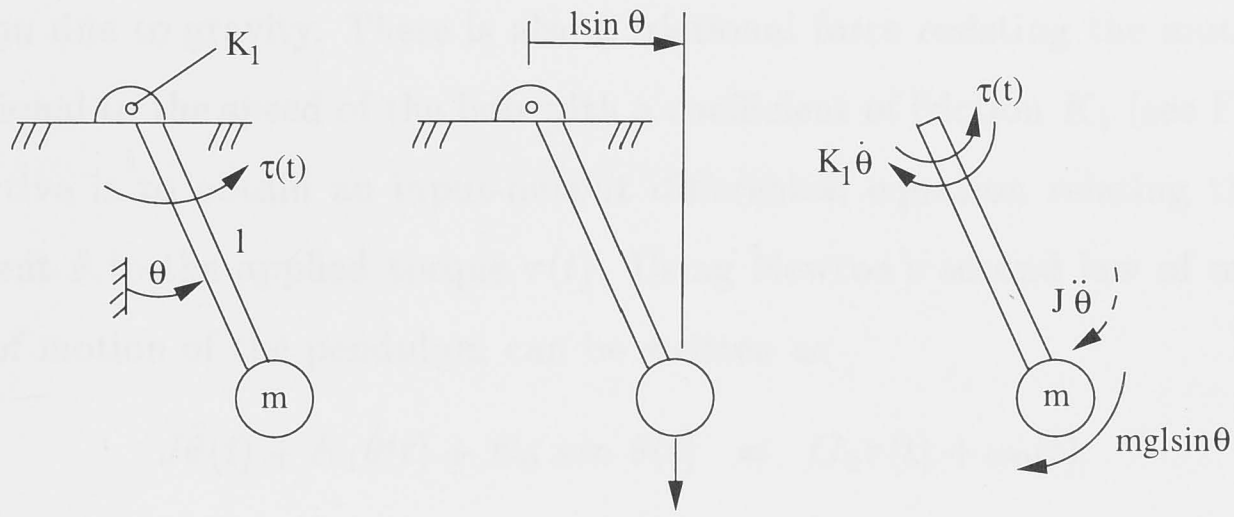


Figure 2.2: (a) Pendulum model (b) Partial diagram to determine the torque produced by the weight (c) Free-body diagram showing all torques about the pivot point

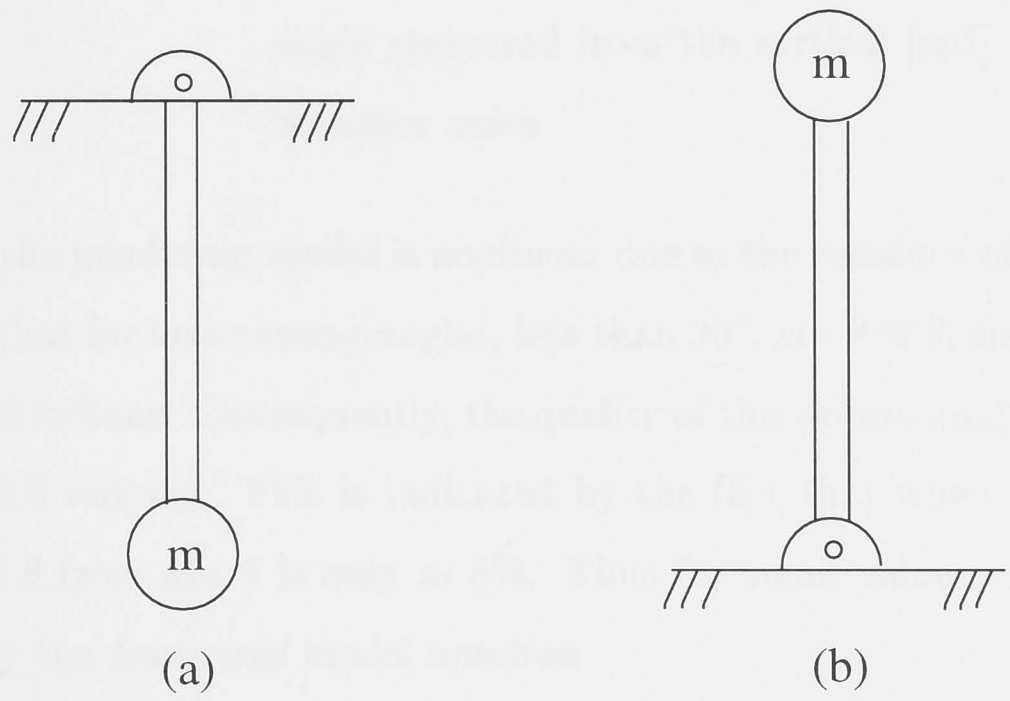


Figure 2.3: (a) Stable equilibrium position (b) Unstable equilibrium position

To write the equation of motion of the pendulum, first the forces acting on the bob are to be identified. There is a downward gravitational force equal to to mg where g is the acceleration due to gravity. There is also a frictional force resisting the motion, which is proportional to the speed of the bob with a coefficient of friction K_1 (see Figure 2.2). The objective is to obtain an input-output differential equation relating the angular displacement θ to the applied torque $\tau(t)$. Using Newton's second law of motion, the equation of motion of the pendulum can be written as

$$J\ddot{\theta}(t) + K_1\dot{\theta}(t) + K_0 \; sin \; \theta(t) \quad = \quad G_0\tau(t) + \omega_0(t) \tag{2.1}$$

where

J	inertia of the pendulum [kgm^2]
$K_1, \;\; K_0 = mg\ell, G_0 = K_0$	are constants associated with the pendulum
m	mass of the pendulum bob [kg]
g	acceleration due to gravity $[\frac{m}{sec^2}]$
ℓ	length of the pendulum [m]
τ	control torque [Nm]
θ	angle measured from the vertical [rad]
$\omega_0(t)$	actuator noise

As it is evident, the pendulum model is nonlinear due to the presence of the factor $sin \; \theta$. It can be noted that for lower swing-angles, less than 30°, $sin \; \theta \approx \theta$, since $sin \; 30° = 0.5$ and $30° = 0.5236$ radians. Consequently, the quality of this approximation is reasonably good for $|\theta| \leq 0.5$ radians. This is indicated by the fact that when $\theta = 0.5$ radians, the deviation of θ from $sin \; \theta$ is only $\approx 5\%$. Thus for small values of θ, (2.1) can be approximated by the *linearised model equation*

$$J\ddot{\theta}(t) + K_1\dot{\theta}(t) + K_0\theta(t) \quad = \quad G_0\tau(t) + \omega_0(t) \tag{2.2}$$

Thus, when nonlinearities are not severe, local linearisation may be employed to arrive at linear models which are approximations of the nonlinear equations in the neighbourhood of an operating point. Unfortunately, this pendulum model (like most problems

in robotics) is not well suited to this approach since the effect of nonlinearities becomes severe at larger swing-angles, and consequently the $sin\ \theta$ term must be included in the model description.

2.2.2 Pendulum model with different sensors

Case 1: Model with encoder only

The model and measurement equations and the corresponding state-space equations are given for the case when only the encoder is available for the measurements.

$$\begin{aligned} J\ddot{\theta}(t) + K_1\dot{\theta}(t) + K_0 sin\theta(t) &= G_0\tau(t) + \omega_0(t) \\ y_1(t_k) &= \theta(t_k) + v_0(t_k) \end{aligned} \tag{2.3}$$

where $\omega_0(t)$ is the process (actuator) noise and $v_0(t_k)$ is the sampled measurement (encoder) noise with appropriate covariances and t_k denotes the sampling time instant during which the measurements are obtained.

Let $x_1 \triangleq \theta$ and $x_2 \triangleq \dot{\theta}$ be the state variables. Then the state-space representation of the above equations is

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\left(\frac{K_0}{J}\right) sin(x_1) - \left(\frac{K_1}{J}\right) x_2 + \left(\frac{G_0}{J}\right)\tau(t) + \left(\frac{1}{J}\right)\omega_0(t) \\ y_1(t_k) &= x_1(t_k) + v_0(t_k) \end{aligned} \tag{2.4}$$

Case 2: Model with rate-gyro only

The model and measurement equations and the corresponding state-space equations are given for the case when only the rate-gyro is available for the measurements.

$$\begin{aligned} J\ddot{\theta}(t) + K_1\dot{\theta}(t) + K_0 sin\theta(t) &= G_0\tau(t) + \omega_0(t) \\ y_2(t_k) &= \dot{\theta}(t_k) + v_1(t_k) \end{aligned} \tag{2.5}$$

where $\omega_0(t)$ is the process (actuator) noise and $v_1(t_k)$ is the sampled measurement (rate-gyro) noise with appropriate covariances.

Let $x_1 \stackrel{\triangle}{=} \theta$ and $x_2 \stackrel{\triangle}{=} \dot{\theta}$ be the state variables. Then the state-space representation of the above equations is

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\left(\frac{K_0}{J}\right) sin(x_1) - \left(\frac{K_1}{J}\right) x_2 + \left(\frac{G_0}{J}\right) \tau(t) + \left(\frac{1}{J}\right) \omega_0(t) \\ y_2(t_k) &= x_2(t_k) + v_1(t_k) \end{aligned} \tag{2.6}$$

Case 3: Model with encoder and rate-gyro

The model equation for this case is

$$J\ddot{\theta}(t) + K_1\dot{\theta}(t) + K_0 sin\theta(t) = G_0\tau(t) + \omega_0(t) \tag{2.7}$$

and the measurement equations are

$$\begin{aligned} y_1(t_k) &= \theta(t_k) + v_0(t_k) \\ y_2(t_k) &= \dot{\theta}(t_k) + v_1(t_k) \end{aligned} \tag{2.8}$$

where $\omega_0(t)$ is the process (actuator) noise and $v_0(t_k)$ and $v_1(t_k)$ are the sampled measurement (encoder and rate-gyro) noises with appropriate covariances.

Let $x_1 \stackrel{\triangle}{=} \theta$ and $x_2 \stackrel{\triangle}{=} \dot{\theta}$ be the state variables. Then the state-space representation of the above equations is

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\left(\frac{K_0}{J}\right) sin(x_1) - \left(\frac{K_1}{J}\right) x_2 + \left(\frac{G_0}{J}\right) \tau(t) + \left(\frac{1}{J}\right) \omega_0(t) \\ y_1(t_k) &= x_1(t_k) + v_0(t_k) \\ y_2(t_k) &= x_2(t_k) + v_1(t_k) \end{aligned} \tag{2.9}$$

Case 4: Model with tilt-sensor only

When only the tilt-sensor is included for the measurements, due to the tilt-sensor dynamics, an additional state z is introduced and thus the order of the model becomes equal to three.

The model equations pertaining to this case are

$$\begin{aligned} J\ddot{\theta}(t) + K_1\dot{\theta}(t) + K_0 sin\theta(t) &= G_0\tau(t) + \omega_0(t) \\ \dot{z}(t) &= -\alpha_{ts}\ z(t) + \beta_{ts}\ \theta(t) \end{aligned} \tag{2.10}$$

and the measurement equation is

$$y_3(t_k) = z(t_k) + v_2(t_k) \tag{2.11}$$

where α_{ts} and β_{ts} are constants associated with the tilt-sensor and the subscript ts indicates the tilt-sensor. $\omega_0(t)$ is the process (actuator) noise and $v_2(t_k)$ is the sampled measurement (tilt-sensor) noise with appropriate covariances.

Let $x_1 \stackrel{\triangle}{=} \theta$, $x_2 \stackrel{\triangle}{=} \dot{\theta}$, $x_3 \stackrel{\triangle}{=} z$ be the state variables. Then the state-space representation is

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\left(\frac{K_0}{J}\right) sin(x_1) - \left(\frac{K_1}{J}\right) x_2 + \left(\frac{G_0}{J}\right)\tau(t) + \left(\frac{1}{J}\right)\omega_0(t) \\ \dot{x}_3 &= -\alpha_{ts}\ x_3 + \beta_{ts}\ x_1 \\ y_3(t_k) &= x_3(t_k) + v_2(t_k) \end{aligned} \tag{2.12}$$

Case 5: Model with tilt-sensor and rate-gyro

The model equations corresponding to this case are

$$\begin{aligned} J\ddot{\theta}(t) + K_1\dot{\theta}(t) + K_0 sin\theta(t) &= G_0\tau(t) + \omega_0(t) \\ \dot{z}(t) &= -\alpha_{ts}\ z(t) + \beta_{ts}\ \theta(t) \end{aligned} \tag{2.13}$$

and the measurement equations are

$$\begin{aligned} y_2(t_k) &= \dot{\theta}(t_k) + v_1(t_k) \\ y_3(t_k) &= z(t_k) + v_2(t_k) \end{aligned} \tag{2.14}$$

where $\omega_0(t)$ is the sampled process (actuator) noise and $v_1(t_k)$ and $v_2(t_k)$ are the sampled measurement (rate-gyro and tilt-sensor) noises with appropriate covariances.

Let $x_1 \stackrel{\triangle}{=} \theta$, $x_2 \stackrel{\triangle}{=} \dot{\theta}$, $x_3 \stackrel{\triangle}{=} z$ be the state variables. Then the state-space representation is

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\left(\frac{K_0}{J}\right) sin(x_1) - \left(\frac{K_1}{J}\right) x_2 + \left(\frac{G_0}{J}\right) \tau(t) + \left(\frac{1}{J}\right) \omega_0(t) \\ \dot{x}_3 &= -\alpha_{ts}\ x_3 + \beta_{ts}\ x_1 \\ y_2(t_k) &= x_2(t_k) + v_1(t_k) \\ y_3(t_k) &= x_3(t_k) + v_2(t_k) \end{aligned} \tag{2.15}$$

2.3 Parameter Estimation

A significant problem in systems theory is to determine the dynamics of a system from a history of its input and output measurements. After determining the structure of the dynamical systems, one must then estimate the various parameters in the particular structure. Choosing the proper representation of the system helps estimate the parameters. The equation of motion of the pendulum derived in section 2.2 provides such a representation.

The tilt-sensor model can be represented by a first order model and the pendulum by a second order linearised model for purposes of parameter estimation. A parameter estimation procedure is developed to identify the parameters associated with the pendulum and the tilt-sensor models. The estimation of the parameters is by using the *Least-Squares* (LS) algorithm. Figure 2.4 shows the combined pendulum and tilt-sensor

model and the associated parameters where PRBS indicates a Pseudo-Random Binary Sequence which is used as the input signal.

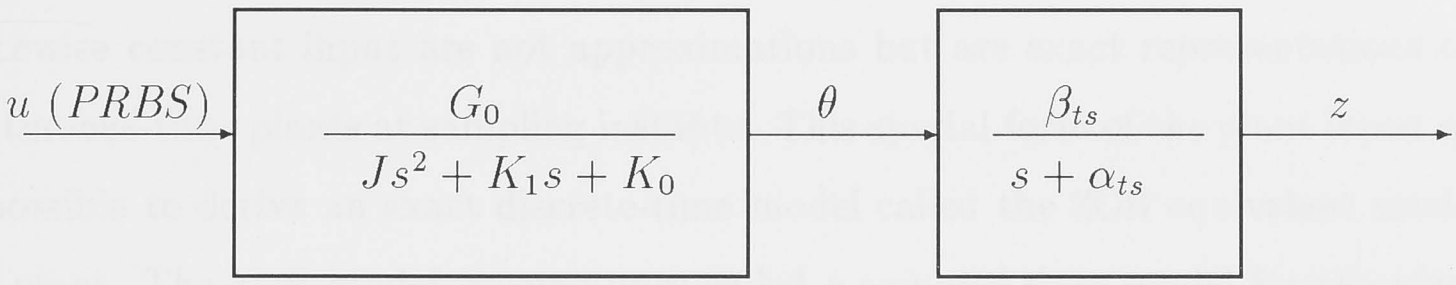


Figure 2.4: Transfer function representation of the pendulum and tilt-sensor model

Pseudo-random binary sequence

Pseudo-random sequences of various types have been extensively used for system identification purposes for many years. Among the specialised system test signals, one of the most widely used is the Pseudo-Random Binary Sequence (PRBS) [PM82]. As well as being easy to generate in a digital computer, PRBS is attractive for both software and hardware correlations as its binary nature removes the need for time consuming multiplications [NG69]. PRBS is a member of a broader class of odd-level maximal length sequences used for identification [DFC71].

2.3.1 Discretisation of continuous-time systems

Zero order hold

The process of converting digital samples to an analog signal that connects those samples is called *reconstruction* [Vac95]. A device that performs this reconstruction is called a digital-to-analog or D/A converter (DAC). The most common form of reconstruction used in practice is a *Zero-Order-Hold* (ZOH) reconstruction. A ZOH reconstruction of a sequence of samples produces a constant output value that is proportional to a given input sample for a fixed amount of time. Then the output changes to a new constant value that is proportional to the value of the next sample. The term zero order means that the sample points are interpolated by polynomials of degree zero (constants). D/A

converters which implement the zero-order-hold are most commonly used in practice.

The discrete-time models of linear, continuous-time plants that result based on a piecewise constant input are not approximations but are exact representations of the continuous-time plants at sampling instants. This special form of the plant input makes it possible to derive an exact discrete-time model called the ZOH equivalent model for the plant. The ZOH model is sometimes called a sampled-data model for the plant.

Discretisation using a ZOH

Consider a continuous time-invariant plant given by

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{b}u(t) \\ y(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{d}u(t) \end{aligned} \tag{2.16}$$

with state $\mathbf{x}(t) \in \mathcal{R}^n$, measured output $y(t) \in \mathcal{R}$ and the control input $u(t) \in \mathcal{R}$.

For digital control purposes, it is desired to define a discrete time index k such that $t = kT_s$ where T_s is the sampling interval. The piecewise constant input $u(t)$ is to be switched at times $t = kT_s$, $k = 0, 1, \cdots$. Then, the continuous plant input $u(t)$ is given by $u(t) = u(kT_s)$; $kT_s \leq t < (k+1)T_s$ where $u(t)$ is switched at times kT_s so that it is continuous from the right. An A/D device (ADC) generates the samples $y(kT_s)$ (of the output vector) and $\mathbf{x}(kT_s)$ (of the state vector). The ZOH sampling technique provides an *exact discretised equivalent* of the continuous plant for which discrete controls can be designed. For simplicity, the explicit dependence on T_s is omitted and we write,

$$x(k) \triangleq x(kT_s);\ u(k) \triangleq u(kT_s);\ y(k) \triangleq y(kT_s)$$

Variation of constants formula

The solution of (2.16) is given by [Wil91]

$$\mathbf{x}(t) = e^{\mathbf{A}(t-t_0)}\mathbf{x}(t_0) + \int_{t_0}^{t} e^{\mathbf{A}(t-t_0)}\ \mathbf{b}\ u(\sigma)\ d\sigma;\ t \geq t_0 \tag{2.17}$$

It is possible to use this solution to obtain the solution of the corresponding discrete-time system and it is desired to use this solution over one sample period to obtain a difference equation. Let $t = (k+1)T_s$ and $t_0 = kT_s$. Then the corresponding discrete-time version of (2.17) is

$$\mathbf{x}(k+1) = e^{\mathbf{A}T_s}\mathbf{x}(k) + \int_{kT_s}^{(k+1)T_s} e^{\mathbf{A}((k+1)T_s-\sigma)}\ \mathbf{b}\ u(\sigma)\ d\sigma; \tag{2.18}$$

This result is not dependent on the type of hold, since u is specified in terms of its continuous time history $u(t)$ over the sample interval. Then assuming a zero-order-hold with no delay, i.e.,

$$u(t) = u(kT_s),\ kT_s \leq \sigma < (k+1)T_s, \tag{2.19}$$

the corresponding discrete-time representation of (2.16) is then given by,

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{F}\mathbf{x}(k) + \mathbf{g}u(k) \\ y(k) &= \mathbf{H}\mathbf{x}(k) + \mathbf{j}u(k) \end{aligned} \tag{2.20}$$

where

$$\begin{aligned} \mathbf{F} &= e^{\mathbf{A}T_s};\ \mathbf{g} = \int_0^{T_s} e^{\mathbf{A}\tau}\ \mathbf{b}\ d\tau \\ \mathbf{H} &= \mathbf{C};\ \mathbf{j} = \mathbf{d}; \end{aligned} \tag{2.21}$$

It is important to note that the discretised plant matrix $\mathbf{F}$ is always nonsingular. Also note that $\mathbf{C}$ and $\mathbf{d}$ remain unchanged on discretisation.

2.3.2 Theory of least-squares

The (deterministic) input-output relationship of an ARX model can be described by a linear difference equation of the form

$$\begin{aligned} y(t) + a_1 y(t-1) + \cdots + a_{n_a} y(t-n_a) &= b_1 u(t-1) \\ &+ \cdots + b_{n_b} u(t-n_b) \end{aligned} \tag{2.22}$$

where AR refers to the autoregressive part and X to the extra input [Lju87]. Linear regression model structures are useful in describing basic linear and nonlinear systems.

The linear regression employs a predictor of the form

$$\hat{y}(t|\theta_{ls}) \quad = \quad \psi^T(t)\theta_{ls} \tag{2.23}$$

that is linear in θ_{ls} where the subscript ls refers to least-squares and ψ is the vector of regressors and is given by

$$\psi(t) \quad = \quad [-y(t-1) \; -y(t-2)\cdots \; -y(t-n_a) \; u(t-1)\cdots \; u(t-n_b)]^T$$

The prediction error can be written as

$$\epsilon(t,\theta_{ls}) \quad = \quad y(t) - \hat{y}(t|\theta_{ls}) = y(t) - \psi^T(t)\theta_{ls}$$

A batch of data is collected from the system whose parameters are to be identified. This information will be used in estimating the parameters of the system under consideration.

$$Z^N \quad = \quad [y(1), u(1), y(2), u(2), \; \cdots \; , y(N), u(N)]$$

and the Least-Squares (LS) criterion for the linear regression in (2.23) is given by the following expression

$$V_N(\theta_{ls}, Z^N) \quad = \quad \frac{1}{N}\sum_{t=1}^{N} \; \frac{1}{2}\Big[y(t) - \psi^T(t)\theta_{ls}\Big]^2 \tag{2.24}$$

Also a weighted criterion could be introduced to weight the measurement data according to the needs. One pragmatic approach is to replace the least-squares criterion of (2.24) with

$$V_N(\theta_{ls}, Z^N) \quad = \quad \frac{1}{N}\sum_{t=1}^{N} \; \lambda^{N-t} \; \frac{1}{2}\Big[y(t) - \psi^T(t)\theta_{ls}\Big]^2 \tag{2.25}$$

where λ is a parameter such that $0 < \lambda \leq 1$. The parameter λ is called forgetting factor or discounting factor. The loss function of (2.25) implies that a time-varying of the data is introduced. The most recent data is given unit weight, but data that is n time units old is weighted by λ^n. The method is called exponential forgetting or exponential discounting. A detailed treatment of the properties of the least-squares estimate (LSE) can be found in [AW89, Lju87].

2.3.3 Parameter estimation

a. Parameter estimation of the pendulum model

u $\rightarrow \theta$ algorithm

The theory for the estimation of the parameters of the pendulum model is developed in this section. The transfer function representation of the linearised pendulum model is shown in Figure 2.5 and the parameters to be estimated are K_1, J and G_0 ($K_0 = mg\ell$ is a constant).

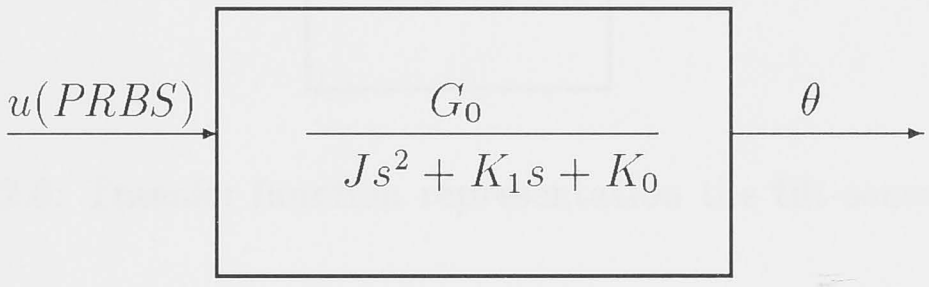


Figure 2.5: Transfer function representation of the linearised pendulum model

Consider the discretisation of the continuous-time system (2.16). For a small sampling interval, $e^{\mathbf{A}T_s} \approx \mathbf{I} + \mathbf{A}T_s$ where $\mathbf{I}$ is the identity matrix of appropriate order. The corresponding discrete-time relationship is given by (2.20) and the discretised system matrices in (2.21) by

$$\mathbf{F} = \mathbf{I} + \mathbf{A}T_s;\ \mathbf{g} = \mathbf{b}T_s;\ \mathbf{H} = \mathbf{C}; \tag{2.26}$$

The discrete transfer function is then given approximately by

$$\begin{aligned} H(z) &= \mathbf{H}(z\mathbf{I} - \mathbf{F})^{-1}\mathbf{g} \\ &= T_s\mathbf{C}\left[(z-1)\mathbf{I} - \mathbf{A}T_s\right]^{-1}\mathbf{b} \end{aligned}$$

$$H(z) = \frac{\frac{G_0 T_s^2}{J}}{z^2 + \left(\frac{K_1 T_s}{J} - 2\right) z + \left(1 - \frac{K_1 T_s}{J} + \frac{K_0 T_s^2}{J}\right)} \tag{2.27}$$

which in terms of (2.22) gives,

$$\begin{aligned} b_1 &= 0;\ b_0 = \frac{G_0 T_s^2}{J}; \\ a_1 &= \frac{K_1 T_s}{J} - 2;\ a_0 = 1 - \frac{K_1 T_s}{J} + \frac{K_0 T_s^2}{J}; \end{aligned} \tag{2.28}$$

b. Parameter estimation of the tilt-sensor model

u $\rightarrow$ z algorithm

The transfer function representation of the linearised pendulum model is shown in Figure 2.6. The parameters to be estimated are α_{ts} and β_{ts} with the knowledge of the already estimated pendulum parameters $\hat{K}_1$, $\hat{J}$, $\hat{G}_0$ and $\hat{K}_0$ where $\hat{}$ indicates the LSE.

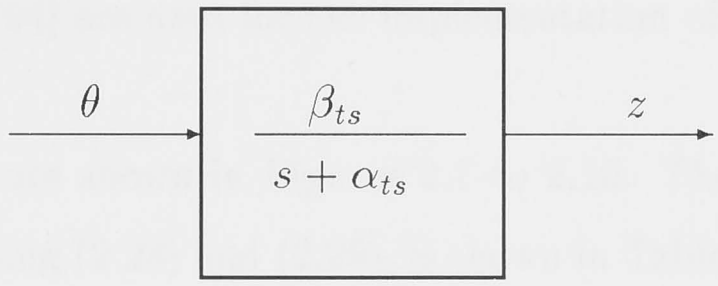


Figure 2.6: Transfer function representation the tilt-sensor model

The discrete transfer function for this case is

$$\begin{aligned} H(z) &= \mathbf{H}(z\mathbf{I}-\mathbf{F})^{-1}\mathbf{g} \\ &= T_s\mathbf{C}\left[(z-1)\mathbf{I}-\mathbf{A}T_s\right]^{-1}\mathbf{b} \\ &= \frac{pt_0}{z^3+pt_1z^2+pt_2z+pt_3} \end{aligned} \tag{2.29}$$

$$\text{where } \mathbf{A} = \begin{pmatrix} 0 & 1 & 0 \\ -\frac{K_0}{J} & -\frac{K_1}{J} & 0 \\ \beta_{ts} & 0 & -\alpha_{ts} \end{pmatrix};\ \mathbf{b} = \begin{pmatrix} 0 & \frac{G_0}{J} & 0 \end{pmatrix}^T;\ \mathbf{C} = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix};$$

$$\begin{aligned} \text{and } pt_0 &\triangleq \frac{G_0\beta_{ts}T_s^3}{J} \\ pt_1 &\triangleq \left(\frac{K_1T_s}{J}+\alpha_{ts}T_s-3\right) \\ pt_2 &\triangleq \left[3-2T_s\left(\frac{K_1}{J}+\alpha_{ts}\right)+\left(\frac{K_0+K_1\alpha_{ts}}{J}\right)T_s^2\right] \\ pt_3 &\triangleq \left[\frac{K_0\alpha_{ts}}{J}T_s^3-\left(\frac{K_0+K_1\alpha_{ts}}{J}\right)T_s^2+\left(\frac{K_1}{J}+\alpha_{ts}\right)T_s-1\right] \end{aligned}$$

From (2.28) and (2.29), the required parameters are determined.

2.3.4 Numerical results

In this section, the results of the parameter estimation of the pendulum and the tilt-sensor models are presented. A sampling interval of 0.02 is chosen. The original system is simulated to generate the data vector which serves as the measurement vector (Z) for purposes of simulation. The system identification and the control system toolboxes of MATLAB [Lju89, G$^+$94] are used for the implementation of the algorithm.

The simulation results are shown in Figures 2.7 to 2.10. The estimated value of the coefficients, obtained using (2.28) and (2.29), is shown in Table 2.6 where LSE indicates the Least Squares Estimate.

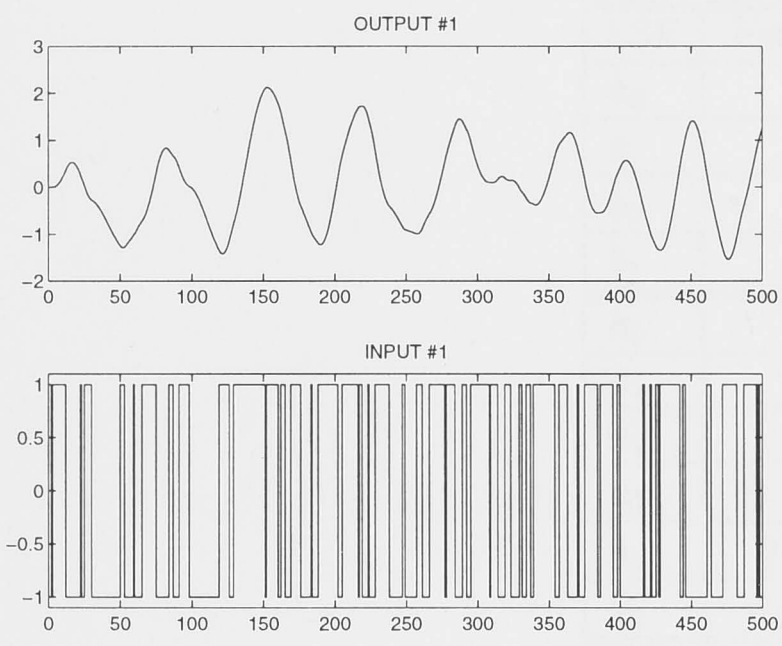


Figure 2.7: Output-input data (pendulum)

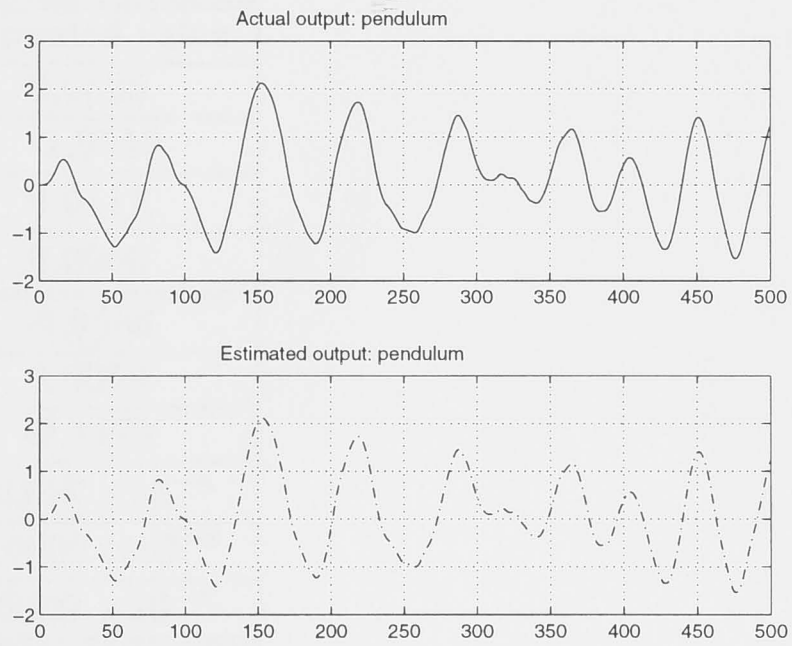


Figure 2.8: Actual and estimated outputs (pendulum)

From Table 2.6, $\hat{a}_0 = 0.9625$ and from (2.28)

$$a_0 = 1 - \frac{K_1 T_s}{J} + \frac{K_0 T_s^2}{J} \tag{2.30}$$

Note that the notation a_0 has been retained instead of $\hat{a}_0$ in (2.28) for notational simplicity with respect to equation (2.22), though it indicates the estimate. Equating (2.30) and its value, we have

$$0.9625 = 1 - \frac{K_1 T_s}{J} + \frac{K_0 T_s^2}{J}$$

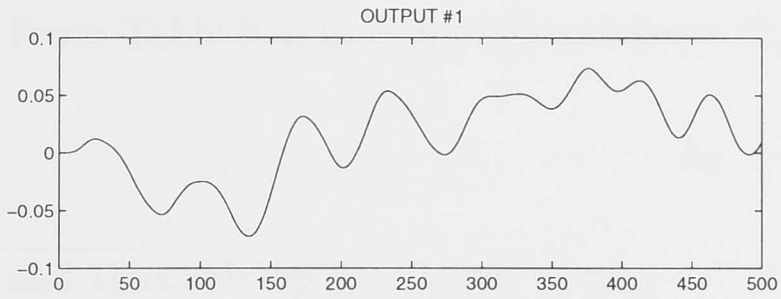


Figure 2.9: Output-input data (pendulum and tilt-sensor)

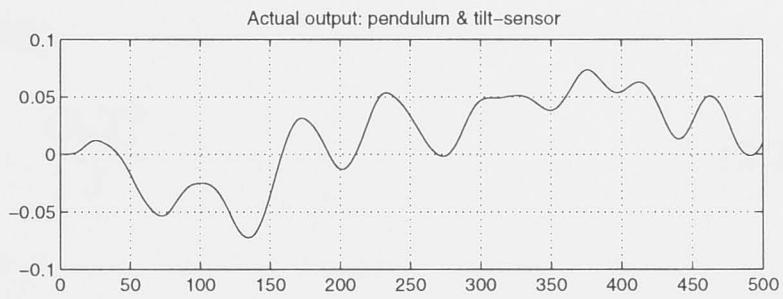
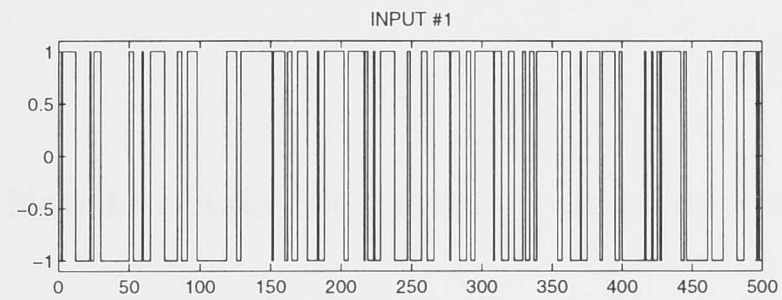


Figure 2.10: Actual and estimated outputs (pendulum and tilt-sensor)



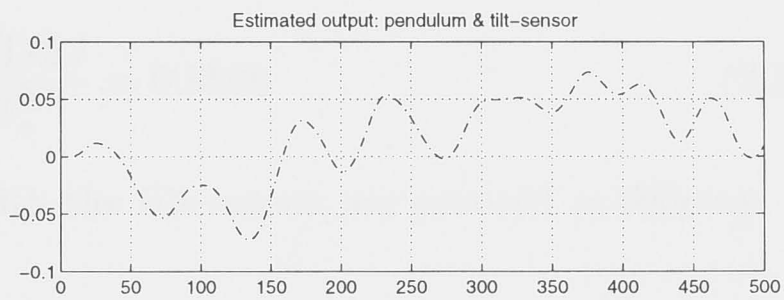


Table 2.6: Estimated coefficients

LSE	Estimated value
$\hat{a}_0$	0.9625
$\hat{a}_1$	-1.9512
$\hat{b}_0$	0.0112
$\hat{b}_1$	0.0000
$\hat{a}_0$	-0.9586
$\hat{a}_1$	2.9059
$\hat{a}_2$	-2.9472
$\hat{b}_0$	0.2248×10^{-4}
$\hat{b}_1$	-0.0000×10^{-4}
$\hat{b}_2$	0.0000×10^{-4}

Similarly, $\hat{a}_1 = -1.9512$ and from (2.28),

$$-1.9512 = \frac{K_1 T_s}{J} - 2 \tag{2.31}$$

Solving (2.30) and (2.31) for J, we obtain,

$$J = \frac{K_0 T_s^2}{0.0113} = 0.0306 \tag{2.32}$$

Substituting this value of J in (2.31), we arrive at

$$K_1 = \frac{(-1.9512 + 2)J}{T_s} = 0.0747 \tag{2.33}$$

From Table 2.6, $\hat{b}_0 = 0.0112$ and from (2.28)

$$b_0 = \frac{G_0 T_s^2}{J} \tag{2.34}$$

and thus solving for G_0, we obtain

$$G_0 = \frac{0.0112J}{T_s^2} = 0.8568 \tag{2.35}$$

Now to obtain the constants associated with the tilt-sensor, we proceed as follows:

Note that $pt_1 \triangleq \hat{a}_2$ and $pt_0 \triangleq \hat{b}_0$. From (2.29) and using the values of K_1 and J determined from (2.33) and (2.32) respectively,

$$\alpha_{ts} = \frac{pt_1 + 3 - \frac{K_1 T_s}{J}}{T_s} = 0.1988 \tag{2.36}$$

and

$$\beta_{ts} = \frac{pt_0 J}{G_0 T_s^3} = 0.1004 \tag{2.37}$$

Thus from (2.32), (2.33), (2.35), (2.36) and (2.37), the estimated parameter values are obtained and are tabulated in Table 2.7.

Table 2.7: Estimated parameters

Parameter	Description	Associated model	Value
J	inertia	pendulum	0.0306
K_1	constant	pendulum	0.0747
G_0	constant	pendulum	0.8568
α_{ts}	constant	tilt-sensor	0.1988
β_{ts}	constant	tilt-sensor	0.1004

2.4 ZOH Equivalence of Polynomic Nonlinear Systems

2.4.1 Mathematical preliminaries

a. Uniform pulse amplitude modulation

The discretisation procedure was detailed in section 2.3.1. Let the digital to analog conversion times be equally spaced. Then

$$u(t) \;=\; \sum_{k=-\infty}^{\infty} u\left(kT_s\right) p\left(t - kT_s\right) \tag{2.38}$$

where

$$p(t) = \begin{cases} 1 & \forall \;\; 0 \leq t < T_s \\ 0 & \text{otherwise} \end{cases} \tag{2.39}$$

This type of input signal is called the *Pulse Amplitude Modulated* (PAM) signal and along with the ZOH, it gives an *exact* discrete representation of the continuous-time system.

b. Vector homogenous forms

The concept of vector homogenous forms is a natural extension of tensor theory. The elements of the vector quantity $\mathbf{x}^{[p]}$ are homogenous forms of degree p of the elements of the vector $\mathbf{x} \in \mathcal{R}^n$. The necessary properties are introduced.

Definition 2.1 $\mathbf{x}^{[p]}$ *is defined as follows:*

$$\textit{Let } \mathbf{x} \;=\; \left(x_1 \; x_2 \; x_3 \; \cdots \; x_n\right)^T \tag{2.40}$$

Then $\mathbf{x}^{[p]}$ *denotes a lexicographical listing of the* N_p^n*-tuple in (2.41).*

The number of linearly independent p-degree ($p \geq 1$) [Mac46] terms in the n-variables $x_1, x_2, x_3, \cdots, x_n$ [Bro73] is given by the following definition:

Definition 2.2 *The number of linearly independent p-degree terms is*

$$N_p^n \;\equiv\; \frac{(n+p-1)\,!}{(n-1)\,!\;p\,!} \tag{2.41}$$

Definition 2.3 *The set of elements*

$$\sqrt{\binom{p}{p_1}\binom{p-p_1}{p_2}\cdots\binom{p-p_1-\cdots-p_{n-1}}{p_n}}\, x_1^{p_1} x_2^{p_2}\cdots x_n^{p_n} \qquad (2.42)$$

$$p_i \text{ nonnegative integers such that } \sum_{i=1}^{n} p_i = p$$

forms a basis for the set of p-forms of n-variables.

Property 2.1 *Although the lexicographic order is arbitrary the choice of normalising coefficients in (2.42) is such that*

$$\begin{aligned} ||\mathbf{x}||_2^{2p} &= ||\mathbf{x}^{[p]}||_2^2 \qquad (2.43)\\ \text{where } ||\mathbf{x}|| &\equiv \langle \mathbf{x}, \mathbf{x}\rangle^{\frac{1}{2}} \end{aligned}$$

c. Properties of $\mathbf{x}^{[p]}$

Property 2.2 *If*

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t), \text{ then}\\ \frac{d}{dt}\,\mathbf{x}^{[p]}(t) &= \mathbf{A}_{[p]}\mathbf{x}^{[p]}(t) \qquad (2.44) \end{aligned}$$

where

$$\mathbf{A}_{[p]} = \lim_{h\to 0}\frac{1}{h}\left[(\mathbf{I}+h\mathbf{A})^{[p]} - \mathbf{I}_{N_p^n}\right]$$

Note that $\mathbf{A}_{[p]}$ is also such that

$$\mathbf{A}_{[p]}\mathbf{x}^{[p]}(t) \triangleq \left(\frac{\partial \mathbf{x}^{[p]}}{\partial \mathbf{x}}\right)^T \mathbf{A}\mathbf{x}(t) \qquad (2.45)$$

where $\frac{\partial \mathbf{x}^{[p]}}{\partial \mathbf{x}}$ is an $n \times N_p^n$ matrix whose $(i,\ j)$-th element is $\frac{\partial \mathbf{x}_i^{[p]}}{\partial \mathbf{x}_j}$ and $\mathbf{x}_i{}^{[p]}$ is the i-th element of $\mathbf{x}^{[p]}$ and $\mathbf{x}_j$ is the j-th element of $\mathbf{x}$.

Property 2.3 *Let $\mathbf{\Phi}_\mathbf{A}(t)$ denote the transition matrix that satisfies the matrix differential equation*

$$\dot{\mathbf{\Phi}}(t) = \mathbf{A}\mathbf{\Phi}(t);\quad \mathbf{\Phi}_0 = \mathbf{I}_n$$

Then $[\mathbf{\Phi_A}(t)]^{[p]}$ *is the transition matrix of the differential equation*

$$\dot{\Psi}(t) \;=\; \mathbf{A}_{[p]}\Psi(t)$$

The exposition essentially follows that of [Bro73] although for the sake of completeness, a number of proofs are different for which the reader is referred to [San77].

d. Homogenous system

Consider a linear time invariant system

$$\dot{\mathbf{x}}(t) \;=\; \mathbf{A}\mathbf{x}(t) \tag{2.46}$$

where $\mathbf{x}(t) \in \mathcal{R}^n$, followed by a polynomial nonlinearity of the type as shown in Figure 2.11.

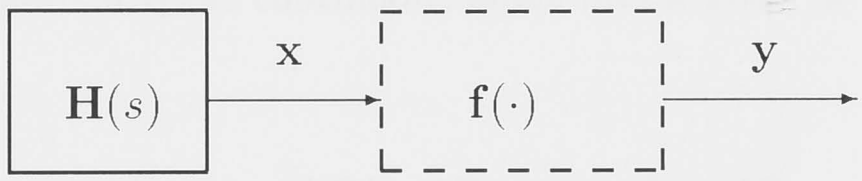


Figure 2.11: Homogenous polynomic nonlinear system

The nonlinearity is of the form

$$\mathbf{f}(\cdot) \;=\; \sum_{p=1}^{m} \beta_p \mathbf{x}^{[p]} \tag{2.47}$$

$$\mathbf{y} \;=\; \beta_p \left[\mathbf{C}^T\mathbf{x}\right]^{[p]} \tag{2.48}$$

where $\mathbf{y}(t) \in \mathcal{R}^m$ and $\mathbf{C}$ is the output matrix. Then

$$\mathbf{x}^{[p]} \;=\; \mathbf{A}_{[p]}\,\mathbf{x}^{[p]};\; p = 1, 2, \cdots m;$$

$$\mathbf{y} \;=\; \beta_p\,\mathbf{C}^{[p]^T}\mathbf{x}^{[p]}$$

e. Non-homogenous system

Consider a linear time invariant system with the forcing term $u(t)$

$$\dot{\mathbf{x}}(t) \;=\; \mathbf{A}\mathbf{x}(t) + \mathbf{b}u(t) \tag{2.49}$$

$$\dot{u}(t) \;=\; 0 \tag{2.50}$$

followed by a polynomial nonlinearity of the type as shown in Figure 2.12 and as in (2.47).

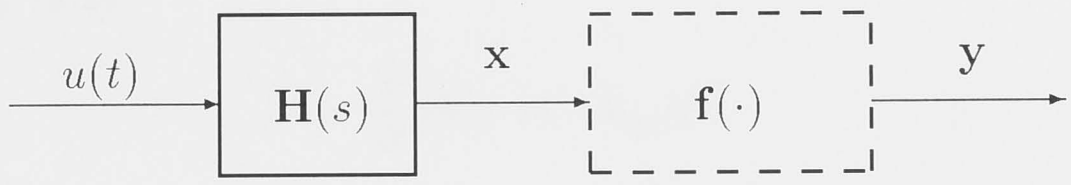


Figure 2.12: Non-homogenous polynomic nonlinear system

where $\mathbf{x}(t) \in \mathcal{R}^n$, $\mathbf{y}(t) \in \mathcal{R}^m$, $u(t) \in \mathcal{R}^\ell$ and $H(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{b}$ and $\mathbf{C}$ is the output matrix.

As described in section 2.3.1, the continuous input $u(t)$ is given in terms of the discrete input $u(kT_s)$ by

$$u(t) \quad = \quad u(kT_s); \quad kT_s \leq t < (k+1)T_s \tag{2.51}$$

where $u(t)$ is switched at discrete time instants kT_s.

Writing the augmented system,

$$\begin{pmatrix} \dot{\mathbf{x}} \\ \dot{u} \end{pmatrix} \quad = \quad \begin{pmatrix} \mathbf{A} & \mathbf{b} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ u \end{pmatrix}; \quad kT_s \leq t < (k+1)T_s \tag{2.52}$$

subject to the initial conditions $\begin{pmatrix} \mathbf{x}(k) \\ u(k) \end{pmatrix}$.

(2.52) is of the form

$$\dot{\tilde{\mathbf{x}}} \quad = \quad \tilde{\mathbf{A}}\tilde{\mathbf{x}} \tag{2.53}$$

where

$$\tilde{\mathbf{x}} \quad \stackrel{\triangle}{=} \quad \begin{pmatrix} \mathbf{x} \\ u \end{pmatrix}$$

$$\tilde{\mathbf{A}} \triangleq \begin{pmatrix} \mathbf{A} & \mathbf{b} \\ 0 & 0 \end{pmatrix}$$

From property (2.2),

$$\frac{d}{dt}(\tilde{\mathbf{x}}) = \tilde{\mathbf{A}}_{[p]}\tilde{\mathbf{x}}^{[p]}$$

Assuming $\mathbf{A}^{-1}$ exists, the discrete input-output relationship is given by

$$\begin{aligned} \mathbf{y}(k) &= \mathbf{C}^T\mathbf{x}(k) + \mathbf{C}^{[p]^T}\tilde{\mathbf{x}}^{[p]}(k) \text{ where} \\ \mathbf{x}(k+1) &= e^{\mathbf{A}T_s}\mathbf{x}(k) + u(k)\mathbf{A}^{-1}\left[e^{\mathbf{A}T_s} - \mathbf{I}_n\right]\mathbf{b} \\ \tilde{\mathbf{x}}^{[p]}(k+1) &= e^{\tilde{\mathbf{A}}_{[p]}T_s}\tilde{\mathbf{x}}^{[p]}(k) \end{aligned} \tag{2.54}$$

where $\mathbf{I}_n$ is the identity matrix of order n and T_s is the sampling interval.

2.4.2 Input-output representation via Volterra series

Volterra series

Considerable effort has been expended in attempts to find exact solutions for nonlinear feedback systems excited by random inputs [Van79]. Many physical systems exhibit a nonlinear behaviour that requires a more accurate analysis than that afforded by the linearisation of the system model. The Volterra series provides a powerful way of portraying the input-output relationship of a nonlinear model which has the following advantages:

i. All input-output relationships are explicit thus eliminating the need for solving complicated (nonlinear) differential equations.

ii. It facilitates easy and straightforward analysis. Random and deterministic inputs and disturbances can be included.

Functional representations of single-valued nonlinear functions were first introduced by Volterra and they provide a generalised non-parametric method of expressing the response of a nonlinear system [Geo59]. The Volterra series representation to be obtained is not only an explicit nonlinear representation of the system response but also

affords insight into the system operation. Contributions to the theory of input-output models for nonlinear systems, from a Volterra-type view-point, have been made by [Bro76, Gil78] among others. For further details, see [Rug81] in addition to the cited references. A Volterra model can describe nonlinear behaviour such as assymetric output changes in response to symmetric changes in the input. A controller based on this model can yield improved performance over a linear model-based controller [M$^+$96].

a. Pendulum case

Using the power series expansion of $sin\ \theta$, (2.1) becomes,

$$\begin{aligned} \ddot{\theta}(t) + \left(\frac{K_1}{J}\right)\dot{\theta}(t) + \left(\frac{K_0}{J}\right)\left(\theta(t) - \frac{\theta^3(t)}{6} + \frac{\theta^5(t)}{120}\right) &= \left(\frac{G_0}{J}\right)u(t) \qquad (2.55) \\ &+ \left(\frac{1}{J}\right)\omega_0(t) \end{aligned}$$

where $u(t) \equiv \tau(t)$ is used for notational simplicity.

Considering (2.1) and writing a Volterra series expansion as in [Van79],

$$\theta(t) = \theta_1(t) + \theta_2(t) + \cdots = \sum_{i=1}^{\infty} \theta_i(t) \qquad (2.56)$$

where $\theta_1(t)$ is defined as the output of the linear system, $\theta_2(t)$ as the output of the second-order system, and so forth. Substituting (2.56) into (2.1), expanding $sin\ \theta(t)$ and equating the equal-order terms of $\theta(t)$ with the relevant substitutions leads to

$$\begin{aligned} &\left[\ddot{\theta}_1(t) + \ddot{\theta}_2(t) + \ddot{\theta}_3(t) + \cdots\right] + \left(\frac{K_1}{J}\right)\left[\dot{\theta}_1(t) + \dot{\theta}_2(t) + \dot{\theta}_3(t) + \cdots\right] \\ &+ \left(\frac{K_0}{J}\right)[\theta_1(t) + \theta_2(t) + \theta_3(t) + \cdots] - \left(\frac{K_0}{3!J}\right)[\theta_1(t) + \theta_2(t) + \theta_3(t) + \cdots]^3 \\ &+ \left(\frac{K_0}{5!J}\right)[\theta_1(t) + \theta_2(t) + \theta_3(t) + \cdots]^5 = \left(\frac{G_0}{J}\right)u(t) + \left(\frac{1}{J}\right)\omega_0(t) \end{aligned}$$

Equating terms of the same order,

$$\begin{aligned} \ddot{\theta}_1(t) + \left(\frac{K_1}{J}\right)\dot{\theta}_1(t) + \left(\frac{K_0}{J}\right)\theta_1(t) &= \left(\frac{G_0}{J}\right)u(t) + \left(\frac{1}{J}\right)\omega_0(t) \\ \ddot{\theta}_2(t) + \left(\frac{K_1}{J}\right)\dot{\theta}_2(t) + \left(\frac{K_0}{J}\right)\theta_2(t) &= 0 \end{aligned}$$

$$\begin{aligned}
\ddot{\theta}_3(t) + \left(\frac{K_1}{J}\right)\dot{\theta}_3(t) + \left(\frac{K_0}{J}\right)\theta_3(t) &= \left(\frac{K_0}{3!J}\right)\theta_1^3(t) \\
\ddot{\theta}_4(t) + \left(\frac{K_1}{J}\right)\dot{\theta}_4(t) + \left(\frac{K_0}{J}\right)\theta_4(t) &= 0 \\
\ddot{\theta}_5(t) + \left(\frac{K_1}{J}\right)\dot{\theta}_5(t) + \left(\frac{K_0}{J}\right)\theta_5(t) &= \left(-\frac{K_0}{5!J}\right)\theta_1^5(t) + \left(\frac{K_0}{3!J}\right)3\theta_1^2(t)\theta_3(t)
\end{aligned}$$

All higher order even terms are equal to zero. The closed form expressions for the linear and the non-linear (cubic) part with the additive measurement noise up to the third order are

$$\ddot{\theta}_1(t) = -\left(\frac{K_0}{J}\right)\theta_1(t) - \left(\frac{K_1}{J}\right)\dot{\theta}_1(t) + \left(\frac{G_0}{J}\right)u(t) + \left(\frac{1}{J}\right)\omega_0(t) \tag{2.57}$$

$$\ddot{\theta}_3(t) = -\left(\frac{K_0}{J}\right)\theta_3(t) - \left(\frac{K_1}{J}\right)\dot{\theta}_3(t) + \left(\frac{K_0}{3!J}\right)\theta_1^3(t) \tag{2.58}$$

$$\theta(t) \approx \theta_1(t) + \theta_3(t) \tag{2.59}$$

$$\dot{\theta}(t) \approx \dot{\theta}_1(t) + \dot{\theta}_3(t)$$

This representation is shown in Figure 2.13.

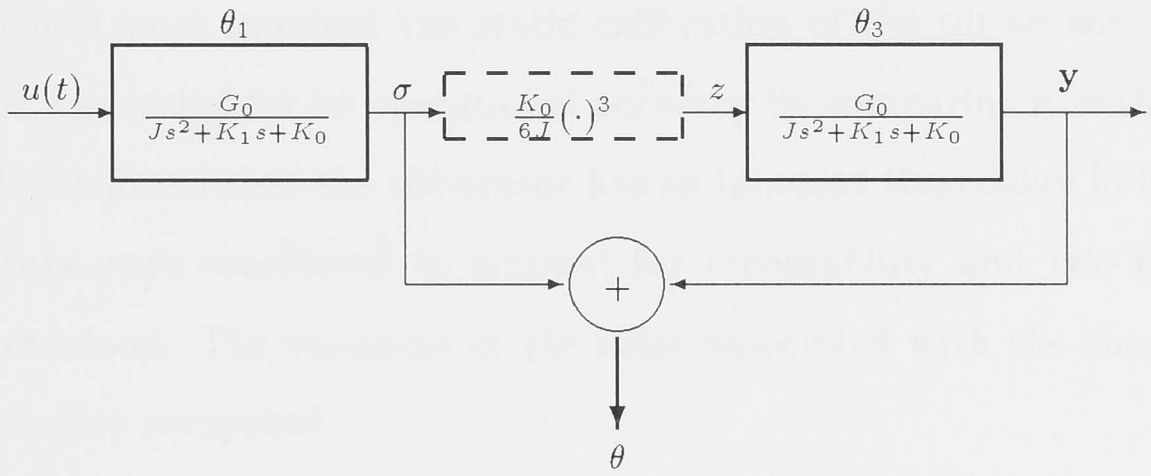


Figure 2.13: Volterra series representation of the pendulum model

b. Pendulum and tilt-sensor case

The model equation of the tilt-sensor is given by

$$\dot{z}(t) = -\alpha_{ts}\, z(t) + \beta_{ts}\, \theta(t)$$

Writing $z(t)$ as

$$z(t) = z_1(t) + z_2(t) + \cdots = \sum_{i=1}^{\infty} z_i(t) \tag{2.60}$$

Substituting (2.60) into the model equation of the tilt-sensor yields,

$$\begin{aligned}[\dot{z}_1(t) + \dot{z}_2(t) + \dot{z}_3(t) + \cdots] &= -\alpha_{ts}\,[z_1(t) + z_2(t) + z_3(t) + \cdots] \\ &+ \beta_{ts}\,[\theta_1(t) + \theta_2(t) + \theta_3(t) + \cdots]\end{aligned}$$

Proceeding as in the pendulum case, the expressions for the linear and the non-linear (cubic) part with the additive measurement noise up to the third order are

$$\begin{aligned}\ddot{\theta}_1(t) &= -\left(\frac{K_0}{J}\right)\theta_1(t) - \left(\frac{K_1}{J}\right)\dot{\theta}_1(t) + \left(\frac{G_0}{J}\right)u(t) + \left(\frac{1}{J}\right)\omega_0(t) \\ \dot{z}_1(t) &= -\alpha_{ts}\; z_1(t) + \beta_{ts}\; \theta_1(t) \\ \ddot{\theta}_3(t) &= -\left(\frac{K_0}{J}\right)\theta_3(t) - \left(\frac{K_1}{J}\right)\dot{\theta}_3(t) + \left(\frac{K_0}{3!J}\right)\theta_1^3(t) \\ \dot{z}_3(t) &= -\alpha_{ts}\; z_3(t) + \beta_{ts}\; \theta_3(t) \\ z(t) &\approx z_1(t) + z_3(t) \qquad (2.61)\end{aligned}$$

2.5 Sensor and Actuator Experiments

The experimental work required the static calibration of the tilt-sensor. In addition, the tilt-sensor was tested for its operational accuracy by comparing it with the encoder operation. It was found that the tilt-sensor has an inherent time delay in its operation. Two tilt-sensors were considered to account for repeatability and two runs for each tilt-sensor were done. The variances of the noise associated with the encoder and the rate-gyro were also computed.

Experimentation environment

The setup that is used for the experiments exists in the automated systems laboratory. The experimental data is collected using the real-time *Vx-Works* operating system which differs from a standard UNIX operating system in that the *Vx-Works* is a real-time operating system. Compiled programs are downloaded to a Motorola 68040 microprocessor which is located on aVME board. The pendulum and the system setups are shown in Figures 2.14 and 2.15 respectively.

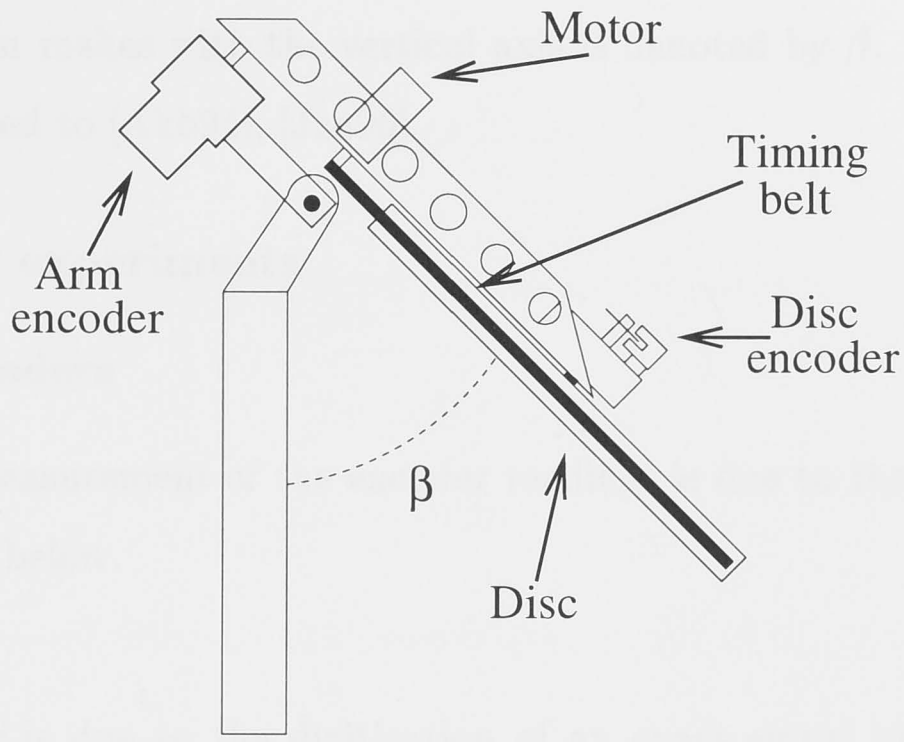


Figure 2.14: Side-view of the pendulum setup

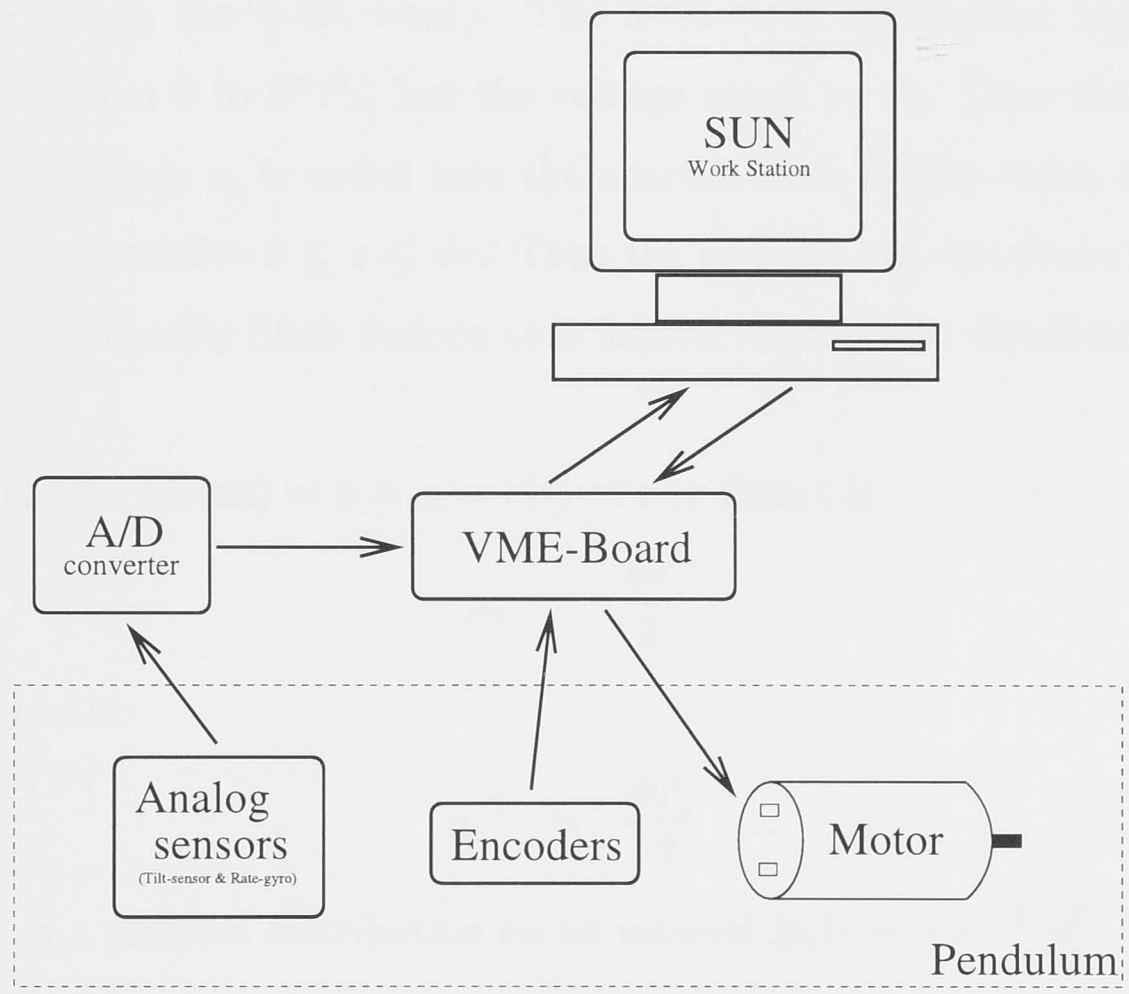


Figure 2.15: Experimental setup

The pendulum consists of an arm that can swing in a circle of radius r. On one end of the arm is a disc that can be rotated by a motor through a timing-belt. By accelerating the disc, a torque is imparted to the pendulum-arm which enables it to swing in a circular fashion and eventually reach the inverted (balanced) position. The

angle the pendulum makes with the vertical axis is denoted by β. For further details, the reader is referred to [Akb94], [Jen96].

2.5.1 Encoder experiments

Error in the encoders

The error in the measurement of the encoder readings is due to the quantisation noise which is described below.

Quantisation noise

Quantisation noise is due to the digitisation of an exact signal value $v_t = v(t)$ captured at sampling time T_s by an A/D converter [Dor93]. The binary representation is $b_{n-1}b_{n-2} \cdots b_1b_0$ (an n-bit word). The n-bit word digitisation has 2^n different values possible, from 0 to 2^{n-1}. Let the voltage range be R. Then the *resolution* is $dv = \frac{R}{2^n}$. Any voltage v_t is coded into the nearest lower binary value x_b, where the error $e = x_t - x_b$ satisfies $0 \leq e \leq dv$. Thus the errors e are distributed over the interval $[0, dv]$ in an equally likely fashion that implies the uniform distribution on $[0, dv]$.

The expected value (mean) of $e = e_t = e(t)$ at any time t is

$$\mu_e = \frac{dv}{2}$$

and the variance is

$$\mu_e{}^2 = \frac{dv^2}{12}$$

(the variance of a uniform distribution on an interval [a,b] is $\sigma = \frac{(b-a)^2}{12}$).

Calculation of error

The encoder quantisation for 360° can be written as

$$\Delta = \frac{360°}{resolution} \tag{2.62}$$

Two possibilities for two's complement arithmetic are *roundoff* and *truncation*. Ignoring the LSB (least significant bit) in a two's complement representation is equivalent

to implementing truncation [Wil91]. It is observed that the HEDS-series encoders use truncation. Thus the error lies in the range $[0, \Delta]$.

Furthermore, the mean and the variance are calculated using the relations,

$$\text{mean } (\mu) = \frac{\Delta}{2} \tag{2.63}$$

$$\text{variance } (\sigma^2) = \frac{\Delta^2}{12} \tag{2.64}$$

Thus the variance of different encoder models are calculated and are tabulated in Table 2.8.

Table 2.8: Encoder variances

Encoder model #	Resolution *per* 360°	Quantisation error	Mean	Variance
HEDS 9100A	2000	0.1800	0.0900	0.0027
HEDS 9100G	1440	0.2500	0.1250	0.0052
HEDS 550K	384	0.9375	0.4688	0.0732
HEDS 550C	400	0.9000	0.4500	0.0675
HEDS 550D	768	0.4688	0.2344	0.0183
HEDS 550E	800	0.4500	0.2250	0.0169
HEDS 550F	1024	0.3516	0.1758	0.0103
HEDS 550H	1600	0.2250	0.1125	0.0042
HEDS 550I	2048	0.1758	0.0879	0.0026

2.5.2 Rate-gyro experiments

Error in the rate-gyros

From Table 2.2 in section 2.1.2, it can be seen that the error in the measurement of the angular velocity by the gyro is introduced in the following ways:

i. Linearity : $\pm$ 2 mV

ii. Temperature offset : $\pm$ 2 mV

iii. Drift : $\pm$ 2 mV

iv. Output noise : Within 10 mV RMS

The error given above in mV translates to $0.27°/sec$ due to linearity, temperature offset and drift. But the error due to output noise is $10 * \sqrt{2} = 14.1$ mV. The full scale voltage output corresponds to 2.5 V DC and the maximum angular velocity is $\pm\ 90^o$. Thus the error due to the output noise is $\left(\frac{14}{2500}\right) \times\ 90 = 0.5040°/sec$. Therefore it can be seen that the error due to the output noise contributes the most towards the error and thus has to be accounted for.

Gaussian distribution

Gaussian distribution, or Normal distribution, is probably the most extensively used probability distribution in engineering applications. Apart from its simplicity, another justification for its widespread use is provided by the Central Limit Theorem.

Theorem 2.1 Central Limit Theorem

A random variable that is formed by summing a very large number of independent random variables takes Gaussian distribution in the limit.

Since many engineering phenomena are consequences of numerous independent random cases, the assumption of normal distribution is justified in many cases.

The Gaussian probability density function is given by

$$f(x) \quad = \quad \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]$$

Note that only two parameters, *mean* μ and *standard deviation* σ are required to determine a Gaussian distribution completely. A closed-form algebraic expression cannot be given for the cumulative probability distribution function $F(x)$ of a Gaussian distribution. It should be evaluated by numerical integration. Numerical values for the normal distribution curve are available in tabulated form, with the random variable X being normalised with respect to μ and σ according to

$$Z \quad = \quad \frac{X-\mu}{\sigma}$$

Furthermore, the probability density function of Z is

$$f(z) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{z^2}{2}\right)$$

What is usually tabulated is the area under the density curve $f(z)$ of the normalised random variable Z for different values of z. The area under the curve is given by

$$A = \int_0^z f(z)\, dz$$

A table where the area under the $f(z)$ curve of the normalised random variable Z, for different values of z, from 0 to z is available. Since the density curve is symmetric about the mean value (zero for the normalised case), values for negative z do not have to be tabulated. In Figure 2.16, a Gaussian curve with zero mean and unity variance is shown.

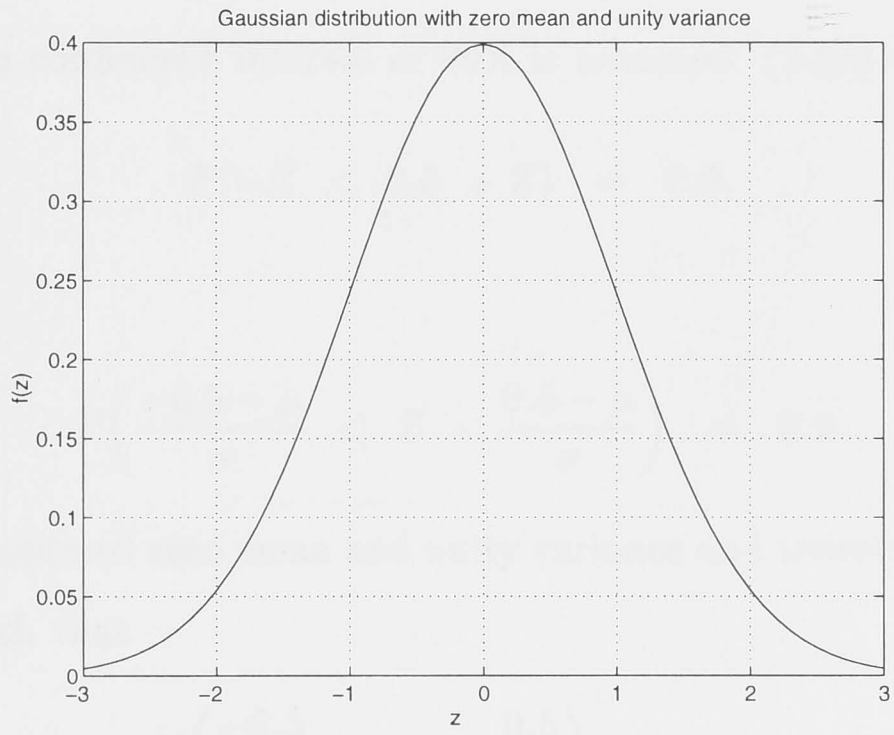


Figure 2.16: Gaussian distribution

Confidence intervals

The probability that the value of a random variable would fall within a specified interval is called a confidence level. Considering a Gaussian random variable X that has mean μ and standard deviation σ, it is denoted by

$$X = N(\mu, \sigma^2)$$

This is a random variable with zero mean and unity standard deviation. The probability P that the values of Z fall within $\pm z_0$ is

$$P(-z_0 < Z < z_0) \quad = \quad p$$

Estimation of variance of the noise

As noted earlier, the output noise is the major source of the error in the measurements of the rate-gyro. Consider a Gaussian distribution with zero mean and unity variance (and thus unity standard deviation is considered). The variance of the noise associated with the rate-gyro is calculated such that

$$P(\mid Z \mid \leq \ 0.5) \quad = \quad 0.9 \tag{2.65}$$

which means that a confidence interval of 90% is assumed. (2.65) can be rewritten as

$$P(-Z \ < \ 0.5 \ + Z) \quad = \quad 0.9$$

and thus

$$P\left(\frac{-0.5-\mu}{\sigma} \ < \ Z \ < \frac{0.5-\mu}{\sigma}\right) \quad = \quad 0.9$$

But this is for the assumed zero mean and unity variance and therefore the new variance has to be found such that

$$P\left(\frac{-0.5}{\sigma} \ < \ Z \ < \frac{0.5}{\sigma}\right) \quad = \quad 0.9 \tag{2.66}$$

For zero mean and unity variance, the area under the z curve is found to be 1.645. But for an unknown variance subject to (2.66), the standard deviation $\sigma_{new} \ = \ \frac{0.5}{1.645} \times 1 \ =$ 0.3040. Thus the variance of the noise associated with the rate-gyro, V_{rg} is found to be

Result 2.1

$$V_{rg} \quad = \quad \sigma^2_{new} = 0.0924. \tag{2.67}$$

2.5.3 Tilt-sensor experiments

Experiments were done with the existing set-up in the automated systems laboratory using the real time *Vx-Works* environment and the data collection was done along with the arm encoder readings (As it will be seen, this will facilitate the comparison of the relative merits and demerits of using one instead of the other).

The tilt-sensor uses a tilt-sensor signal conditioner as detailed in section 2.1.3. First the experiments were done with the gain adjustment, sensor offset correction and the symmetry adjustment kept in an arbitrary position and the results were recorded and plotted. The results were recorded for a tilt-angle range of $-90°$ to $+90°$ and the corresponding average output voltage of the tilt-sensor, peak-peak noise amplitude and variance corresponding to a particular angle were calculated and tabulated.

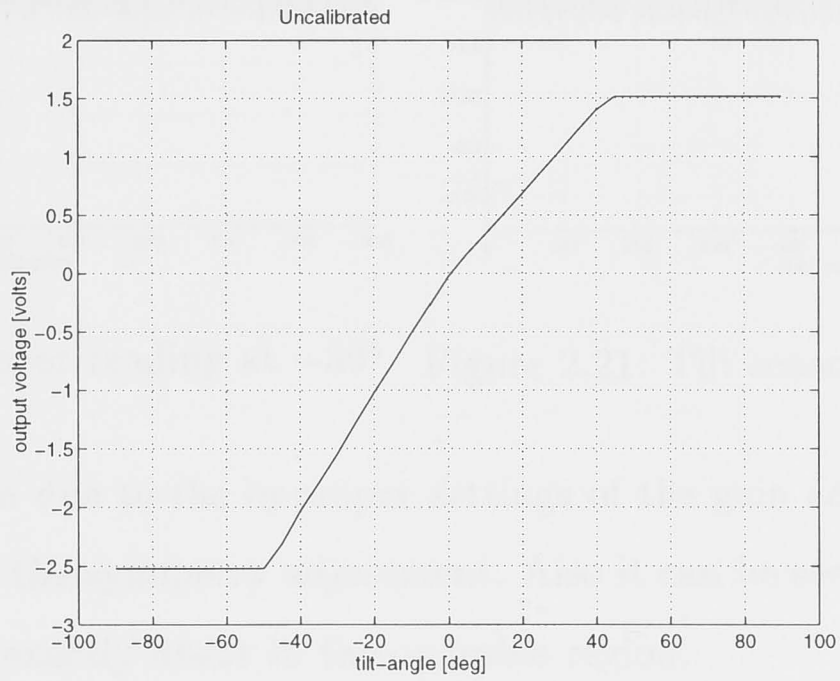


Figure 2.17: Static calibration curve of the tilt-sensor

The *static calibration curve* is shown in Figure 2.17. The measurement data used for plotting this curve is shown for some angles in Figures 2.18 to 2.21.

From Figure 2.17, it can be seen that the operable range of the tilt-sensor is $[-50°$ $+45°]$. But this is less than the range the specification sheets advise (see section 2.1.3).

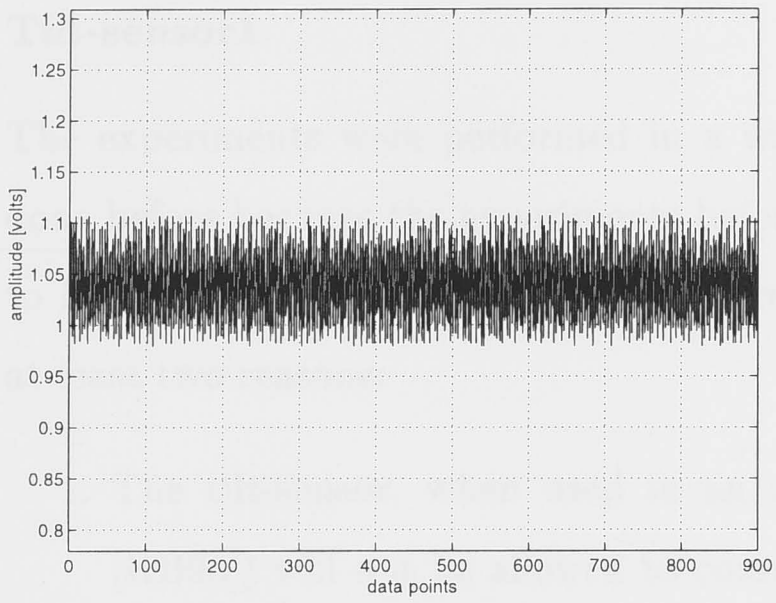


Figure 2.18: Tilt-sensor reading at 30°

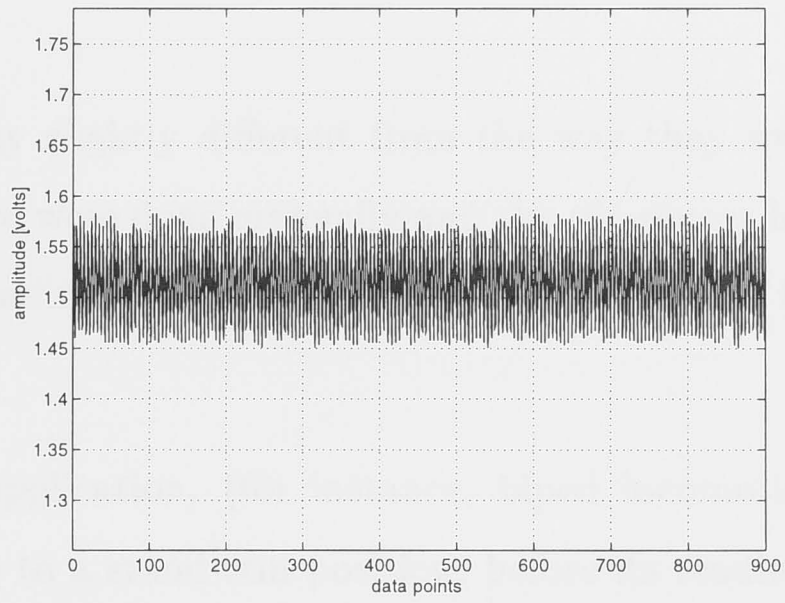


Figure 2.19: Tilt-sensor reading at 60°

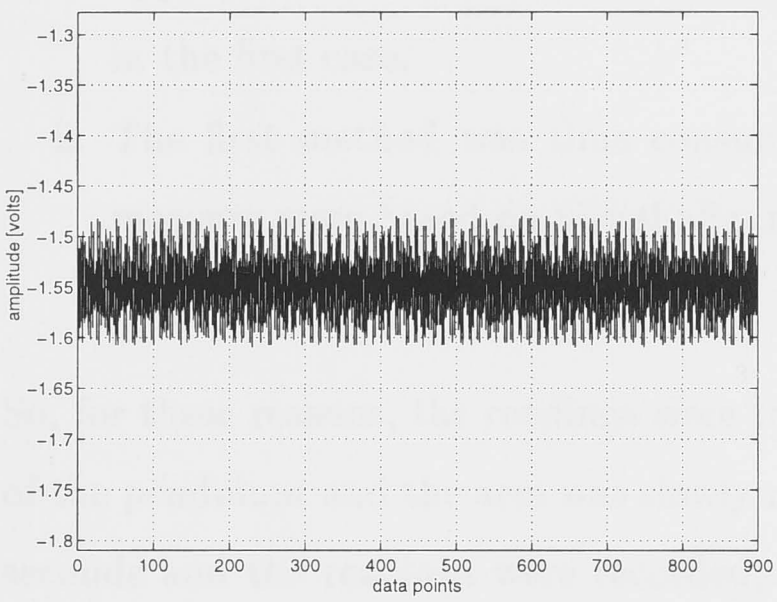


Figure 2.20: Tilt-sensor reading at −30°

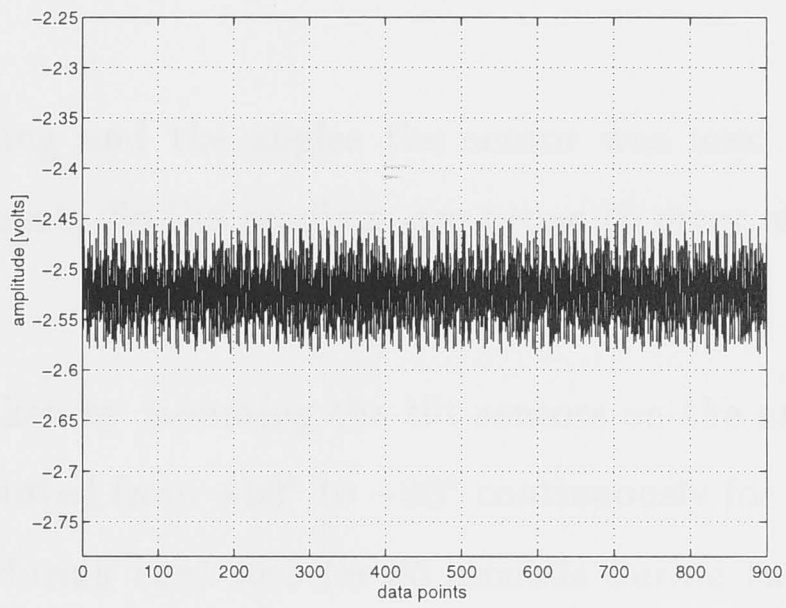


Figure 2.21: Tilt-sensor reading at −60°

This could have been due to the improper settings of the gain adjustment, the sensor offset correction and the symmetry adjustment. Also it can be seen that the tilt-sensor characteristic is not exactly linear in the operable region.

So the experiments were re-done with new settings for the gain adjustment, sensor offset correction and the symmetry adjustment and the results are plotted. The experiments now were performed with two different tilt-sensors (but of the same model). The experiments were done twice for each of the tilt-sensors (the corresponding results are plotted as run1 and run2). First the results of tilt-sensor1 are discussed and then that of tilt-sensor2.

Tilt-sensor1

The experiments were performed in a way slightly different from the way they were done before because the experiments before were done manually and the tilt-sensor has to be stopped at each angle before the readings were taken. This is not desirable for at least two reasons:

i. The tilt-sensor, when used in an application, (for instance, biped locomotion [KB93]) will not be allowed to come to a stand-still position, before its readings are utilised. The sensor is supposed to deliver the readings as and when the application-system moves. This is not the case when the experiments were done in the first case.

ii. The first method was time consuming and the angles the sensor was used to measure were based on visual-alignment. So the readings are prone to error and thus not desirable.

So, for these reasons, the readings were taken by mounting the tilt-sensors on the arm of the pendulum and the arm was slowly moved from $+90°$ to $-90°$ continuously for 30 seconds and the readings were recorded, during run1 and for 60 seconds during run2 (The same time-limits were followed for tilt-sensor2 to preserve uniformity). It was found that the operable range increases. This is because of the new settings as described earlier. Also it can be seen that in the operable range, the tilt-sensor is almost linear and this shows that linear characteristics can be obtained by proper settings of gain, symmetry and offset adjustments.

Run1 The experiments were done for an angle range of $[-90.575°\ \ 94.75°]$. The static calibration curve is shown in Figure 2.22.

Run2 The experiments were done for an angle range of $[-90.500°\ 93.375°]$. The static calibration curve is shown in Figure 2.23. The arm of the pendulum was allowed to go back to the initial (starting) position before the 60 seconds time limit. Accordingly, it can be seen that the tilt-sensor measurements trace the motion of the pendulum-arm.

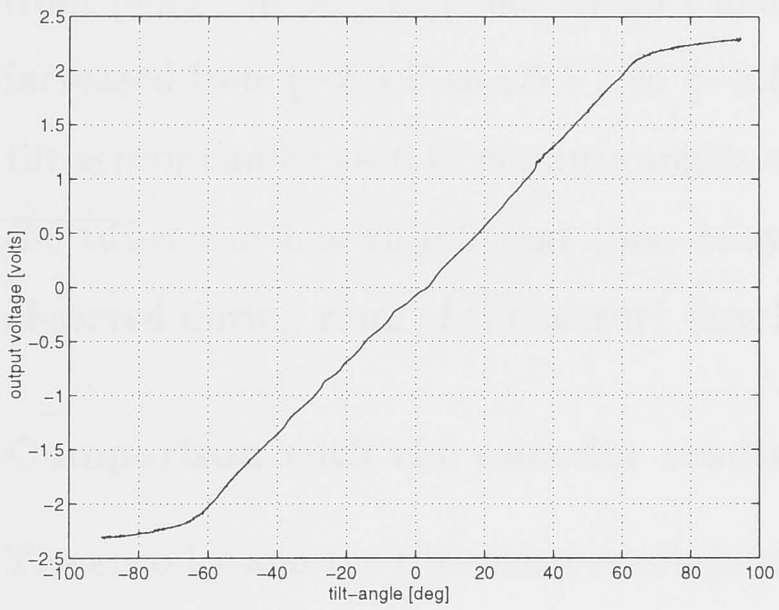


Figure 2.22: Tilt-sensor1 run1

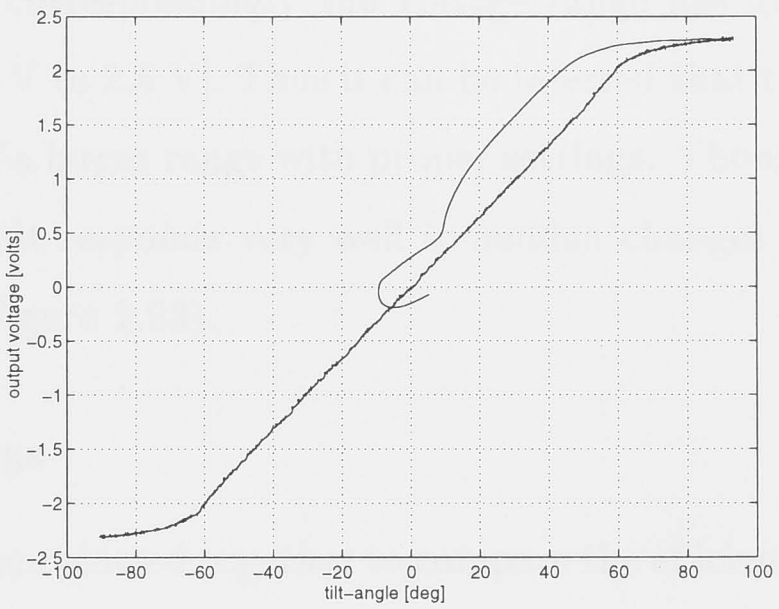


Figure 2.23: Tilt-sensor1 run2

Tilt-sensor2

The run1 and run2 were done for 30 and 60 seconds as in the tilt-sensor1 case.

Run1 The experiments were done for an angle range of [−93.675° 89.55°]. The static calibration curve is shown in Figure 2.24.

Run2 The experiments were done for an angle range of [−93.825 84.70]. The static calibration curve is shown in Figure 2.25.

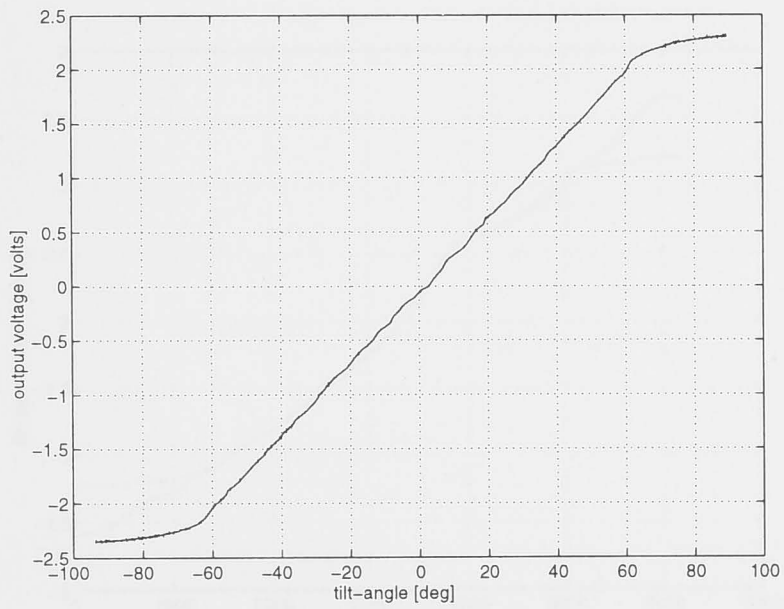


Figure 2.24: Tilt-sensor2 run1

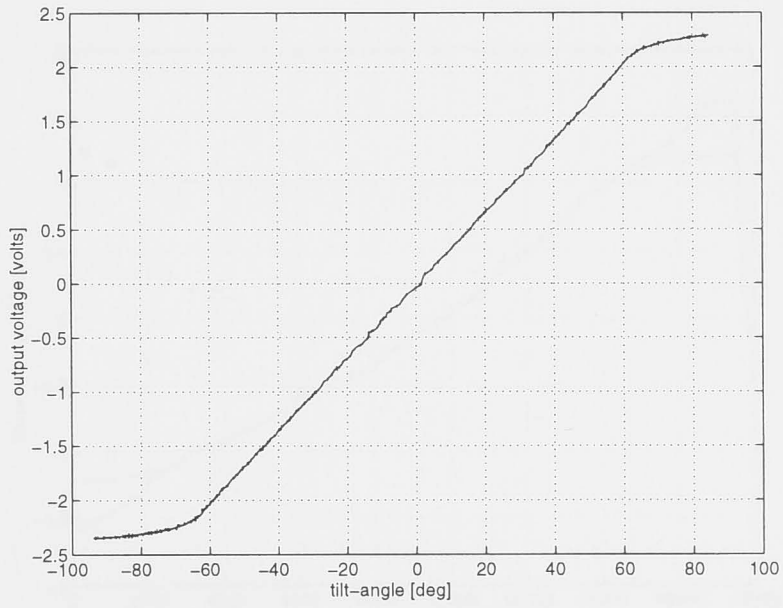


Figure 2.25: Tilt-sensor2 run2

From Figures 2.22 to 2.25, it can be noted that the range of operation has increased

from $[-50^\circ \ +45^\circ]$ to $[-60^\circ \ +60^\circ]$ and correspondingly the voltage range has also increased from [−2.5 V to 1.5 V] to [−2.5 V to 2.5 V]. Thus it can be inferred that the tilt-sensor can be used to measure angles of a larger range with proper settings. Though the tilt-sensor has an inherent time delay, it responds very well to sudden changes as observed during run2 of tilt-sensor1 (see Figure 2.23).

Comparison with the encoder readings

The encoder and the tilt-sensor readings were plotted together to compare the efficiency of operation of the tilt-sensor. The comparative plots for tilt-sensor1 (run1 and run2) are shown in Figures 2.26 and 2.27 and for tilt-sensor2 (run1 and run2) in Figures 2.28 and 2.29. It can be seen that there is a delay in the operation of the tilt-sensor as compared to the encoder operation. This delay is determined by the constants α_{ts} and β_{ts}.

The tilt-sensor has an inherent delay in its operation as compared to the encoder operation when used to measure the position of the arm of the pendulum. This is clearly a disadvantage. But in applications like bipedal walking [KB93], it is not feasible to use the encoder because of the size of the encoder that will make the design messier. So the use of the tilt-sensor becomes inevitable in such applications.

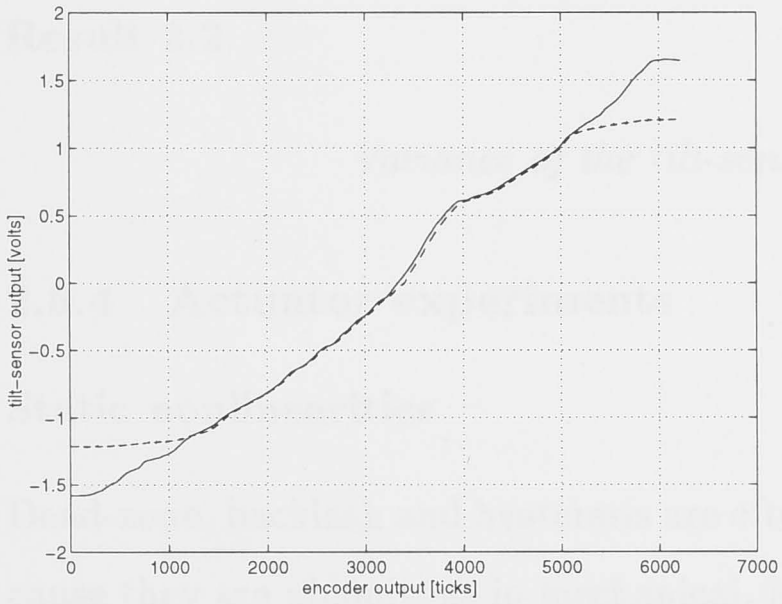


Figure 2.26: Tilt-sensor1 comparative-run1

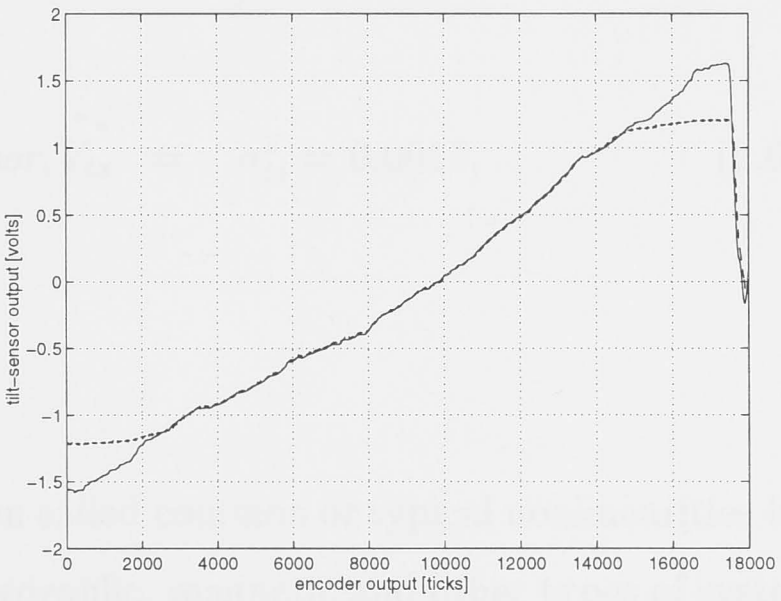


Figure 2.27: Tilt-sensor1 comparative-run2

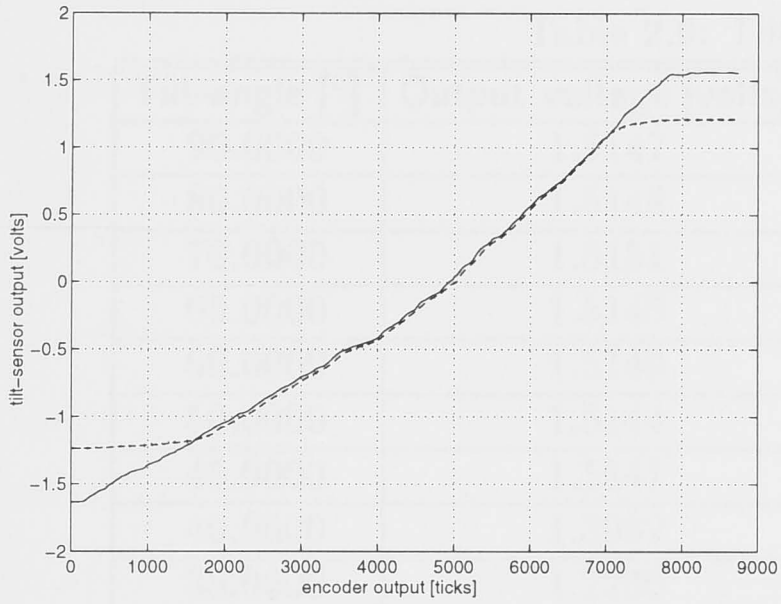


Figure 2.28: Tilt-sensor2 comparative-run1

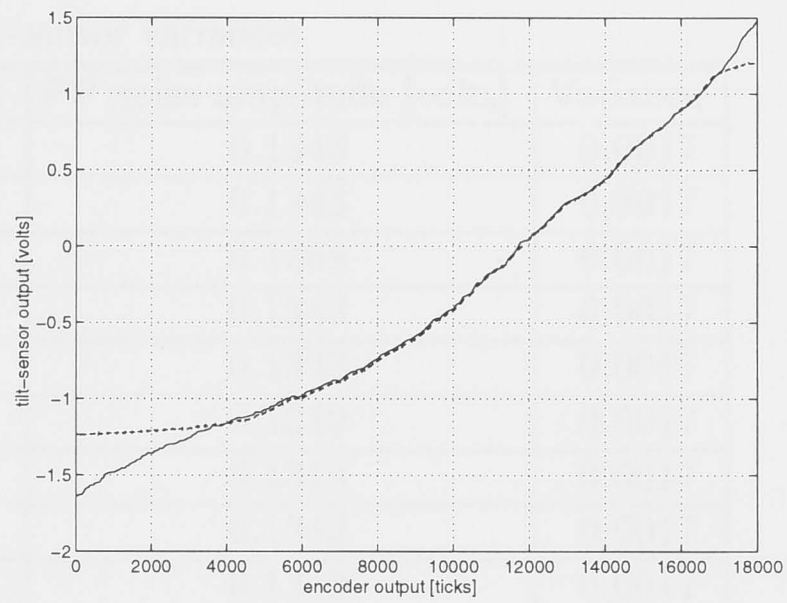


Figure 2.29: Tilt-sensor2 comparative-run2

Calculation of the variance of the noise

For the angles shown between $-90°$ and $+90°$, 900 data points were collected for each angle at a frequency of 300 Hz. Then the mean reading of each value was tabulated as the output voltage [volts], the maximum minus minimum value of the data was tabulated as the peak-peak noise amplitude [volts] and the variance at each angle was calculated using the MATLAB command *cov*( ) and tabulated against the corresponding angle. The variance of the noise associated with the tilt-sensor is obtained by averaging the values of the variance column in Table 2.9 and is found to be

Result 2.2

$$\textit{Variance of the tilt-sensor}, V_{ts} \quad = \quad \sigma_{ts}^2 = 0.0017. \tag{2.68}$$

2.5.4 Actuator experiments

Static nonlinearities

Dead-zone, backlash and hysteresis are often called common or typical nonlinearities because they are ubiquitous in mechanical, hydraulic, magnetic and other types of system components. In most cases, they are treated as imperfections of component characteristics. Dead-zone is a static memoryless nonlinearity which describes the component's

Table 2.9: Tilt-sensor variances

Tilt-angle [°]	Output voltage [volts]	PP noise amplitude [volts]	Variance
90.0000	1.5147	0.1343	0.0017
80.0000	1.5148	0.1343	0.0017
70.0000	1.5151	0.1368	0.0017
65.0000	1.5145	0.1343	0.0017
60.0000	1.5149	0.1343	0.0017
50.0000	1.5144	0.1319	0.0017
45.0000	1.5147	0.1319	0.0017
40.0000	1.3982	0.1343	0.0017
35.0000	1.2230	0.1319	0.0017
30.0000	1.0408	0.1295	0.0017
25.0000	0.8647	0.1295	0.0017
20.0000	0.6847	0.1319	0.0017
15.0000	0.5086	0.1290	0.0017
10.0000	0.3356	0.1270	0.0016
5.0000	0.1652	0.1295	0.0017
0.0000	-0.0276	0.1295	0.0018
-5.0000	-0.2751	0.1295	0.0017
-10.0000	-0.5176	0.1295	0.0017
-15.0000	-0.7767	0.1295	0.0017
-20.0000	-1.0222	0.1319	0.0017
-25.0000	-1.2800	0.1295	0.0017
-30.0000	-1.5477	0.1319	0.0017
-35.0000	-1.7943	0.1319	0.0017
-40.0000	-2.0303	0.1343	0.0017
-45.0000	-2.3106	0.1343	0.0017
-50.0000	-2.5215	0.1343	0.0017
-55.0000	-2.5212	0.1343	0.0017
-60.0000	-2.5217	0.1343	0.0017
-70.0000	-2.5214	0.1343	0.0017
-80.0000	-2.5213	0.1343	0.0017
-90.0000	-2.5217	0.1343	0.0017

insensitivity to small signals. In addition to this type of insensitivity, backlash and hysteresis also include delays and are, in fact, dynamic. There are applications in which nonlinear characteristics are intentionally introduced, such as in heating-cooling systems, where a dead-zone is needed to prevent simultaneous heating and cooling. In some hydraulic valves an intentional dead-zone prevents flow of fluid when the valve is

inactive [TK96]. Whether intentional or not, dead-zone, backlash and hysteresis usually have undesirable effects on feedback loop dynamics and control system performance.

Performance deterioration

It is easy to see that the presence of a dead-zone, backlash or hysteresis adversely affects the static accuracy of feedback control systems. With a linear controller, the static accuracy is limited by the width of the dead-zone, backlash or hysteresis. Attempts to improve it by increasing the gain of the feedback loop lead to sustained oscillations which may cause rapid wear of the gear trains and other components. Backlash and hysteresis are also harmful for the dynamic performance because of their inherent phase lag.

Dead zone

Dead-zone is a static input-output relationship which for a range of input values gives no output. Once the output appears, the slope between the input and the output is a constant. The simple dead-zone model appears in numerous studies of wide variety of phenomena, not limited to man-made systems. A dead-zone is often caused by friction. In such applications, the simple dead-zone model serves as an aggregate static approximation of more complex microscopic dynamic phenomena.

Thus for a corresponding change in the input voltage to the DC motor, if there were a dead-zone present, then the motor-shaft will not turn immediately or in other words, there will be a delay in the response of the motor to a subsequent change in the input starting voltage. The DC motor considered for analysis is tested for the presence of a dead-zone. Before proceeding further, a relationship between the motor voltage and the motor torque is developed as follows:

The voltage equation of a DC motor is

$$V_s = E_b + i_a R_a \tag{2.69}$$

where V_s is the supply voltage $[volts]$, R_a is the armature resistance $[Ohms]$ and E_b is the back-emf $[volts]$ and i_a is the armature current $[Amperes]$.

The back-emf can be written as

$$E_b = k_b \omega_m \tag{2.70}$$

where k_b is the back-emf constant $[\frac{V sec}{rad}]$ and ω_m is the rotor speed $[\frac{rad}{sec}]$.

Note that the low inductance in the motor can be neglected. In SI units the torque constant, k_t is equal to the back-emf constant, k_b and after some algebraic manipulations, an equation for the torque developed in terms of the rotor speed is obtained as

$$\tau_d = c_1[V_s - k\omega_m] \tag{2.71}$$

where $k_t \triangleq k_b \triangleq k$ and $c_1 \triangleq \frac{k}{R_a}$.

Experiments

Experiments were done with the existing pendulum setup detailed earlier. The control input signal and the rotor speed were measured. In the case of the rotor-speed measurement, the reading of the encoder mounted on the disc was used to obtain the disc position and thus the disc velocity was calculated which is directly proportional to the rotor-speed (since the shaft of the motor is coupled to the disc).

Thus with the rotor-speed and the input voltage, the motor torque can be calculated using (2.71). The corresponding motor characteristics are shown in Figures 2.30 and 2.31. It can be seen that the characteristics are linear.

In the second part of the experiment, the control signal voltage to the motor was applied after a delay and the motor started immediately. This proves that the motor can be started at any time. If there were a dead band, the instant starting of the motor would

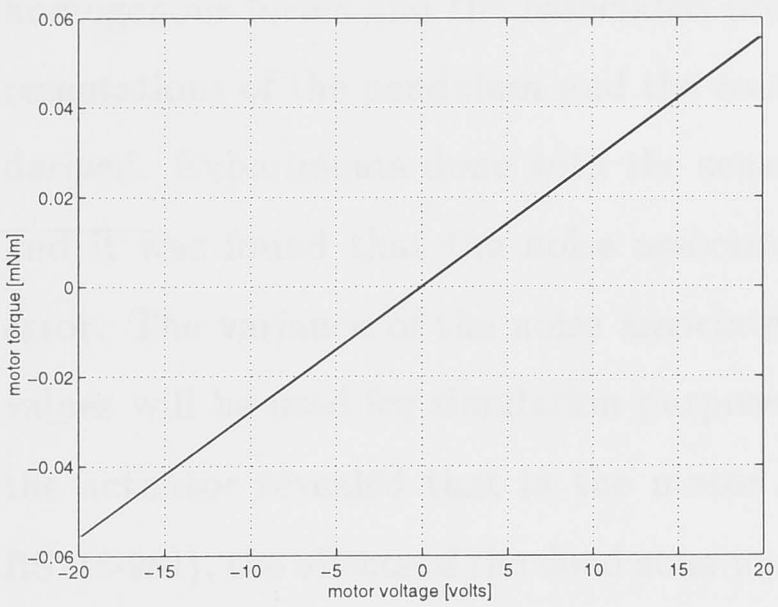


Figure 2.30: DC motor characteristics1

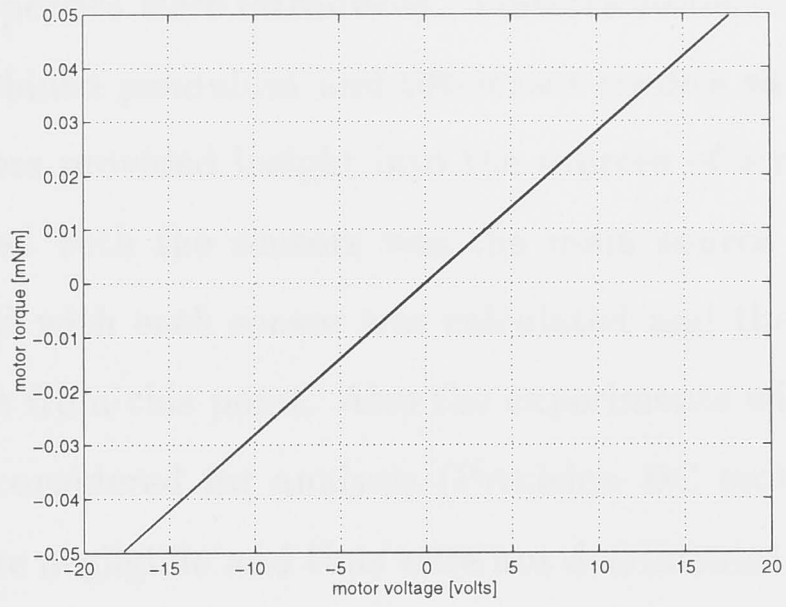


Figure 2.31: DC motor characteristics2

not have been possible. This proves that this motor (Precision DC motor RS718-981) has a very negligible amount of dead band.

Effect of noise

Noise comes into effect during commutation. But there is very little electrical noise present during commutation and thus the effects of noise are not pronounced much in the performance of the motor. So the noise need not be considered as a degrading factor in the performance of the Precision DC motor RS718-981.

2.6 Summary

In this chapter, the issues concerned with the modelling of the pendulum along with the possible combinations of the sensors were presented. The sensors that were considered are the encoder (HEDS series), the rate-gyro (muRata ENV-05S) and the tilt-sensor (Spectron L211U). Five combinations of the pendulum model and the above sensors were formulated and it was shown how these combinations can be employed for measuring the state or states of the pendulum model. A parameter estimation algorithm was developed for determining the parameters of the pendulum and the tilt-sensor models which was based on the assumption of small sampling intervals. The ZOH equivalence of polynomic nonlinear systems was discussed next where the concept of vector

homogenous forms and the associated properties were introduced. Volterra series representations of the pendulum and the combined pendulum and tilt-sensor models were derived. Experiments done with the sensors provided insight into the sources of error and it was found that the noise associated with the sensors was the main source of error. The variance of the noise associated with each sensor was calculated and these values will be used for simulation purposes from this point. Also the experiments with the actuator revealed that in the motor considered for analysis (Precision DC motor RS718-981), the effects of the dead zone were negligible and thus were not detrimental to its operation. The next chapter presents the estimation of the states of the pendulum.

Chapter 3

State Estimation

In this chapter, the state estimation of the pendulum model is discussed. Four Kalman filters: LKF, EKF1, EKF2 and EKF3 are employed for this purpose. LKF is a linear discrete time Kalman filter which is based on the linearised model of the pendulum. EKF1, EKF2 and EKF3 are discrete time extended Kalman filters. EKF1 is based on a continuous time nonlinear model and a discrete time measurement model where the nonlinear term is a sinusoid associated with the pendulum model. EKF2 is based on a continuous time nonlinear model and a discrete time measurement model where the nonlinear term $sin\ \theta$ is approximated by the first two terms of the power series expansion of $sin\ \theta$, i.e., $\theta - (\theta^3/6)$ and EKF3 is an approximated version of EKF2. The Volterra series representation of the pendulum model was derived in Chapter 2. It will be shown that this representation of the pendulum model falls into a special category called the *extended generalised Wiener model* structure. The structure of the extended generalised Wiener model is such that it has a linear subsystem followed by a polynomial type nonlinearity which is followed by another linear subsystem. It is demonstrated that the EKF2 and EKF3 approaches serve as powerful tools for estimating the states of these systems based on the input-output relations developed at discrete instants of time. The results are then compared with the standard extended Kalman filtering approach (i.e., EKF1) and the EKF2 and EKF3 approaches are shown to be more numerically efficient

than the standard approach for the pendulum model. All the possible combinations of the sensor-measurements are used for the estimation of the states of the pendulum and the corresponding state estimation error variances are provided along with the simulation results.

3.1 Kalman Filtering

State estimation can be broadly classified into three major types namely *filtering*, *smoothing* and *prediction* [Gel74]. When the time at which an estimate is desired coincides with the time of the last measurement, the problem is referred to as *filtering*; when the time of interest is before the last measurement, the problem is termed *smoothing*; and when the time of interest occurs after the last measurement, the problem is called *prediction*.

Let $\mathbf{x}(t) \in \mathcal{R}^n$ be a vector function of time representing the state of a process or plant. Then a function $\mathbf{z}(\mathbf{x}, t)$ of both the state $\mathbf{x}$ and time t is called an *observation* (or *measurement*) process. Suppose the observation $\mathbf{z}(\mathbf{x}, t)$ is recorded for all t in the interval $[0, t]$, then the filtering problem is to estimate $\mathbf{x}(t)$ using all the observations in the interval up to time t. Denote the estimate by $\hat{\mathbf{x}}(t)$.

If $\hat{\mathbf{x}}(t)$ is selected so that the probability $\mathbf{x}(t) = \hat{\mathbf{x}}(t)$ is maximised, the estimate $\hat{\mathbf{x}}(t)$ is called the *maximum likelihood estimate* of $\mathbf{x}(t)$. If $\hat{\mathbf{x}}(t)$ is selected so that the covariance of the estimate $\xi\{[\mathbf{x}(t) - \hat{\mathbf{x}}(t)][\mathbf{x}(t) - \hat{\mathbf{x}}(t)]^T\}$ is minimised (where ξ denotes the mathematical expectation operator), the estimate is called the *minimum variance estimate*. For a linear plant with additive Gaussian white noise, the state of the plant $\mathbf{x}(t)$ is a random process with a Gaussian distribution and any estimate of $\mathbf{x}(t)$ based on a linear function of $\mathbf{x}(t)$ with additive Gaussian white noise will also be a Gaussian random variable. In this case, the maximum likelihood estimate and the minimum variance estimate are equal [AM79, Pel70]. A minimum variance (*unbiased*) estimate has the property that its error variance is less than or equal to that of any other unbiased

estimate. A *consistent* estimate is one which converges to the true value of $\mathbf{x}(t)$, as the number of observations increases.

According to [Gel74], an optimal estimator is defined as follows:

> *An optimal estimator is a computational algorithm that processes measurements to deduce a minimum error estimate of the state of a system by utilising knowledge of system and measurement dynamics, assumed statistics of system noises and measurement errors and initial condition information.*

The Kalman filter is such an optimal estimator that can be characterised as an algorithm for computing the conditional mean and covariance of the probability distribution of the state of a linear dynamic finite-dimensional stochastic system with uncorrelated Gaussian process and measurement noise. The conditional mean is the unique unbiased estimate, and is propagated in feedback form by solving a system of linear equations. The conditional covariance is propagated by solving a nonlinear equation.

3.1.1 Linear Kalman filtering

Different types of Kalman filters are possible depending upon on the *type* of plant and measurement models. Before proceeding further, for completeness, the notations and the associated meaning of the Kalman filters that will be used hereafter in this thesis are summarised in Table 3.1. In particular, four types of Kalman filtering are considered for the state estimation of the pendulum model. When the factor $sin\ \theta$ in the pendulum model equation is approximated by θ, the model is linear and the linear Kalman filter, called LKF, is used for the state estimation of the pendulum. LKF uses a continuous time plant model and discrete measurements which are obtained at each sampling instant. In EKF1, the factor $sin\ \theta$ is not approximated and it is based on a continuous time plant model and a discrete time measurement model. EKF2 is based on a continuous time plant model and a discrete time measurement model where the factor $sin\ \theta \approx \theta - (\theta^3/6)$ and EKF3 is based on an approximation of EKF2.

Table 3.1: Kalman filter notations

Abbreviation	Filter name	Type	Plant model	Meas. model
LKF	Linear Kalman Filter	Linear	Continuous	Discrete
EKF1	Extended Kalman Filter 1	Nonlinear	Continuous	Discrete
EKF2	Extended Kalman Filter 2	Nonlinear	Continuous	Discrete
EKF3	Extended Kalman Filter 3	Nonlinear	Continuous	Discrete

Continuous-time Kalman filtering

The continuous-time Kalman filter is summarised in Table 3.2 [AM91]. The notation $\mathbf{x} \sim N(\mathbf{m}, \mathbf{P})$ indicates that $\mathbf{x}$ is a Gaussian (normal) random vector with mean $\mathbf{m}$ and covariance $\mathbf{P}$. For the model, $\mathbf{x} \in \mathcal{R}^n$, $u \in \mathcal{R}^m$ and $\mathbf{z} \in \mathcal{R}^p$ denote respectively the state, input and observation. The system matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ are $n \times n$, $n \times m$ and $p \times n$ respectively. The processes $\omega(\cdot)$ and $v(\cdot)$ are assumed to define zero mean white noise processes such that

$$\xi \left\{ \begin{pmatrix} \omega(t) \\ v(t) \end{pmatrix} \begin{pmatrix} \omega^T(s) & v^T(s) \end{pmatrix} \right\} = \begin{pmatrix} \mathbf{Q} & \mathbf{S} \\ \mathbf{S}^T & \mathbf{R} \end{pmatrix} \delta(t-s) \tag{3.1}$$

where $\delta(\cdot)$ is the Dirac delta function, ξ indicates the mathematical expectation operator and $\mathbf{R} = \mathbf{R}^T > 0$ and $\mathbf{Q} = \mathbf{Q}^T \geq 0$. The initial state $\mathbf{x}(t_0)$ will be assumed to be a Gaussian random variable, of mean $\hat{\mathbf{x}}_\mathbf{0}$ and covariance $\mathbf{P_0}$. The filtering problem is to use the measurements to estimate $\mathbf{x}(t)$, denoted by $\hat{\mathbf{x}}(t)$, so as to minimise $\xi\{||\mathbf{x}(t) - \hat{\mathbf{x}}(t)||\}$. The matrix $\mathbf{P}(t) \stackrel{\triangle}{=} \xi\{[\mathbf{x}(t) - \hat{\mathbf{x}}(t)][\mathbf{x}(t) - \hat{\mathbf{x}}(t)]^T\}$ is given by the solution of the Riccati equation given in Table 3.2.

LKF approach

In this approach, the system model is *continuous* and the measurement model is *discrete*. The measurements are made once in every sampling interval, T_s. Suppose that a measurement has been made at the discrete-time instant kT_s, then the information that it provides is applied in updating the estimate of the state $\mathbf{x}$ of a stochastic system at the discrete-time instant kT_s. The measurements are linearly related to the state.

Table 3.2: Summary of the continuous-continuous Kalman filter

System model Measurement model	$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t) + \omega(t);\ \omega(t) \sim N(\mathbf{0}, \mathbf{Q})$ $\mathbf{z}(t) = \mathbf{C}\mathbf{x}(t) + v(t);\ v(t) \sim N(\mathbf{0}, \mathbf{R})$
State estimate	$\dot{\hat{\mathbf{x}}}(t) = \mathbf{A}\hat{\mathbf{x}}(t) + \mathbf{B}u(t) + \mathbf{K}(t)\left[\mathbf{z}(t) - \mathbf{C}\hat{\mathbf{x}}(t)\right]$ $\hat{\mathbf{x}}(0) = \hat{\mathbf{x}}_0$
Error covariance propagation	$\dot{\mathbf{P}}(t) = \mathbf{A}\mathbf{P}(t) + \mathbf{P}(t)\mathbf{A}^T + \mathbf{Q}$ $-\mathbf{K}(t)\mathbf{R}(t)\mathbf{K}^T(t);\ \mathbf{P}(0) = \mathbf{P}_0$
Kalman gain matrix	$\mathbf{K}(t) = \mathbf{P}(t)\mathbf{C}^T\mathbf{R}$

This type of filtering is employed for the state estimation of the pendulum based on the linearised model presented in section 2.2 of chapter 2 and will be called the *Linear Kalman Filtering* (LKF) approach. The corresponding discrete-time system matrices are obtained by the discretisation procedure outlined in section 2.3.1. The LKF is summarised in Table 3.3 where

$\mathbf{I}$ denotes the identity matrix of appropriate order,

$\mathbf{x}_k$ is the state at the discrete time instant kT_s,

$\mathbf{z}_k$ is the measurement made at the discrete time instant kT_s,

v_k is the sampled measurement noise at the discrete time instant kT_s,

$\hat{\mathbf{x}}_k(-)$ is the *a priori* estimate at the discrete time instant kT_s
(i.e., the estimate before the measurement $\mathbf{z}_k$),

$\hat{\mathbf{x}}_k(+)$ is the *a posteriori* estimate at the discrete time instant kT_s using measurements $\mathbf{z}_k$
(i.e., the estimate after the measurement $\mathbf{z}_k$),

$\mathbf{P}_k(-)$ is the *a priori* covariance at the discrete time instant kT_s
(i.e., the error covariance matrix before the measurement $\mathbf{z}_k$),

$\mathbf{P}_k(+)$ is the *a posteriori* covariance at discrete time instant kT_s using measurements $\mathbf{z}_k$
(i.e., the error covariance matrix after the measurement $\mathbf{z}_k$) and

$\mathbf{K}_k$ is the Kalman gain matrix at the discrete time instant kT_s,

All the above indicated notations are common to all the Kalman filters that will be mentioned from this point.

Table 3.3: Summary of the continuous-discrete Kalman filter

System model	$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t) + \omega(t);\ \omega(t) \sim N(\mathbf{0}, \mathbf{Q})$
Measurement model	$\alpha_j \dot{v}_j(t) = -v_j(t) + \eta_j(t);\ j = 1, 2.$ $\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + v_k;\ k = 1, 2, \cdots$ $\eta \sim N(\mathbf{0}, 1);\ v_k = v(kT_s);$
State estimate propagation Error covariance propagation	$\dot{\hat{\mathbf{x}}}(t) = \mathbf{A}\hat{\mathbf{x}}(t) + \mathbf{B}u(t);\ kT_s \leq t < (k+1)T_s$ $\dot{\mathbf{P}}(t) = \mathbf{A}\mathbf{P}(t) + \mathbf{P}(t)\mathbf{A}^T + \mathbf{Q}$
Kalman gain matrix Error covariance update State estimate update	$\mathbf{K}_k = \mathbf{P}_k(-)\,\mathbf{H}^T\left[\mathbf{H}\mathbf{P}_k(-)\,\mathbf{H}^T + \mathbf{R}_k\right]^{-1}$ $\mathbf{P}_k(+) = [\mathbf{I} - \mathbf{K}_k\mathbf{H}]\,\mathbf{P}_k(-)$ $\hat{\mathbf{x}}_k(+) = \hat{\mathbf{x}}_k(-) + \mathbf{K}_k[\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_k(-)]$
Initial conditions Other assumptions	$\mathbf{x_0} \sim N(\hat{\mathbf{x}}_0, \mathbf{P}_0)$ $\xi\left\{\omega(t)v_k^T\right\} = 0\ \ \forall\ k \text{ and } t$

3.1.2 Measurement noise model

Many signals occur in physical processes that cannot be predicted in advance. Informally, such signals can be modelled as noise. They may occur either as exogenous inputs to the plant or as extraneous signals in the outputs of sensors. The former case can be modelled as the process noise and the latter as the measurement noise. The noise associated with the sensors was discussed in chapter 2. In many situations, a noise source can be adequately represented as the response of a linear system to white noise, as illustrated in Figure 3.1.

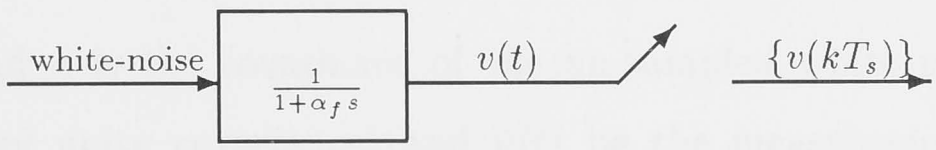


Figure 3.1: Sampled measurement noise representation

An anti-aliasing filter is used in the generation of the measurement noise as the sampled-noise is utilised once in every T_s seconds. The covariance of the noise will change

according to the sampling interval. The sampled white-noise is to be utilised in the Kalman filtering approaches and this in turn will affect the Kalman filter gain and thus the entire estimation process. Thus the relation between the covariance of the sampled noise and the sampling interval is to be determined. The frequency response of the anti-aliasing filter is shown in Figure 3.2.

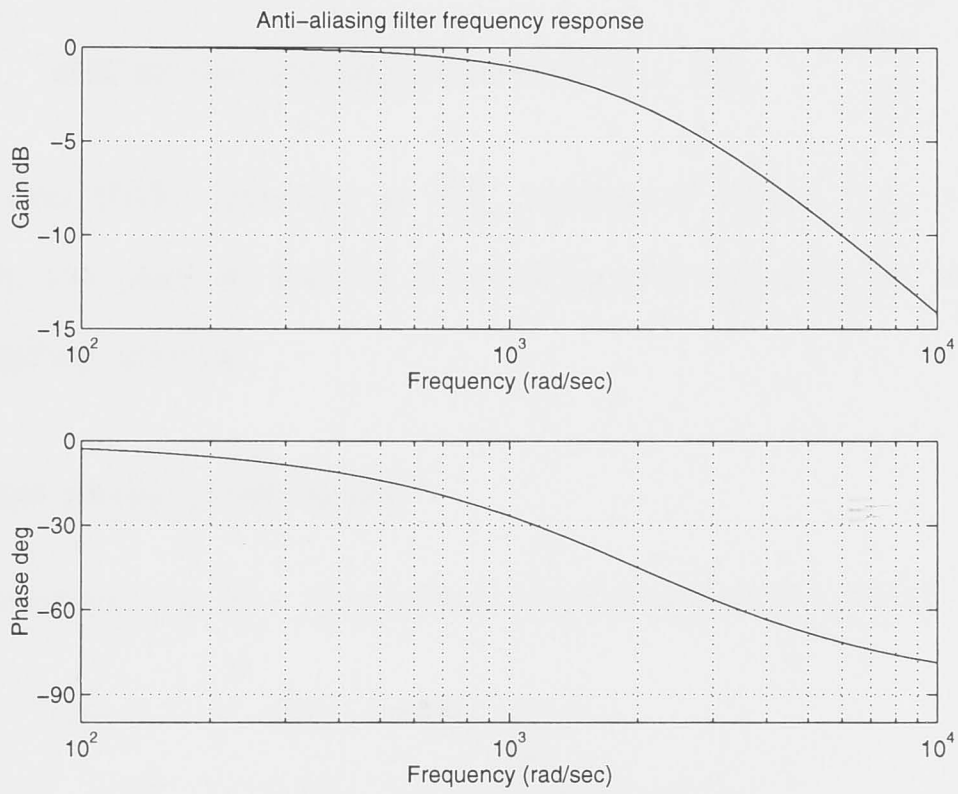


Figure 3.2: Anti-aliasing filter frequency response

Discrete measurement noise covariance

Let R_k be the covariance of the sampled measurement noise sequence $\{v(kT_s)\}$. Then [Wil91]

$$R_k = 0.5\lambda a^{-1}\left(e^{2aT_s} - 1\right) \text{ when } a < 0 \tag{3.2}$$

where $a = \frac{-1}{\alpha_f}$ and λ is the covariance of the un-sampled white noise. Let $\omega(t)$ be the white noise (of unity covariance) and $v(t)$ be the measurement noise (i.e., the measurement noise before sampling which is the output of the anti-aliasing filter). $v(t)$ after sampling results in the sampled sequence $\{v(kT_s)\}$ as shown in Figure 3.1. For small sampling intervals (i.e., T_s small), $e^{2aT_s} \approx 1 + 2aT_s$ and using this approximation in (3.2), the desired relation between the covariance of the sampled measurement noise

and the sampling interval, T_s, is found to be,

$$R_k = \frac{T_s}{\alpha_f^2} \tag{3.3}$$

Then the covariance R_k of $\{v(kT_s)\}$ when

$$T_s = \Delta T_s,\ R_{k1} = \frac{\Delta T_s}{\alpha_f^2} \tag{3.4}$$

$$\text{and at the sampling intervals } T_s,\ R_{k2} = \frac{T_s}{\alpha_f^2} \tag{3.5}$$

Note that LKF and EKF1 operate at ΔT_s instants, where $\Delta T_s = \frac{T_s}{n}$ is the Euler-integration (to be discussed in section 3.2.1) step-size and n is the number of propagations in one sampling interval.

Discrete process noise covariance

The discrete representation of a continuous-time plant with $\{\mathbf{A}, \mathbf{B},\ \omega(t)\}$ is given by,

$$\mathbf{x}_{k+1} = \mathbf{F}\mathbf{x}_k + \mathbf{G}u_k + \omega_k \tag{3.6}$$

$$\mathbf{F} = e^{\mathbf{A}T_s};\ \mathbf{G} = \int_0^{T_s} e^{\mathbf{A}\tau}\ \mathbf{B}\ d\tau$$

$$\xi\left\{\omega_k \omega_m^T\right\} = \mathbf{Q}_k\ \delta(k-m);\ \mathbf{Q}_k = \int_0^{T_s} e^{\mathbf{A}\tau}\ \mathbf{Q}\ e^{\mathbf{A}^T\tau}\ d\tau \tag{3.7}$$

where $\mathbf{Q}_k$ is the discrete process noise covariance and $\mathbf{Q}$ is the continuous process noise covariance. Also $\mathbf{Q}_k$ in (3.7) satisfies the continuous-Lyapunov equation [Wil91]

$$\mathbf{A}\mathbf{Q}_k + \mathbf{Q}_k\mathbf{A}^T = -\mathbf{Q}_1 \tag{3.8}$$

$$\text{where } \mathbf{Q}_1 = -\int_0^{T_s} \frac{d}{d\tau}\left[e^{\mathbf{A}\tau}\mathbf{Q}e^{\mathbf{A}^T\tau}\right] d\tau$$

$$= \mathbf{Q} - e^{\mathbf{A}T_s}\mathbf{Q}e^{\mathbf{A}^T T_s}$$

The Lyapunov equation for $\mathbf{Q}_k$ has a unique solution *iff* $\lambda_i + \lambda_j \neq 0\ \ \forall$ eigenvalues λ_j of $\mathbf{A}$.

3.1.3 Application of LKF to the pendulum model

The LKF is employed for the state estimation of the linearised pendulum model. All the five sensor combinations (that were discussed in section 2.2 of chapter 2) are employed.

The output matrix $\mathbf{H}$ changes according to the type of sensor employed. For instance, when only the encoder is employed, $\mathbf{H} = (1 \;\; 0)$. In the case of the encoder and rate-gyro combination and the rate-gyro and tilt-sensor combination, two anti-aliasing filters are employed, one for each sensor whereas in the individual sensor cases, one anti-aliasing filter is employed for the generation of the measurement noise. The matrices $\mathbf{A}$ and $\mathbf{B}$ for this approach are

$$\mathbf{A} = \begin{pmatrix} 0 & 1 \\ -\frac{K_0}{J} & -\frac{K_1}{J} \end{pmatrix}; \; \mathbf{B} = \begin{pmatrix} 0 & \frac{G_0}{J} \end{pmatrix}^T$$

The true state and its encoder measurement and the (transient) state estimation error plots (when the measurements are provided by the encoder) are shown in Figures 3.3 and 3.4. A sampling interval of 0.2 seconds was chosen for the simulation study. It can be seen from Figure 3.3, that θ (state1 of the pendulum) is approximately in the range of $\pm$ 60° and thus the nonlinearities are severe around these angles. This encoder measurement is the same for all the approaches considered. Also the noise sequences remain the same for all the simulations done and hence comparisons between different approaches are valid.

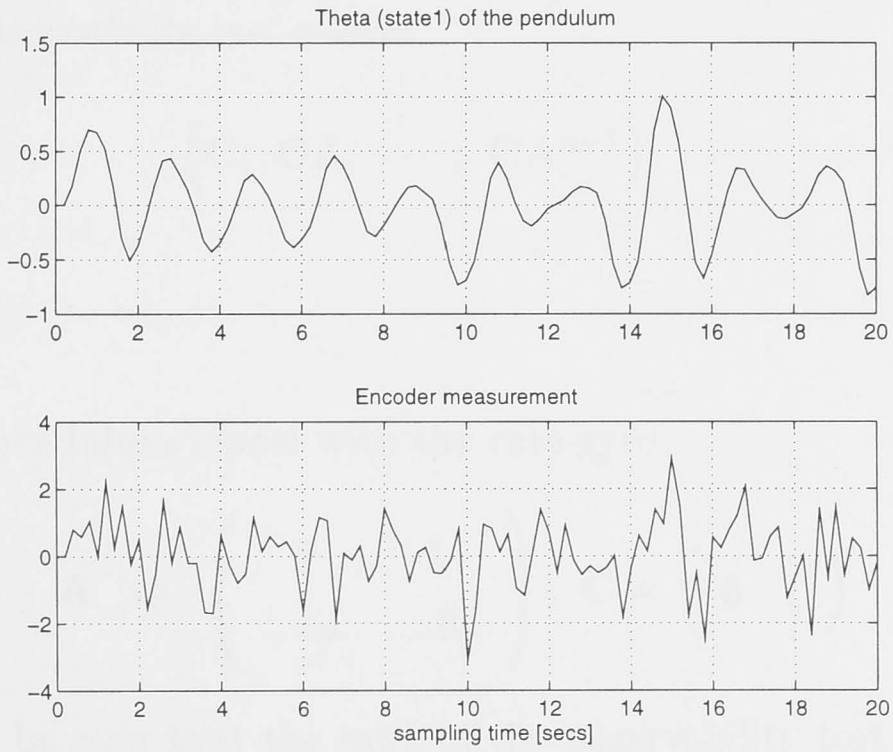


Figure 3.3: True θ of the pendulum and its encoder measurement

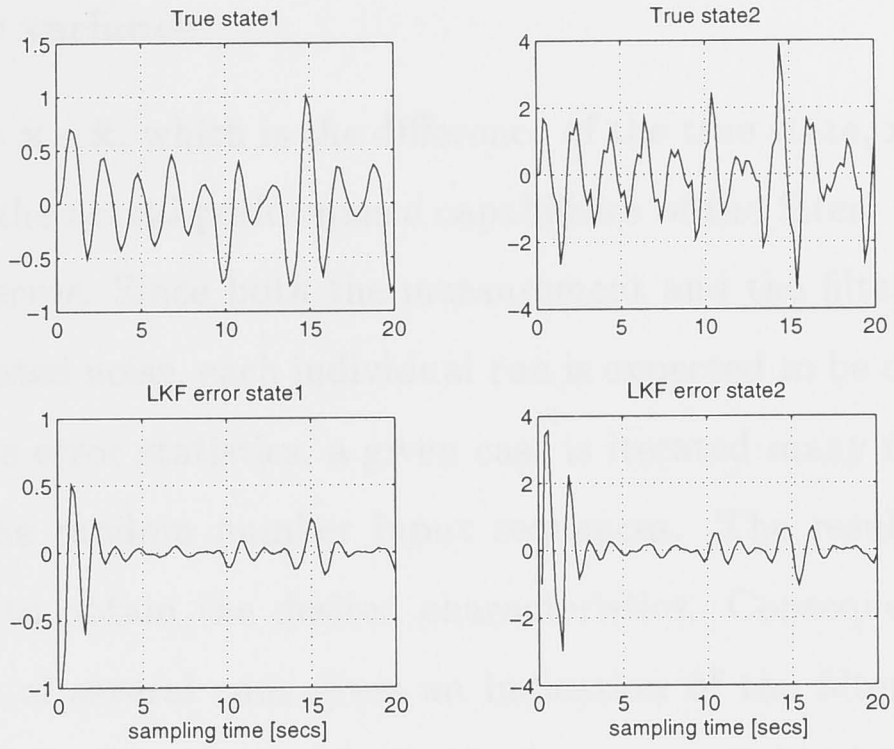


Figure 3.4: LKF state estimation errors with encoder measurements

Observability

Observability questions relate to the problem of determining the values of the state vector knowing only the output y over some interval of time [Bro70].

Lemma 3.1 *For a continuous-time system with system matrices* $\mathbf{A}$, $\mathbf{B}$ *and* $\mathbf{C}$, *for* $\mathbf{A}$ *and* $\mathbf{C}$ *constant and* $\mathbf{A}$ $n \times n$, *the* n *dimensional linear constant system is called observable if the observability test matrix*

$$\left(\mathbf{C};\ \mathbf{CA};\ \cdots\ ;\ \mathbf{CA}^{n-1}\right) \tag{3.9}$$

is of rank n.

For the linearised pendulum model with the rate-gyro,

$$\mathbf{A} = \begin{pmatrix} 0 & 1 \\ -\frac{K_0}{J} & -\frac{K_1}{J} \end{pmatrix};\ \mathbf{C} = \begin{pmatrix} 0 & 1 \end{pmatrix}$$

Using (3.9), it can be seen that the rank of the observability test matrix is of rank 2 and thus the system is observable. So from measuring the output $\dot{\theta}$, it is possible to estimate the value of θ of the pendulum.

Estimation error variance

The error state $\mathbf{e} = \mathbf{x} - \hat{\mathbf{x}}$, which is the difference of the true state, $\mathbf{x}$ and the estimated state, $\hat{\mathbf{x}}$, describes the actual performance capabilities of the filter. This error is known as the *estimation error*. Since both the measurement and the filter models are driven by randomly generated noise, each individual run is expected to be different. Therefore, in order to generate error statistics, a given case is iterated many times, the iterations differing only in the random number input sequences. The results of the iterations are then averaged to obtain the desired characteristics. Consequently, observing the ensemble statistics of several runs gives an indication of the filter's expected performance. Naturally, the more runs are made, the more reliable become the statistics. Each run produces a different sequence of random numbers to generate the samples of input white-noise processes. Sample error statistics are computed for each sampling time point using the equations

$$\mu_{est} = \frac{1}{L}\sum_{i=1}^{L}(\mathbf{x}_i - \hat{\mathbf{x}}_i) \tag{3.10}$$

$$\sigma^2_{est} = \frac{1}{L-1}\sum_{i=1}^{L}\left(\mathbf{x}_i^2 - \frac{L}{L-1}\,\mu^2_{est}\right) \tag{3.11}$$

where μ indicates the sample mean, σ^2 indicates the sample variance, the subscript *est* indicates estimation and L is the number of computer runs. The estimation error variances for all the sensor combinations and for each state (separated by commas from the next state) are shown in Table 3.4. Also note that, when the tilt-sensor is employed, the tilt-sensor dynamics introduce a third state.

Table 3.4: Estimation error variances in the LKF case

Sensor	Estimation error variance
Encoder	0.0226, 0.3867
Rate-gyro	0.0228, 0.3865
Encoder & rate-gyro	0.0224, 0.3863
Tilt-sensor	0.0230, 0.3868, 0.0003
Rate-gyro & tilt-sensor	0.0228, 0.3867, 0.0003

3.2 Extended Kalman Filtering

The optimal estimation problem for nonlinear systems is in general very complicated and is considerably more difficult when the system contains nonlinear elements because the probability density functions of signals and noise are altered as they are transmitted through these elements. Gaussian inputs cause non-Gaussian response and the criteria for best state estimates are not quite so obvious as in the linear case. The mean and standard deviation are incomplete descriptors of the probability density function and a state estimate based on the conditional mean may be different from one based on the mean or median. For a nonlinear plant and nonlinear observations, the probability density function of the estimate conditioned on all of the observations is *not* generally Gaussian. In this case, the maximum likelihood estimate will differ from the minimum variance estimate [Pel70]. Thus the procedure used to solve the nonlinear filtering problems will differ from its linear counterpart depending upon which type of estimate is desired.

Fortunately estimators for many nonlinear systems can be based on the Kalman-Bucy filters for linear systems [Ste86]. These *modified* linear-optimal estimators are useful when the stochastic effects are additive and small, either as a result of the original system's structure or of reasonable assumptions regarding magnitudes of these stochastic effects. Details of specific probability density functions may not be well portrayed but the overall performance in state estimation can be satisfactory. As the nonlinear system is complicated by the fact that that the probability density function is not necessarily Gaussian, the mean and covariance may not completely describe it. In the general case, a countably infinite number of parameters are necessary to completely describe the filtered parameters. Because of this, approximations involving a finite number of parameters are sought. In the case of nonlinear filtering theory, one approach is to approximate the expectation of a function by Taylor series truncated after the quadratic terms which is equivalent to assuming certain higher order moments to be negligible which has the advantage of relative simplicity.

Extended Kalman filtering

The (linear) Kalman filter is a recursive data-processing algorithm in which at the update time, it combines all available measurements, plus prior knowledge about the system and the measuring devices, to produce an estimate of the state $\mathbf{x}(t)$ in such a manner that the mean square is minimised statistically. During propagation, it advances the estimate in such a way as to again maintain optimality. Thus the Kalman filter performs the above tasks for linear systems and linear measurements in which the driving and measurement noises are assumed to be mutually uncorrelated, white, zero-mean and Gaussian. In nature, however, most physical problems or processes are nonlinear. Consequently, the nonlinear systems must be linearised (that is approximated) before the linear filter theory can be applied.

The idea of the extended Kalman filter, EKF, is to use the ideas of Kalman filtering for a nonlinear problem. The filter gain is computed by linearising the nonlinear model. The EKF, in contrast to the Kalman filter for linear systems, is *not* an optimal filter, since it is based on approximations. It retains the linear calculation of the covariance and gain matrices, and it updates the state estimate using a linear function of the filter residual; however it uses the original nonlinear equations for the state propagation (extrapolation in the discrete case) and the definition of the output vector. Thus the EKF assumes the validity of linearisation.

When applying the extended Kalman filter to real-time applications, the requirement to recalculate the Jacobian matrices continually can impose a severe computational burden. The time-varying Jacobian matrices cannot be precomputed because they are functions of the state estimate. Therefore, as soon as a new state estimate is made, a new and better reference-state trajectory is incorporated into the estimation process. In this manner, the validity of the assumption, that deviations from the reference trajectory are small enough to allow linearisation techniques to be employed with adequate results, is enhanced. Consequently the Kalman gain matrix also depends on the state

estimate and all the filter calculations must be done in real-time.

Improved estimates could be obtained from a second- or higher-degree filter that retains more terms in the Taylor series expansions and the accompanying filter derivations. Two variations of the conventional EKF used occasionally are the iterated EKF and the second-order filters. Both of these are higher-order filters. The approach in the former filter is to iterate within the linearisation step few times, thus improving the quality of the estimates; however, this results in an increase in the number of operations. The second-order filter, as the name implies, is obtained by including second-order terms in the expansion for the Jacobians.

EKF1 approach

If sampled-data measurements of a continuous process are available, a continuous-discrete extended Kalman filter is most appropriate for state estimation [Ste86]. This EKF will be hereafter referred to as the EKF1. The EKF1 is based on a continuous-time plant model and a discrete-time measurement model. The EKF1 is summarised in Table 3.5.

For the propagation of the state estimates and the error covariance, some form of propagation scheme is to be employed. A discussion of various schemes is presented next.

3.2.1 Numerical integration of nonlinear equations

Numerical approximation of a continuous-time equation leads directly to a discrete-time equation where the sequence of evaluation times is uniformly spaced. The general form of the dynamic equation of the model is represented by

$$\dot{\mathbf{x}}(t) \;=\; \mathbf{f}\left(\mathbf{x}(t), \mathbf{u}(t)\right) \tag{3.12}$$

Solving for $\mathbf{x}(t)$ in the finite interval $[t_i, t_f]$ requires the integration of (3.12) which can

Table 3.5: Summary of the continuous-discrete extended Kalman filter

Nonlinear system model	$\dot{\mathbf{x}}(t) = \mathbf{A_0}\mathbf{x}(t) + \mathbf{b_0}u(t) + \mathbf{d_0}\ sin\left(\mathbf{C_0}^T\mathbf{x}(t)\right) + \omega(t)$ $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), u(t)) + \omega(t);\ \omega(t) \sim N(\mathbf{0}, \mathbf{Q})$
Measurement model	$\alpha_j \dot{v}_j(t) = -v_j(t) + \eta_j(t);\ j = 1, 2.$ $\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + v_k;\ k = 1, 2, \cdots$ $\eta \sim N(\mathbf{0}, 1);\ v_k = v(kT_s);$
State estimate propagation Error covariance propagation	$\dot{\hat{\mathbf{x}}}(t) = \mathbf{f}\left(\hat{\mathbf{x}}(t), u(t)\right);\ kT_s \leq t < (k+1)T_s$ $\dot{\mathbf{P}}(t) = \mathbf{F_1}\left(\hat{\mathbf{x}}(t)\right)\mathbf{P}(t) + \mathbf{P}(t)\mathbf{F_1}^T\left(\hat{\mathbf{x}}(t)\right) + \mathbf{Q}$
Kalman gain matrix Error covariance update State estimate update	$\mathbf{K}_k = \mathbf{P}_k(-)\mathbf{H}^T\left[\mathbf{H}\mathbf{P}_k(-)\mathbf{H}^T + \mathbf{R}_k\right]^{-1}$ $\mathbf{P}_k(+) = \left[\mathbf{I} - \mathbf{K}_k\mathbf{H}\right]\mathbf{P}_k(-)$ $\hat{\mathbf{x}}_k(+) = \hat{\mathbf{x}}_k(-) + \mathbf{K}_k\left[\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_k(-)\right]$
Definitions	$\mathbf{F_1}\left(\hat{\mathbf{x}}(t)\right) = \left.\frac{\partial \mathbf{f}(\mathbf{x}(t), u(t))}{\partial \mathbf{x}(t)}\right\|_{\mathbf{x}(t)=\hat{\mathbf{x}}(t)}$
Initial conditions Other assumptions	$\mathbf{x_0} \sim N(\hat{\mathbf{x}}_\mathbf{0}, \mathbf{P_0})$ $\xi\left\{\omega(t)v_k^T\right\} = 0\ \forall\ \ k\ \ \text{and}\ \ t$

be expressed by

$$\mathbf{x}(t) \ = \ \mathbf{x}(t_0) + \int_{t_i}^{t_f} \mathbf{f}\left[\mathbf{x}(\tau), \mathbf{u}(\tau)\right] d\tau$$

where the necessary quantities are assumed to be known in the finite interval and $\mathbf{x}(t)$ is instantaneously available to allow evaluation of $\mathbf{f}[\cdot]$. It is sufficient to tabulate $\mathbf{x}(t)$ at discrete instants of time, the integration algorithm could step from one instant of time to the next based on the previous result as an initial condition for the next time-increment.

$$\mathbf{x}(k) \ = \ \mathbf{x}(k-1) + \int_{k-1}^{k} \mathbf{f}\left[\mathbf{x}(t), \mathbf{u}(t)\right] dt$$

where k is the sampling index. The problem though the same as before, the time-increment $(k - (k-1))$ has been made arbitrarily small, allowing computational approximations based on multiplication and addition to substitute for the operation of integration.

Algorithms for numerical integration differ from each other in the way that $\mathbf{f}[\cdot]$ is assumed to vary in the interval. There is a trade off to be made between the length

of the time increment and the precision with which the variation in $\mathbf{f}[\cdot]$ is represented. If $\mathbf{f}[\cdot]$ is described precisely in the interval, large time steps can be used; if not, small steps are required.

Euler integration

Euler integration (also called as *rectangular integration*) provides the simplest representation of $\mathbf{f}[\cdot]$ (and thus solving for $\mathbf{x}(t)$) at discrete instants of time [Ste86]. With an integration step-size, $\Delta T = (k - (k-1))$,

$$\mathbf{x}(k) \;=\; \mathbf{x}(k-1) + \Delta T\ \mathbf{f}[\mathbf{x}(k-1), \mathbf{u}(k-1)] \tag{3.13}$$

This algorithm requires a single function evaluation - computation of $\mathbf{f}[\cdot]$ with given values of $\mathbf{x}$, $\mathbf{u}$ and t.

Modified Euler integration

Modified Euler or *trapezoidal integration* makes a correction based on the first function evaluation and the known variations in $\mathbf{x}$, $\mathbf{u}$ and t to improve the result. Incremental changes in the state are calculated as

$$\begin{aligned} \Delta\mathbf{x}_1 &= \Delta T\ \mathbf{f}[\mathbf{x}(k-1), \mathbf{u}(k-1)] \\ \Delta\mathbf{x}_2 &= \Delta T\ \mathbf{f}\{[\mathbf{x}(k-1) + \Delta\mathbf{x}_1], \mathbf{u}(k)\} \end{aligned}$$

and the final computation of $\mathbf{x}(k)$ is

$$\mathbf{x}(k) \;=\; \mathbf{x}(k-1) + \frac{1}{2}(\Delta\mathbf{x}_1 + \Delta\mathbf{x}_2)$$

As two function evaluations are required, it doubles the computation per time step but the accuracy of the result is improved and the time interval can be increased.

Runge-Kutta integration

The idea of using interim calculations of $\mathbf{f}[\cdot]$ to improve the fit in the interval is generalised in *Runge-Kutta integration.* For $n^{th} - order$ Runga-Kutta integration, n function

evaluations are made to produce n state increments; a weighted sum of the increments is used to update the state. The Euler and modified-Euler algorithms can be considered as Runga-Kutta methods of order one and two, respectively. The fourth-order algorithm provides a good trade-off between accuracy and computation; it can be expressed as follows [Ste86]:

$$\begin{aligned}
\Delta\mathbf{x}_1 &= \Delta T\ \mathbf{f}\left[\mathbf{x}(k-1), \mathbf{u}(k-1)\right] \\
\Delta\mathbf{x}_2 &= \Delta T\ \mathbf{f}\left\{\left[\mathbf{x}(k-1) + \frac{\Delta\mathbf{x}_1}{2}\right], \mathbf{u}\left(k-1\right)/2)\right\} \\
\Delta\mathbf{x}_3 &= \Delta T\ \mathbf{f}\left\{\left[\mathbf{x}(k-1) + \frac{\Delta\mathbf{x}_2}{2}\right], \mathbf{u}\left(k-1\right)/2)\right\} \\
\Delta\mathbf{x}_4 &= \Delta T\ \mathbf{f}\left\{\left[\mathbf{x}(k-1) + \Delta\mathbf{x}_3\right], \mathbf{u}(k)\right\}
\end{aligned}$$

where $(k-1)/2$ symbolises $t = (k-1) + \Delta T/2$ and the final computation of $\mathbf{x}(k)$ is

$$\mathbf{x}(k) = \mathbf{x}(k-1) + \frac{1}{6}\left(\Delta\mathbf{x}_1 + 2\Delta\mathbf{x}_2 + 2\Delta\mathbf{x}_3 + \Delta\mathbf{x}_4\right)$$

Based on the above discussion with respect to the numerical integration of state equations, the selection of a particular algorithm is a trade-off between accuracy and computational burden. For the propagation of the state equations in the EKF1 case, the Euler integration was selected because of its simplicity.

3.2.2 Computational issues

Kalman filtering is a real-time process, in the sense that the filter must do its work in the time between measurement inputs. One must then consider the amount of computation and thus the computational time required for each filter iteration. These computational requirements will, in turn, depend on the implementation method.

Computational complexity

The main computational burden in the EKF1 algorithm is in the state estimate and error covariance propagations. The error covariance propagation involves computing the Jacobian, $\mathbf{F_1}$ (see Table 3.5) using current estimates at each time instant. Also the state estimate propagation via Euler integration has to be done n times (where n being the number of propagations in each sampling interval).

Need for exact discrete-time filtering

As the propagations are done every ΔT_s seconds, the number of propagations are much more than what would have been in the case of propagating every T_s seconds. Also the numerical complexity and thus the computational time is increased by a significant factor. To overcome this disadvantage, the state estimate extrapolation and the error covariance extrapolation are desired to be done every T_s seconds. If a closed-form input-output relation of the model at discrete time instants is obtained, then this is possible instead of propagating every ΔT_s seconds.

Error covariance propagation

For small and constant sampling period, *i.e.* T_s *small*, $e^{\mathbf{A}T_s} \approx \mathbf{I} + \mathbf{A}T_s$ (where $\mathbf{I}$ indicates the identity matrix of appropriate order) and the state transition matrix,

$$\begin{aligned} \mathbf{\Phi}(k) &\approx \mathbf{\Phi}\left(k\Delta T_s, t_0\right);\ t_0 = k_0 \Delta T_s;\ \mathbf{\Phi}_0 = \mathbf{I}; \\ \mathbf{\Phi}(k+1) &\approx \left[\mathbf{I} + \Delta T_s \mathbf{F_1}(k\Delta T_s)\right] \mathbf{\Phi}(k) \end{aligned}$$

$\dot{\mathbf{P}}(t) = \mathbf{F_1}(t)\mathbf{P}(t) + \mathbf{P}(t)\mathbf{F_1}^T(t) + \mathbf{Q}$ is the error covariance propagation to be computed. The state space representation of the pendulum model is given by

$$\begin{aligned} \dot{x}_1(t) &= x_2(t) \\ \dot{x}_2(t) &= -\left(\frac{K_0}{J}\right) sin\ x_1(t) - \left(\frac{K_1}{J}\right) x_2(t) + \left(\frac{G_0}{J}\right) \tau(t) \\ &+ \left(\frac{1}{J}\right) \omega_0(t) \end{aligned}$$

which can be written simply as

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), u(t)) + \omega(t)$$

and the Jacobian is given by

$$\begin{aligned} \mathbf{F_1}\left(\hat{\mathbf{x}}(t)\right) &= \left.\frac{\partial \mathbf{f}(\mathbf{x}(t), u(t))}{\partial \mathbf{x}(t)}\right|_{\mathbf{x}(t)=\hat{\mathbf{x}}(t)} \\ &= \begin{pmatrix} 0 & 1 \\ -\frac{K_0}{J}\ cos\ \hat{x}_1(t) & -\frac{K_1}{J} \end{pmatrix} \end{aligned}$$

Thus $\mathbf{F_1}(t)$ and $\mathbf{Q}$ are available. It is desired to solve for $\mathbf{P}$. Using the Euler approximation and the state transition matrix, $\mathbf{\Phi}$, this can be done as follows:

Consider the matrix differential equation

$$\dot{\mathbf{\Phi}}(t, t_0) = \mathbf{A}(t)\mathbf{\Phi}(t, t_0) \tag{3.14}$$

The solution of (3.14) leads to

$$\mathbf{P}(t) = -\int_{t_0}^{t} \left[\mathbf{\Phi}(t_0, \sigma)\mathbf{Q}(\sigma)\mathbf{\Phi}^T(t_0, \sigma)\right] d\sigma$$

At discrete instants of time, this can be written as

$$\mathbf{P}(k+1) = -\int_{k}^{k+1} \left[\mathbf{\Phi}(k, \sigma)\mathbf{Q}(\sigma)\mathbf{\Phi}^T(k, \sigma)\right] d\sigma \tag{3.15}$$

$$\mathbf{P}(k) = -\int_{k-1}^{k} \left[\mathbf{\Phi}(k, \sigma)\mathbf{Q}(\sigma)\mathbf{\Phi}^T(k, \sigma)\right] d\sigma \tag{3.16}$$

(3.15) - (3.16) gives,

$$\mathbf{P}(k+1) - \mathbf{P}(k) = -\int_{k}^{k+1} \left[\mathbf{\Phi}(k, \sigma)\mathbf{Q}(\sigma)\mathbf{\Phi}^T(k, \sigma)\right] d\sigma$$

$$\mathbf{P}(k+1) = \mathbf{P}(k) - \Delta T_s \left[\mathbf{\Phi}(k-1, k)\mathbf{Q}(k)\mathbf{\Phi}^T(k-1, k)\right] \tag{3.17}$$

(3.17) is used for the error covariance propagation in the EKF1 approach.

3.2.3 Application of EKF1 to the pendulum model

The EKF1 is employed for the state estimation of the pendulum model and the results are presented for all the possible combinations of the sensors discussed in chapter 2. As indicated earlier, the noise sequences and the measurements are the same for this case as in the LKF case. Also note that this EKF works with the full nonlinear model. i.e., the nonlinearity $sin\ \theta$ is not approximated.

The (transient) state estimation error plots (when the measurements are provided by the encoder) are shown in Figure 3.5. Also the estimation error variances for each sensor combination are calculated and are tabulated in Table 3.6. A sampling interval of 0.2 seconds and an Euler integration step-size of 0.01 seconds were chosen for the simulation.

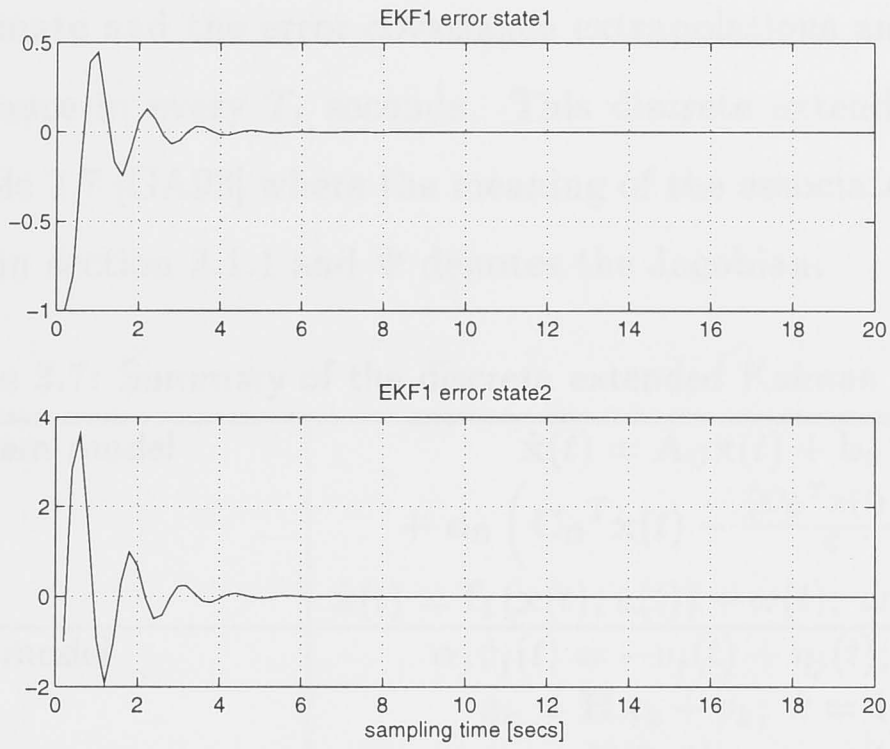


Figure 3.5: EKF1 state estimation errors with encoder measurements

Table 3.6: Estimation error variances in the EKF1 case

Sensor	Estimation error variance
Encoder	0.0218, 0.3267
Rate-gyro	0.0219, 0.3113
Encoder & rate-gyro	0.0217, 0.3082
Tilt-sensor	0.0220, 0.3269, 0.0000
Rate-gyro & tilt-sensor	0.0220, 0.3113, 0.0000

3.3 Extended Kalman Filtering of Polynomic Systems

In this section, the extended Kalman filtering of polynomic nonlinear systems subject to PAM control is considered. The nonlinear input-output equations are developed at discrete-time instants kT_s for the pendulum model. The results are then extended to the case where the model includes the tilt-sensor in which case the order of the combined model becomes equal to three. The EKF2 and EKF3 approaches are proposed for the state estimation of the pendulum and the combined pendulum and tilt-sensor models based on the afore-mentioned discrete-time equations where it is shown that these approaches are numerically efficient than the EKF1 approach. The results are provided for each sensor and are compared with the corresponding EKF1 case with respect to estimation accuracy and computational speed. In the EKF2 and EKF3 approaches,

both the state estimate and the error covariance extrapolations and the measurement updates are done once in every T_s seconds. This discrete extended Kalman filter is summarised in Table 3.7 [GA93] where the meaning of the associated terms remain the same as described in section 3.1.1 and Ψ denotes the Jacobian.

Table 3.7: Summary of the discrete extended Kalman filter

Nonlinear system model	$\dot{\mathbf{x}}(t) = \mathbf{A_0}\mathbf{x}(t) + \mathbf{b_0}u(t)$ $+ \mathbf{d_0}\left(\mathbf{C_0}^T\mathbf{x}(t) - \frac{\left(\mathbf{C_0}^T\mathbf{x}(t)\right)^3}{6}\right) + \omega(t)$ $\dot{\mathbf{x}}(t) = \mathbf{f_1}(\mathbf{x}(t), u(t)) + \omega(t);\ \omega(t) \sim N(\mathbf{0}, \mathbf{Q})$
Measurement model	$\alpha_j \dot{v}_j(t) = -v_j(t) + \eta_j(t);\ j = 1, 2.$ $\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + v_k;\ k = 1, 2, \cdots$ $\eta \sim N(\mathbf{0}, 1);\ v_k = v(kT_s);$
State estimate extrapolation	$\hat{\mathbf{x}}_k(-) = \mathbf{f_1}(\hat{\mathbf{x}}_{k-1}(+),\ u_{k-1})$
Error covariance extrapolation	$\mathbf{P}_k(-) = \Psi_{k-1}\mathbf{P}_{k-1}(+)\,\Psi_{k-1}^T + \mathbf{Q}_k$
Kalman gain matrix	$\mathbf{K}_k = \mathbf{P}_k(-)\,\mathbf{H}^T\left[\mathbf{H}\mathbf{P}_k(-)\,\mathbf{H}^T + \mathbf{R}_k\right]^{-1}$
Error covariance update	$\mathbf{P}_k(+) = [\mathbf{I} - \mathbf{K}_k\mathbf{H}]\,\mathbf{P}_k(-)$
State estimate update	$\hat{\mathbf{x}}_k(+) = \hat{\mathbf{x}}_k(-) + \mathbf{K}_k\left[\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_k(-)\right]$
Definitions	$\Psi_{k-1} = \left.\frac{\partial \mathbf{f_1}(\mathbf{x}_{k-1})}{\partial \mathbf{x}}\right\|_{\mathbf{x}=\hat{\mathbf{x}}_{k-1}(-)}$
Initial conditions	$\mathbf{x}_0 \sim N(\hat{\mathbf{x}}_0, \mathbf{P_0})$
Other assumptions	$\xi\left\{\omega(t)v_k^T\right\} = 0\ \forall\ \ k\ \text{ and }\ t$

3.3.1 Extended generalised Wiener model

It has been shown that [SP78] practically all algorithms designed to identify separable nonlinear systems could be classified as follows:

i. a nonlinear element followed by a linear element (NL-L),

ii. a linear element followed by a nonlinear element (L-NL) or

iii. a linear element followed by a nonlinear element which is in turn followed by either the same or another linear element (L-NL-L).

The first of these is often referred to as a *Hammerstein model* and the second as a *Wiener model*. The last case may be thought of as a *generalised Wiener model* [PM82]. The analysis and estimation of a large class of nonlinear systems that arise in control and

communication networks can be represented as the interconnection of linear subsystems and polynomial type nonlinearities and is dependent on a mathematical description between the system input and the output. Such systems fall under what is called the *extended generalised Wiener model* [San77] structure. These systems constitute a class of nonlinear systems admitting finite Volterra series representation. In estimating the states of such nonlinear systems, the tensor techniques along with the Volterra series and the discrete extended Kalman filtering serve as a powerful tool. These systems can be modelled from the input-output point of view as a combination of a linear system $H(s)$ followed by a polynomial-type nonlinearity and another linear system $G(s)$. A case where the nonlinearity is a cubic law is shown in Figure 3.6 which has the following state-space representation:

$$\begin{aligned} \dot{\mathbf{x}}_1(t) &= \mathbf{A_1}\mathbf{x_1}(t) + \mathbf{B_1}\mathbf{u}(t);\ \sigma(t) = \mathbf{C_1}^T\mathbf{x_1}(t) && (3.18)\\ \gamma(t) &= \mathbf{d_3}\sigma^{[3]}(t) \\ \dot{\mathbf{x}}_2(t) &= \mathbf{A_2}\mathbf{x_2}(t) + \mathbf{B_2}\gamma(t) && (3.19)\\ \mathbf{y}(t) &= \mathbf{C_1}^T\mathbf{x_1}(t) + \mathbf{C_2}^T\mathbf{x_2}(t) \end{aligned}$$

where $\mathbf{x_1}(t) \in \mathcal{R}^n$, $\mathbf{x_2}(t) \in \mathcal{R}^m$, $\mathbf{u}(t) \in \mathcal{R}^\ell$, $\gamma(t) \in \mathcal{R}^r$, $\sigma(t) \in \mathcal{R}^q$, $\mathbf{y}(t) \in \mathcal{R}^k$, $H(s) = \mathbf{C_1}(s\mathbf{I} - \mathbf{A_1})^{-1}\mathbf{B_1}$ and $G(s) = \mathbf{C_2}(s\mathbf{I} - \mathbf{A_2})^{-1}\mathbf{B_2}$ and the matrices have appropriate dimension and without loss of generality constitute minimal realizations of $H(s)$ and $G(s)$. The term $\sigma^{[3]}$ represents a vector homogenous form.

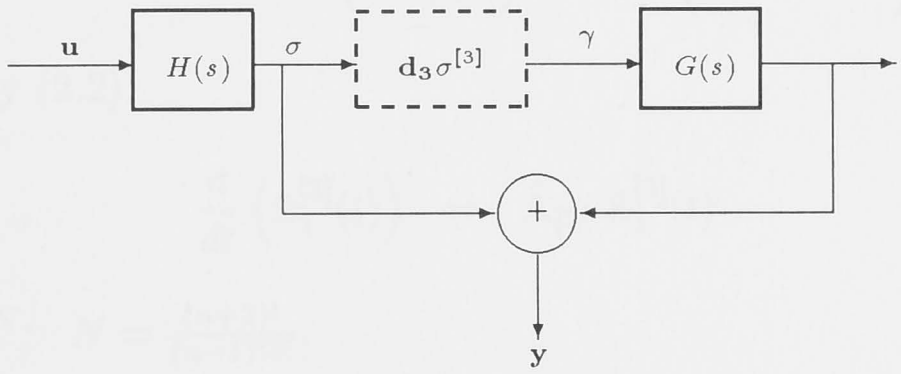


Figure 3.6: Extended generalised Wiener model representation

Such systems arise as a natural extension to work done previously by other authors [Gar73], [BR75], [SJ75], [San77] and [Van79]. For the input-output modelling of physical processes, this form of representation allows various types of cross-couplings between system variables to be included in the model. Also this model provides a low-order open-loop approximation to certain classes of nonlinear systems of infinite Volterra order [SW78]. There exists a nonlinear difference equation from $\mathbf{y} \rightarrow \mathbf{u}$ in terms of $\mathbf{u}(k)$ where $\mathbf{u}(k)$ is a PAM signal (see section 2.4). This equation will then simplify the state and error covariance extrapolations in the discrete extended Kalman filtering (EKF2 and EKF3) approaches.

In Figure 3.6, the nonlinearity is

$$\begin{aligned} \gamma(t) &= \mathbf{d_3 C_1}^{[3]^T} \mathbf{x_1}^{[3]}(t) \\ &= \mathbf{g}^{[3]^T} \mathbf{x_1}^{[3]}(t) \\ \mathbf{g}^{[3]^T} &\triangleq \mathbf{d_3 C_1}^{[3]^T} \end{aligned} \tag{3.20}$$

Writing the augmented system over the time interval $kT_s \leq t < (k+1)T_s$ and utilising (2.51),

$$\dot{\hat{\mathbf{x}}}_\mathbf{1}(t) = \hat{\mathbf{A}}\, \hat{\mathbf{x}}_\mathbf{1}(t)$$

$$\text{where } \hat{\mathbf{x}}_\mathbf{1}(\mathrm{t}) \triangleq \begin{pmatrix} \mathbf{x_1}(t) \\ \mathbf{u}(t) \end{pmatrix};\ \hat{\mathbf{A}} \triangleq \begin{pmatrix} \mathbf{A_1} & \mathbf{B_1} \\ 0 & 0 \end{pmatrix}$$

Thus from property (2.2),

$$\frac{d}{dt}\left(\hat{\mathbf{x}}_\mathbf{1}^{[3]}(t)\right) = \hat{\mathbf{A}}_{[3]}\, \hat{\mathbf{x}}_\mathbf{1}^{[3]}(t) \tag{3.21}$$

where $\hat{\mathbf{x}}_\mathbf{1}^{[3]}(t) \in \mathcal{R}^N$; $N = \frac{(n+2)!}{(n-1)!3!}$.

From (3.19) and (3.20),

$$\dot{\mathbf{x}}_\mathbf{2}(t) = \mathbf{A_2 x_2}(t) + \mathbf{B_2 g}^{[3]^T} \mathbf{x_1}^{[3]}(t)$$

$$\begin{aligned} \dot{\mathbf{x}}_2(t) &= \mathbf{A_2}\mathbf{x_2}(t) + \mathbf{B_2}\mathbf{h}^{[3]^T}\hat{\mathbf{x}}_1^{[3]}(t) \\ \mathbf{h}^{[3]^T}\hat{\mathbf{x}}_1^{[3]}(t) &\triangleq \mathbf{g}^{[3]^T}\mathbf{x}_1{}^{[3]}(t) \end{aligned} \tag{3.22}$$

Combining (3.21) and (3.22),

$$\dot{\mathbf{z}}(t) = \mathbf{F}\,\mathbf{z}(t) \tag{3.23}$$

where

$$\mathbf{z}(t) \triangleq \begin{pmatrix} \hat{\mathbf{x}}_1^{[3]}(t) \\ \mathbf{x_2}(t) \end{pmatrix}; \quad \mathbf{F} \triangleq \begin{pmatrix} \hat{\mathbf{A}}_{[3]} & \mathbf{0} \\ \mathbf{B_2}\mathbf{h}^{[3]^T} & \mathbf{A_2} \end{pmatrix} \tag{3.24}$$

The solution of (3.23) at discrete-time instants $t = kT_s$ is given by

$$\mathbf{z}(k+1) = e^{\mathbf{F}T_s}\mathbf{z}(k) \tag{3.25}$$

where $e^{\mathbf{F}T_s}$ has the form

$$e^{\mathbf{F}T_s} = \begin{pmatrix} e^{\hat{\mathbf{A}}_{[3]}T_s} & \mathbf{0} \\ \mathbf{M}(T_s) & e^{\mathbf{A_2}T_s} \end{pmatrix}$$

for some $m \times N$ matrix $\mathbf{M}(T_s)$. Thus, assuming $\mathbf{A_1^{-1}}$ exists,

$$\begin{aligned} \mathbf{x_2}(k+1) &= e^{\mathbf{A_2}T_s}\mathbf{x_2}(k) + \mathbf{M}(T_s)\hat{\mathbf{x}}_1^{[3]}(k) \\ \mathbf{x_1}(k+1) &= e^{\mathbf{A_1}T_s}\mathbf{x_1}(k) + \mathbf{u}(k)\mathbf{A_1}^{-1}\left[e^{\mathbf{A_1}T_s} - \mathbf{I}_n\right]\mathbf{B_1} \\ \mathbf{y}(k) &= \mathbf{C_1}^T\mathbf{x_1}(k) + \mathbf{C_2}^T\mathbf{x_2}(k) \end{aligned} \tag{3.26}$$

define the behaviour of (3.18) and (3.19) at the time instants $t = kT_s$.

An illustrative example

Consider the system shown in Figure 3.7 with $u(t)$ subject to (2.51) and $x_1 \in \mathcal{R}$ and $x_2 \in \mathcal{R}$ to be the states.

Then

$$\hat{\mathbf{x}}_1 = \begin{pmatrix} x_1 \\ u \end{pmatrix}; \quad \hat{\mathbf{x}}_1^{[3]}(k) = \begin{pmatrix} x_1^3(k) \\ \sqrt{3}x_1^2(k)u(k) \\ \sqrt{3}x_1(k)u^2(k) \\ u^3(k) \end{pmatrix};$$

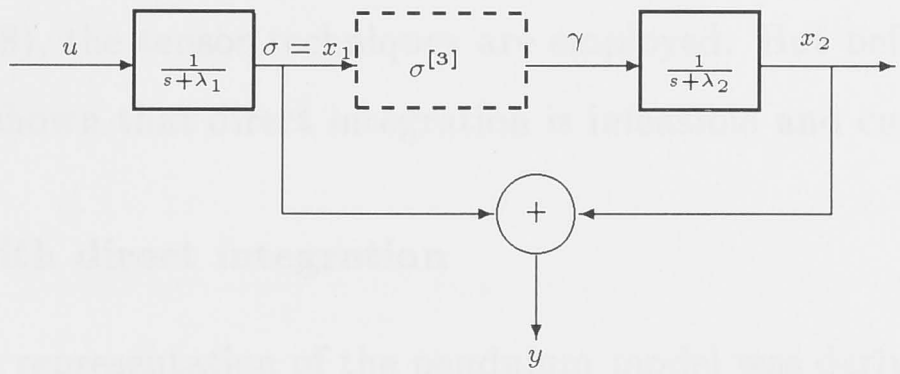


Figure 3.7: An illustrative example

and for some $\mathbf{M}(T_s)$ of the form,

$$\begin{pmatrix} m_1 & m_2 & m_3 & m_4 \end{pmatrix}$$

the discrete input-output relationship is

$$\begin{aligned} x_1(k+1) &= e^{-\lambda_1 T_s} x_1(k) + u(k)\left(1 - e^{-\lambda_1 T_s}\right) \\ x_2(k+1) &= e^{-\lambda_2 T_s} x_2(k) + m_1 x_1^3(k) + m_4 u^3(k) \\ &+ \sqrt{3} m_2 x_1^2(k) u(k) + \sqrt{3} m_3 x_1(k) u^2(k) \\ y(k) &= x_1(k) + x_2(k) \end{aligned}$$

Digital computer implementation

The vector $\hat{\mathbf{x}}^{[p]}$ and the matrix $\hat{\mathbf{A}}_{[p]}$ can be calculated numerically. The computation of $\hat{\mathbf{x}}^{[p]}$ and the associated linear transformation become significant for large n and/or p and in these cases the dimension of these quantities precludes hand computation. An ordering algorithm which generates a standard lexicographical order for $\hat{\mathbf{x}}^{[p]}$, $\mathbf{x} \in \mathcal{R}^n$ and an algorithm for the generation of the matrix $\hat{\mathbf{A}}_{[p]}$, for $\mathbf{A}$ $n \times n$ can be found in [San77].

It can thus be seen that the pendulum model has the same type of nonlinearity that was considered in this section. The Volterra series representation of the pendulum was derived in section 2.4 of chapter 2. It can be clearly seen that an approximate model of the pendulum has the extended generalised Wiener model structure. To avoid solving

for (2.57) and (2.58), the tensor techniques are employed. But before proceeding any further, it will be shown that direct integration is infeasible and cumbersome.

A comparison with direct integration

The Volterra series representation of the pendulum model was derived in section 2.4 of chapter 2. The associated variables technique [Geo59] is cumbersome. Direct integration is generally impractical and this will be demonstrated for the pendulum case as follows:

In (2.57), consider the deterministic linear part. i.e.,

$$\ddot{\theta}_1(t) = -\left(\frac{K_0}{J}\right)\theta_1(t) - \left(\frac{K_1}{J}\right)\dot{\theta}_1(t) + \left(\frac{G_0}{J}\right)u(t)$$

Let $x_{11}(t) \triangleq \theta_1(t)$ and $x_{12}(t) \triangleq \dot{\theta}_1(t)$. Then

$$\begin{aligned}\dot{x}_{11}(t) &= x_{12}(t)\\ \dot{x}_{12}(t) &= -\left(\frac{K_0}{J}\right)x_{11}(t) - \left(\frac{K_1}{J}\right)x_{12}(t) + u(t)\end{aligned}$$

for which the state space matrices are,

$$\mathbf{A}_\ell = \begin{pmatrix} 0 & 1 \\ -\frac{K_0}{J} & -\frac{K_1}{J} \end{pmatrix};\ \mathbf{B}_\ell = \begin{pmatrix} 0 & \frac{G_0}{J} \end{pmatrix}^T \tag{3.27}$$

Using the variation of constants formula, (see section 2.3.1),

$$\mathbf{x_1}(\sigma) = e^{\mathbf{A}_\ell(\sigma - kT_s)}\mathbf{x_1}(k) + \int_{kT_s}^{\sigma}\left\{e^{\mathbf{A}_\ell(\sigma-\gamma)}\ \mathbf{B}_\ell\ u(\gamma)\right\}\ d\gamma$$

and with a PAM input signal,

$$\begin{aligned}\mathbf{x_1}(\sigma) &= e^{\mathbf{A}_\ell(\sigma-kT_s)}\ \mathbf{x_1}(k) + u(k)\ \int_{kT_s}^{\sigma} e^{\mathbf{A}_\ell(\sigma-\gamma)}\ d\gamma\\ &= e^{\mathbf{A}_\ell(\sigma-kT_s)}\ \mathbf{x_1}(k) + u(k)\ e^{\mathbf{A}_\ell\sigma}\ \int_{kT_s}^{\sigma}\left(e^{-\mathbf{A}_\ell\gamma}\ \mathbf{B}_\ell\right)\ d\gamma\\ &= e^{\mathbf{A}_\ell(\sigma-kT_s)}\ \mathbf{x_1}(k) + u(k)\ e^{\mathbf{A}_\ell\sigma}\ (-\mathbf{A}_\ell{}^{-1})\ e^{-\mathbf{A}_\ell\gamma}\Big|_{kT_s}^{\sigma}\mathbf{B}_\ell\\ &= e^{\mathbf{A}_\ell(\sigma-kT_s)}\ \mathbf{x_1}(k) + u(k)\ (-\mathbf{A}_\ell{}^{-1})\ \left\{\mathbf{I}_2 - e^{\mathbf{A}_\ell(\sigma-kT_s)}\right\}\mathbf{B}_\ell\\ &= e^{\mathbf{A}_\ell(\sigma-kT_s)}\ \mathbf{x_1}(k) + u(k)\mathbf{A}_\ell{}^{-1}\left(e^{\mathbf{A}_\ell(\sigma-kT_s)} - \mathbf{I}_2\right)\mathbf{B}_\ell \text{ and thus}\\ \mathbf{x_1}(k+1) &= e^{\mathbf{A}_\ell T_s}\ \mathbf{x_1}(k) + u(k)\mathbf{A}_\ell{}^{-1}\left(e^{\mathbf{A}_\ell T_s} - \mathbf{I}_2\right)\mathbf{B}_\ell\end{aligned}$$

where $\mathbf{A}_\ell$ and $\mathbf{B}_\ell$ are 2×2 and 2×1 matrices as defined in (3.27) and $\mathbf{I}_2$ is an identity matrix of order 2.

In (2.58), consider the deterministic nonlinear part. i.e.,

$$\ddot{\theta}_3(t) = -\left(\frac{K_0}{J}\right)\theta_3(t) - \left(\frac{K_1}{J}\right)\dot{\theta}_3(t) + \left(\frac{K_0}{6J}\right)\left[\mathbf{C}^T\mathbf{x_1}(\sigma)\right]^3 d\sigma$$

where $x_{11}(\sigma)$ has been written as $\mathbf{C}^T\mathbf{x_1}(\sigma)$ and $\mathbf{C}^T = (1 \;\; 0)$.

Let $x_{31}(t) \triangleq \theta_3(t)$ and $x_{32}(t) \triangleq \dot{\theta}_3(t)$. Then

$$\begin{aligned}
\dot{x}_{31}(t) &= x_{32}(t) \\
\dot{x}_{32}(t) &= -\left(\frac{K_0}{J}\right)x_{31}(t) - \left(\frac{K_1}{J}\right)x_{32}(t) + \left(\frac{K_0}{6J}\right)\left[\mathbf{C}^T\mathbf{x_1}(\sigma)\right]^3 \quad \text{for which} \\
\mathbf{A_{n\ell}} &= \begin{pmatrix} 0 & 1 \\ -\frac{K_0}{J} & -\frac{K_1}{J} \end{pmatrix}; \; \mathbf{B_{n\ell}} = \begin{pmatrix} 0 & \frac{K_0}{6J} \end{pmatrix}^T
\end{aligned} \tag{3.28}$$

Proceeding as with the linear part,

$$\begin{aligned}
\mathbf{x_3}(k+1) &= e^{\mathbf{A_{n\ell}}(\sigma - kT_s)}\mathbf{x_3}(k) \\
&+ \int_{kT_s}^{(k+1)T_s} \left\{ e^{\mathbf{A_{n\ell}}(kT_s + T_s - \sigma)}\mathbf{B_{n\ell}}\left[\mathbf{C}^T\mathbf{x_1}(\sigma)\right]^3 \right\} d\sigma
\end{aligned} \tag{3.29}$$

First evaluate $x_{11}^3(\sigma)$

$$\begin{aligned}
x_{11}^3(\sigma) &= \left\{\mathbf{C}^T\mathbf{x_1}(\sigma)\right\}^3 \\
&= \left\{\mathbf{C}^T\left[e^{\mathbf{A}_\ell(\sigma - kT_s)}\,\mathbf{x_1}(k) + u(k)\mathbf{A}_\ell{}^{-1}\left(e^{\mathbf{A}_\ell(\sigma - kT_s)} - \mathbf{I}_2\right)\mathbf{B}_\ell\right]\right\}^3 \\
&= T_1^3 + T_2^3 + 3T_1^2T_2 + 3T_1T_2^2
\end{aligned}$$

where

$$\begin{aligned}
T_1 &= \mathbf{C}^T e^{\mathbf{A_{n\ell}}(\sigma - kT_s)}\,\mathbf{x_1}(k) \\
T_2 &= \mathbf{C}^T u(k)\mathbf{A_{n\ell}}{}^{-1}\left(e^{\mathbf{A_{n\ell}}(\sigma - kT_s)} - \mathbf{I}_2\right)\mathbf{B_{n\ell}}
\end{aligned}$$

Let

$$I_1 = \int_{kT_s}^{(k+1)T_s} \left\{ e^{\mathbf{A_{n\ell}}(kT_s + T_s - \sigma)}\mathbf{B_{n\ell}}\left(T_1^3 + T_2^3 + 3T_1^2T_2 + 3T_1T_2^2\right)\right\} d\sigma$$

Solving for (3.29) involves the computation of the integral I_1. It can be attempted to compute the integral I_1 by either integration by parts or by some other method but this becomes tedious and cumbersome as it is evident. Also as the order of the model increases, such integrals get too complicated to solve. Thus the versatility of the tensor theory approach that was developed becomes evident. The advantage of this method, in addition to the applicability to models of higher order and reduced complexity, is that it uses state-space representation and thus can be easily implemented on a digital computer (for the digital computer implementation, see the beginning of this section).

3.3.2 Application of EKF2 and EKF3 to the pendulum model

Discrete extended Kalman filtering algorithm

a. Pendulum case

For this case, $n = p = 3$ and from (2.41),

$$N_3^3 \triangleq \frac{(3+3-1)\,!}{(3-1)\,!\,3\,!} = 10.$$

From definition 2.42, the p sets in the appropriate order are

$$\left\{ \begin{array}{lllll} \{3,0,0\} & \{2,1,0\} & \{2,0,1\} & \{1,2,0\} & \{1,1,1\} \\ \{1,0,2\} & \{0,3,0\} & \{0,2,1\} & \{0,1,2\} & \{0,0,3\} \end{array} \right\} \tag{3.30}$$

Define

$$\mathbf{x_1} = \begin{pmatrix} \theta_1 \\ \dot{\theta}_1 \end{pmatrix} = \begin{pmatrix} x_{11} \\ x_{12} \end{pmatrix}$$

$$\mathbf{x_2} = \begin{pmatrix} \theta_3 \\ \dot{\theta}_3 \end{pmatrix} = \begin{pmatrix} x_{21} \\ x_{22} \end{pmatrix}$$

where in (3.30), $\{3,0,0\}$ means x_{11}^3 and $\{1,1,1\}$ means $x_{11}x_{12}u$ and so on. The choice of normalising coefficients is governed by (2.43).

Therefore from (3.24),

$$\mathbf{z}(k) = \begin{pmatrix} \begin{bmatrix} x_{11}(k) \\ x_{12}(k) \\ u(k) \end{bmatrix}^{[3]} \\ x_{21}(k) \\ x_{22}(k) \end{pmatrix} = \begin{pmatrix} x_{11}^3(k) \\ \sqrt{3}x_{11}^2(k)x_{12}(k) \\ \sqrt{3}x_{11}^2(k)u(k) \\ \sqrt{3}x_{11}(k)x_{12}^2(k) \\ \sqrt{6}x_{11}(k)x_{12}(k)u(k) \\ \sqrt{3}x_{11}(k)u^2(k) \\ x_{12}^3(k) \\ \sqrt{3}x_{12}^3(k)u(k) \\ \sqrt{3}x_{12}(k)u^2(k) \\ u^3(k) \\ x_{21}(k) \\ x_{22}(k) \end{pmatrix}$$

where in (3.18), (3.19)

$$\begin{aligned} \mathbf{A_1} &= \mathbf{A_2} = \begin{pmatrix} 0 & 1 \\ -\frac{K_0}{J} & -\frac{K_1}{J} \end{pmatrix}; \\ \mathbf{B_1} &= \begin{pmatrix} 0 & \frac{G_0}{J} \end{pmatrix}^T; \ \mathbf{B_2} = \begin{pmatrix} 0 & \frac{K_0}{6J} \end{pmatrix}^T; \\ \mathbf{h}^{[3]T} &= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}; \\ \mathbf{C_1}^T &= \mathbf{C_2}^T = \begin{pmatrix} 1 & 1 \end{pmatrix}; \end{aligned} \tag{3.31}$$

and thus $\mathbf{B_2}\mathbf{h}^{[3]T}$ is of order 2×10. $\mathbf{M}(T_s)$ is a 2×10 matrix as indicated in section 2.4 which has the following structure

$$\mathbf{M}(T_s) = \begin{pmatrix} m_{1,1} & m_{2,1} \\ m_{1,2} & m_{2,2} \\ \vdots & \vdots \\ m_{1,10} & m_{2,10} \end{pmatrix}^T \tag{3.32}$$

The $\hat{\mathbf{A}}$ and the $\mathbf{F}$ matrices for the pendulum model are

$$\hat{\mathbf{A}} = \begin{pmatrix} 0 & 1 & 0 \\ -\frac{K_0}{J} & -\frac{K_1}{J} & \frac{G_0}{J} \\ 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{F} = \begin{pmatrix} 0 & \sqrt{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\sqrt{3}\frac{K_0}{J} & -\frac{K_1}{J} & \frac{G_0}{J} & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -2\frac{K_0}{J} & 0 & -2\frac{K_1}{J} & \sqrt{2}\frac{G_0}{J} & 0 & \sqrt{3} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\sqrt{2}\frac{K_0}{J} & 0 & -\frac{K_1}{J} & \sqrt{2}\frac{G_0}{J} & 0 & \sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\sqrt{3}\frac{K_0}{J} & 0 & 0 & -3\frac{K_1}{J} & \sqrt{3}\frac{G_0}{J} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\sqrt{2}\frac{K_0}{J} & 0 & 0 & -2\frac{K_1}{J} & 2\frac{G_0}{J} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -\frac{K_0}{J} & 0 & 0 & -\frac{K_1}{J} & \sqrt{3}\frac{G_0}{J} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \frac{K_0}{6J} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{K_0}{J} & -\frac{K_1}{J} \end{pmatrix}$$

EKF3 approach

The $\mathbf{M}(T_s)$ matrix is dependent on the sampling time, T_s (the argument T_s is used to indicate the explicit dependence of $\mathbf{M}(T_s)$ on T_s). Components of $\mathbf{M}(T_s)$ which are small relative to others can be neglected in order to reduce the number of nonlinear terms. The exclusion of these terms may have little effect on the estimation but results in a significant increase in the speed of the algorithm. For example, if only the six elements $\{m_{1,1}, m_{1,2}, m_{1,3}, m_{2,1}, m_{2,2}, m_{2,3}\}$ of the $\mathbf{M}(T_s)$ matrix are considered instead of the twenty elements and the value of the other elements are set to zero, the $\mathbf{M_a}$ matrix is obtained where the additional subscript a indicates the approximated matrix such that

$$\begin{aligned} \mathbf{M_a}(T_s)\hat{\mathbf{x}}_{\mathbf{1}}^{[3]} &\approx \begin{pmatrix} m_{1,1} & m_{2,1} \\ m_{1,2} & m_{2,2} \\ m_{1,3} & m_{2,3} \end{pmatrix}^T \begin{pmatrix} x_{11}^3(k) \\ \sqrt{3}x_{11}^2(k)x_{12}(k) \\ \sqrt{3}x_{11}^2(k)u(k) \end{pmatrix} \\ &= \begin{pmatrix} M_{21} & M_{22} \end{pmatrix}^T \end{aligned} \tag{3.33}$$

When the $\mathbf{M_a}$ matrix is employed for the estimation, this type of discrete extended Kalman filtering will be termed as the EKF3 approach. The justification for exclusion of certain elements of the $\mathbf{M}(T_s)$ matrix is based on the selection of a small sampling interval. To make the distinction much clearer, the values of each element of the $\mathbf{M}(T_s)$ matrix, for the particular sampling interval T_s considered for simulation, are tabulated in Tables 3.8 and 3.11 respectively. Note that the pendulum is considered in the down position only, whose equilibrium point is stable and not in the upright position whose

equilibrium point is unstable.

The results are tabulated, for comparison purposes, for two other sampling intervals in Tables 3.9 and 3.10, for the pendulum only case and in Tables 3.12 and 3.13, for the combined pendulum and tilt-sensor case. From the Tables 3.8, 3.9, 3.10, 3.11, 3.12 and 3.13, the justification for the exclusion of certain elements of the $\mathbf{M}(T_s)$ matrix is evident. A word of caution here would be that if the sampling interval becomes large enough to cause significant changes in the elements of the $\mathbf{M}(T_s)$ matrix, care should be exercised in excluding the elements.

Table 3.8: Elements of the $\mathbf{M}(T_s)$ matrix in the pendulum case for $T_s = 0.2$

Element	Value	Element	Value
$m_{1,1}$	0.0565	$m_{2,1}$	0.3359
$m_{1,2}$	0.0056	$m_{2,2}$	0.0559
$m_{1,3}$	0.0077	$m_{2,3}$	0.1101
$m_{1,4}$	0.0005	$m_{2,4}$	0.0072
$m_{1,5}$	0.0012	$m_{2,5}$	0.0235
$m_{1,6}$	0.0017	$m_{2,6}$	0.0416
$m_{1,7}$	0.0000	$m_{2,7}$	0.0006
$m_{1,8}$	0.0001	$m_{2,8}$	0.0028
$m_{1,9}$	0.0002	$m_{2,9}$	0.0074
$m_{1,10}$	0.0003	$m_{2,10}$	0.0118

b. Pendulum and tilt-sensor case

For this case, $n = p = 4$ and from (2.41),

$$N_4^4 \triangleq \frac{(4+4-1)\,!}{(4-1)\,!\,4\,!} = 35.$$

From definition 2.42, the p sets in the appropriate order are

$$\left\{\begin{array}{ccccc} \{4,0,0,0\} & \{2,1,1,0\} & \{2,0,1,1\} & \{0,2,1,1\} & \{0,4,0,0\} \\ \{0,0,1,3\} & \{0,0,4,0\} & \{3,1,0,0\} & \{3,0,1,0\} & \{2,0,2,0\} \\ \{3,0,0,1\} & \{1,3,0,0\} & \{2,2,0,0\} & \{1,0,3,0\} & \{1,0,0,3\} \\ \{2,0,0,2\} & \{0,3,1,0\} & \{0,2,2,0\} & \{0,1,3,0\} & \{0,1,0,3\} \\ \{0,3,0,1\} & \{0,2,0,2\} & \{2,1,0,1\} & \{0,0,3,1\} & \{0,0,2,2\} \\ \{1,2,1,0\} & \{1,1,2,0\} & \{1,2,0,1\} & \{1,1,0,2\} & \{1,0,2,1\} \\ \{1,0,1,2\} & \{0,1,2,1\} & \{0,1,1,2\} & \{1,1,1,1\} & \{0,0,0,4\} \end{array}\right\} \tag{3.34}$$

where $\{1,1,1,1\}$ denotes $x_{11}x_{12}x_{13}u$ and so on. Define

$$\mathbf{x_1} = \begin{pmatrix} \theta_1 \\ \dot{\theta}_1 \\ z_1 \end{pmatrix} = \begin{pmatrix} x_{11} \\ x_{12} \\ x_{13} \end{pmatrix}$$

$$\mathbf{x_2} = \begin{pmatrix} \theta_3 \\ \dot{\theta}_3 \\ z_3 \end{pmatrix} = \begin{pmatrix} x_{21} \\ x_{22} \\ x_{23} \end{pmatrix}$$

Therefore from (3.24),

$$\mathbf{z}(k) = \begin{pmatrix} \begin{bmatrix} x_{11}(k) \\ x_{12}(k) \\ x_{13}(k) \\ u(k) \end{bmatrix}^{[3]} \\ x_{21}(k) \\ x_{22}(k) \\ x_{23}(k) \end{pmatrix} \tag{3.35}$$

$$= \begin{pmatrix} x_{11}^4(k) \\ \sqrt{12}x_{11}^2(k)x_{12}(k)x_{13}(k) \\ \sqrt{12}x_{11}^2(k)x_{13}(k)u(k) \\ \sqrt{12}x_{12}^2(k)x_{13}(k)u(k) \\ x_{12}^4(k) \\ \sqrt{4}x_{13}(k)u^3(k) \\ x_{13}^4(k) \\ \sqrt{4}x_{11}^3(k)x_{12}(k) \\ \sqrt{4}x_{11}^3(k)x_{13}(k) \\ \sqrt{6}x_{11}^2(k)x_{13}^2(k) \\ \sqrt{4}x_{11}^3(k)u(k) \\ \sqrt{4}x_{11}(k)x_{12}^3(k) \\ \sqrt{6}x_{11}^2(k)x_{12}^2(k) \\ \sqrt{4}x_{11}(k)x_{13}^3(k) \\ \sqrt{4}x_{11}(k)u^3(k) \\ \sqrt{6}x_{11}^2(k)u^2(k) \\ \sqrt{4}x_{12}^3(k)x_{13}(k) \\ \sqrt{6}x_{12}^2(k)x_{13}^2(k) \\ \sqrt{4}x_{12}(k)x_{13}^3(k) \\ \sqrt{4}x_{12}(k)u^3(k) \\ \sqrt{4}x_{12}^3(k)u(k) \\ \sqrt{6}x_{12}^2(k)u^2(k) \\ \sqrt{12}x_{11}^2(k)x_{12}(k)u(k) \\ \sqrt{4}x_{13}^3(k)u(k) \\ \sqrt{6}x_{13}^2(k)u^2(k) \\ \sqrt{12}x_{11}(k)x_{12}^2(k)u(k) \\ \sqrt{12}x_{11}(k)x_{12}(k)x_{13}^2(k) \\ \sqrt{12}x_{11}(k)x_{12}^2(k)u(k) \\ \sqrt{12}x_{11}(k)x_{12}(k)u^2(k) \\ \sqrt{6}x_{11}(k)x_{13}^2(k)u(k) \\ \sqrt{12}x_{11}(k)x_{13}(k)u^2(k) \\ \sqrt{12}x_{12}(k)x_{13}^2(k)u(k) \\ \sqrt{12}x_{12}(k)x_{13}(k)u^2(k) \\ \sqrt{24}x_{11}(k)x_{12}(k)x_{13}(k)u(k) \\ u^4(k) \\ x_{21}(k) \\ x_{22}(k) \\ x_{23}(k) \end{pmatrix}$$

where in (3.18), (3.19)

$$\mathbf{A_1} = \mathbf{A_2} = \begin{pmatrix} 0 & 1 & 0 \\ -\frac{K_0}{J} & -\frac{K_1}{J} & 0 \\ \beta_{ts} & 0 & -\alpha_{ts} \end{pmatrix}; \tag{3.36}$$

$$\mathbf{B_1} = \begin{pmatrix} 0 & \frac{G_0}{J} & 0 \end{pmatrix}^T; \ \mathbf{B_2} = \begin{pmatrix} 0 & \frac{K_0}{6J} & 0 \end{pmatrix}^T;$$

$$\mathbf{C_1}^T = \mathbf{C_2}^T = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix};$$

$\mathbf{h}^{[3]^T}$ is of order 1×35 and thus $\mathbf{B_2}\mathbf{h}^{[3]^T}$ is of order 3×35. The $\hat{\mathbf{A}}$ matrix for the combined pendulum and tilt-sensor model is

$$\hat{\mathbf{A}} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -\frac{K_0}{J} & -\frac{K_1}{J} & 0 & \frac{G_0}{J} \\ \beta_{ts} & 0 & -\alpha_{ts} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

and the $\mathbf{F}$ matrix is of order 38×38 and the corresponding $\mathbf{M}(T_s)$ matrix is a 3×35 matrix as indicated in section 2.4 which has the following structure

$$\mathbf{M}(T_s) = \begin{pmatrix} m_{1,1} & m_{2,1} & m_{3,1} \\ m_{1,2} & m_{2,2} & m_{3,2} \\ \vdots & \vdots & \vdots \\ m_{1,35} & m_{2,35} & m_{3,35} \end{pmatrix}^T \tag{3.37}$$

As in the pendulum case, consider only the following twenty seven elements of the $\mathbf{M}$ matrix

$$\left\{ \begin{matrix} m(1,1) & m(1,8) & m(1,11) \\ m(1,15) & m(1,16) & m(1,20) \\ m(1,22) & m(1,29) & m(1,35) \\ m(2,1) & m(2,8) & m(2,11) \\ m(2,15) & m(2,16) & m(2,20) \\ m(2,22) & m(2,29) & m(2,35) \\ m(3,1) & m(3,8) & m(3,11) \\ m(3,15) & m(3,16) & m(3,20) \\ m(3,22) & m(3,29) & m(3,35) \end{matrix} \right\} \tag{3.38}$$

instead of the 105 elements. By employing this EKF3 approach, the computational burden is greatly reduced and the numerical efficiency is increased by a significant factor.

Results

The extended Kalman filters, EKF2 and EKF3 are employed for the state estimation of the pendulum model. The same measurements and noise sequences as in the LKF

Table 3.9: Elements of the $\mathrm{M}(T_s)$ matrix in the pendulum case for $T_s = 0.1$

Element	Value	Element	Value
$m_{1,1}$	0.0197	$m_{2,1}$	0.3441
$m_{1,2}$	0.0011	$m_{2,2}$	0.0287
$m_{1,3}$	0.0008	$m_{2,3}$	0.0278
$m_{1,4}$	0.0001	$m_{2,4}$	0.0018
$m_{1,5}$	0.0001	$m_{2,5}$	0.0028
$m_{1,6}$	0.0000	$m_{2,6}$	0.0023
$m_{1,7}$	0.0000	$m_{2,7}$	0.0001
$m_{1,8}$	0.0000	$m_{2,8}$	0.0002
$m_{1,9}$	0.0000	$m_{2,9}$	0.0002
$m_{1,10}$	0.0000	$m_{2,10}$	0.0001

Table 3.10: Elements of the $\mathrm{M}(T_s)$ matrix in the pendulum case for $T_s = 0.01$

Element	Value	Element	Value
$m_{1,1}$	0.0002	$m_{2,1}$	0.0462
$m_{1,2}$	0.0000	$m_{2,2}$	0.0004
$m_{1,3}$	0.0000	$m_{2,3}$	0.0000
$m_{1,4}$	0.0000	$m_{2,4}$	0.0000
$m_{1,5}$	0.0000	$m_{2,5}$	0.0000
$m_{1,6}$	0.0000	$m_{2,6}$	0.0000
$m_{1,7}$	0.0000	$m_{2,7}$	0.0000
$m_{1,8}$	0.0000	$m_{2,8}$	0.0000
$m_{1,9}$	0.0000	$m_{2,9}$	0.0000
$m_{1,10}$	0.0000	$m_{2,10}$	0.0000

and EKF1 cases are used for the simulation. The estimation error plots (when the measurements are provided by the encoder) for both the EKF2 and EKF3 approaches are shown in Figure 3.8. The estimation error variances for all the sensors are shown in Table 3.14. Comparing these results with that of the EKF1 (see section 3.2.3), it can be seen that the EKF2 and EKF3 perform as well as the EKF1 but the main difference is the computational speed (see section 3.3.3) and thus these approaches have the advantage

Table 3.11: Elements of the $\mathbf{M}(T_s)$ matrix in the pendulum and tilt-sensor case for $T_s = 0.2$

Element	Value	Element	Value	Element	Value
$m_{1,1}$	0.0491	$m_{2,1}$	0.1961	$m_{3,1}$	0.0004
$m_{1,2}$	0	$m_{2,2}$	0	$m_{3,2}$	0
$m_{1,3}$	0	$m_{2,3}$	0	$m_{3,3}$	0
$m_{1,4}$	0	$m_{2,4}$	0	$m_{3,4}$	0
$m_{1,5}$	0	$m_{2,5}$	0	$m_{3,5}$	0
$m_{1,6}$	0	$m_{2,6}$	0	$m_{3,6}$	0
$m_{1,7}$	0	$m_{2,7}$	0	$m_{3,7}$	0
$m_{1,8}$	0.0074	$m_{2,8}$	0.0801	$m_{3,8}$	0.0000
$m_{1,9}$	0	$m_{2,9}$	0	$m_{3,9}$	0
$m_{1,10}$	0	$m_{2,10}$	0	$m_{3,10}$	0
$m_{1,11}$	0.0108	$m_{2,11}$	0.1731	$m_{3,11}$	0.0000
$m_{1,12}$	0	$m_{2,12}$	0	$m_{3,12}$	0
$m_{1,13}$	0	$m_{2,13}$	0	$m_{3,13}$	0
$m_{1,14}$	0	$m_{2,14}$	0	$m_{3,14}$	0
$m_{1,15}$	0.0002	$m_{2,15}$	0.0071	$m_{3,15}$	0.0000
$m_{1,16}$	0.0019	$m_{2,16}$	0.0396	$m_{3,16}$	0.0000
$m_{1,17}$	0	$m_{2,17}$	0	$m_{3,17}$	0
$m_{1,18}$	0	$m_{2,18}$	0	$m_{3,18}$	0
$m_{1,19}$	0	$m_{2,19}$	0	$m_{3,19}$	0
$m_{1,20}$	0.0000	$m_{2,20}$	0.0008	$m_{3,20}$	0.0000
$m_{1,21}$	0.0000	$m_{2,21}$	0.0001	$m_{3,21}$	0.0000
$m_{1,22}$	0.0000	$m_{2,22}$	0.0003	$m_{3,22}$	0.0000
$m_{1,23}$	0.0007	$m_{2,23}$	0.0145	$m_{3,23}$	0.0000
$m_{1,24}$	0	$m_{2,24}$	0	$m_{3,24}$	0
$m_{1,25}$	0	$m_{2,25}$	0	$m_{3,25}$	0
$m_{1,26}$	0	$m_{2,26}$	0	$m_{3,26}$	0
$m_{1,27}$	0	$m_{2,27}$	0	$m_{3,27}$	0
$m_{1,28}$	0.0000	$m_{2,28}$	0.0011	$m_{3,28}$	0.0000
$m_{1,29}$	0.0002	$m_{2,29}$	0.0043	$m_{3,29}$	0.0000
$m_{1,30}$	0	$m_{2,30}$	0	$m_{3,30}$	0
$m_{1,31}$	0	$m_{2,31}$	0	$m_{3,31}$	0
$m_{1,32}$	0	$m_{2,32}$	0	$m_{3,32}$	0
$m_{1,33}$	0	$m_{2,33}$	0	$m_{3,33}$	0
$m_{1,34}$	0	$m_{2,34}$	0	$m_{3,34}$	0
$m_{1,35}$	0.0000	$m_{2,35}$	0.0012	$m_{3,35}$	0.0000

Table 3.12: Elements of the $\mathrm{M}(T_s)$ matrix in the pendulum and tilt-sensor case for $T_s = 0.1$

Element	Value	Element	Value	Element	Value
$m_{1,1}$	0.0192	$m_{2,1}$	0.3240	$m_{3,1}$	0.0001
$m_{1,2}$	0	$m_{2,2}$	0	$m_{3,2}$	0
$m_{1,3}$	0	$m_{2,3}$	0	$m_{3,3}$	0
$m_{1,4}$	0	$m_{2,4}$	0	$m_{3,4}$	0
$m_{1,5}$	0	$m_{2,5}$	0	$m_{3,5}$	0
$m_{1,6}$	0	$m_{2,6}$	0	$m_{3,6}$	0
$m_{1,7}$	0	$m_{2,7}$	0	$m_{3,7}$	0
$m_{1,8}$	0.0013	$m_{2,8}$	0.0353	$m_{3,8}$	0.0000
$m_{1,9}$	0	$m_{2,9}$	0	$m_{3,9}$	0
$m_{1,10}$	0	$m_{2,10}$	0	$m_{3,10}$	0
$m_{1,11}$	0.0009	$m_{2,11}$	0.0347	$m_{3,11}$	0.0000
$m_{1,12}$	0	$m_{2,12}$	0	$m_{3,12}$	0
$m_{1,13}$	0	$m_{2,13}$	0	$m_{3,13}$	0
$m_{1,14}$	0	$m_{2,14}$	0	$m_{3,14}$	0
$m_{1,15}$	0.0000	$m_{2,15}$	0.0002	$m_{3,15}$	0.0000
$m_{1,16}$	0.0001	$m_{2,16}$	0.0043	$m_{3,16}$	0.0000
$m_{1,17}$	0	$m_{2,17}$	0	$m_{3,17}$	0
$m_{1,18}$	0	$m_{2,18}$	0	$m_{3,18}$	0
$m_{1,19}$	0	$m_{2,19}$	0	$m_{3,19}$	0
$m_{1,20}$	0.0000	$m_{2,20}$	0.0000	$m_{3,20}$	0.0000
$m_{1,21}$	0.0000	$m_{2,21}$	0.0000	$m_{3,21}$	0.0000
$m_{1,22}$	0.0000	$m_{2,22}$	0.0000	$m_{3,22}$	0.0000
$m_{1,23}$	0.0000	$m_{2,23}$	0.0015	$m_{3,23}$	0.0000
$m_{1,24}$	0	$m_{2,24}$	0	$m_{3,24}$	0
$m_{1,25}$	0	$m_{2,25}$	0	$m_{3,25}$	0
$m_{1,26}$	0	$m_{2,26}$	0	$m_{3,26}$	0
$m_{1,27}$	0	$m_{2,27}$	0	$m_{3,27}$	0
$m_{1,28}$	0.0000	$m_{2,28}$	0.0001	$m_{3,28}$	0.0000
$m_{1,29}$	0.0000	$m_{2,29}$	0.0002	$m_{3,29}$	0.0000
$m_{1,30}$	0	$m_{2,30}$	0	$m_{3,30}$	0
$m_{1,31}$	0	$m_{2,31}$	0	$m_{3,31}$	0
$m_{1,32}$	0	$m_{2,32}$	0	$m_{3,32}$	0
$m_{1,33}$	0	$m_{2,33}$	0	$m_{3,33}$	0
$m_{1,34}$	0	$m_{2,34}$	0	$m_{3,34}$	0
$m_{1,35}$	0.0000	$m_{2,35}$	0.0000	$m_{3,35}$	0.0000

Table 3.13: Elements of the $\mathrm{M}(T_s)$ matrix in the pendulum and tilt-sensor case for $T_s = 0.01$

Element	Value	Element	Value	Element	Value
$m_{1,1}$	0.0002	$m_{2,1}$	0.0462	$m_{3,1}$	0.0000
$m_{1,2}$	0	$m_{2,2}$	0	$m_{3,2}$	0
$m_{1,3}$	0	$m_{2,3}$	0	$m_{3,3}$	0
$m_{1,4}$	0	$m_{2,4}$	0	$m_{3,4}$	0
$m_{1,5}$	0	$m_{2,5}$	0	$m_{3,5}$	0
$m_{1,6}$	0	$m_{2,6}$	0	$m_{3,6}$	0
$m_{1,7}$	0	$m_{2,7}$	0	$m_{3,7}$	0
$m_{1,8}$	0.0000	$m_{2,8}$	0.0005	$m_{3,8}$	0.0000
$m_{1,9}$	0	$m_{2,9}$	0	$m_{3,9}$	0
$m_{1,10}$	0	$m_{2,10}$	0	$m_{3,10}$	0
$m_{1,11}$	0.0000	$m_{2,11}$	0.0000	$m_{3,11}$	0.0000
$m_{1,12}$	0	$m_{2,12}$	0	$m_{3,12}$	0
$m_{1,13}$	0	$m_{2,13}$	0	$m_{3,13}$	0
$m_{1,14}$	0	$m_{2,14}$	0	$m_{3,14}$	0
$m_{1,15}$	0.0000	$m_{2,15}$	0.0000	$m_{3,15}$	0.0000
$m_{1,16}$	0.0000	$m_{2,16}$	0.0000	$m_{3,16}$	0.0000
$m_{1,17}$	0	$m_{2,17}$	0	$m_{3,17}$	0
$m_{1,18}$	0	$m_{2,18}$	0	$m_{3,18}$	0
$m_{1,19}$	0	$m_{2,19}$	0	$m_{3,19}$	0
$m_{1,20}$	0.0000	$m_{2,20}$	0.0000	$m_{3,20}$	0.0000
$m_{1,21}$	0.0000	$m_{2,21}$	0.0000	$m_{3,21}$	0.0000
$m_{1,22}$	0.0000	$m_{2,22}$	0.0000	$m_{3,22}$	0.0000
$m_{1,23}$	0.0000	$m_{2,23}$	0.0000	$m_{3,23}$	0.0000
$m_{1,24}$	0	$m_{2,24}$	0	$m_{3,24}$	0
$m_{1,25}$	0	$m_{2,25}$	0	$m_{3,25}$	0
$m_{1,26}$	0	$m_{2,26}$	0	$m_{3,26}$	0
$m_{1,27}$	0	$m_{2,27}$	0	$m_{3,27}$	0
$m_{1,28}$	0.0000	$m_{2,28}$	0.0000	$m_{3,28}$	0.0000
$m_{1,29}$	0.0000	$m_{2,29}$	0.0000	$m_{3,29}$	0.0000
$m_{1,30}$	0	$m_{2,30}$	0	$m_{3,30}$	0
$m_{1,31}$	0	$m_{2,31}$	0	$m_{3,31}$	0
$m_{1,32}$	0	$m_{2,32}$	0	$m_{3,32}$	0
$m_{1,33}$	0	$m_{2,33}$	0	$m_{3,33}$	0
$m_{1,34}$	0	$m_{2,34}$	0	$m_{3,34}$	0
$m_{1,35}$	0.0000	$m_{2,35}$	0.0000	$m_{3,35}$	0.0000

of increased computational speed. Also note that in EKF2 and EKF3, the nonlinearity $sin\ \theta \approx \theta - (\theta^3/6)$ whereas in EKF1, the nonlinearity is not approximated.

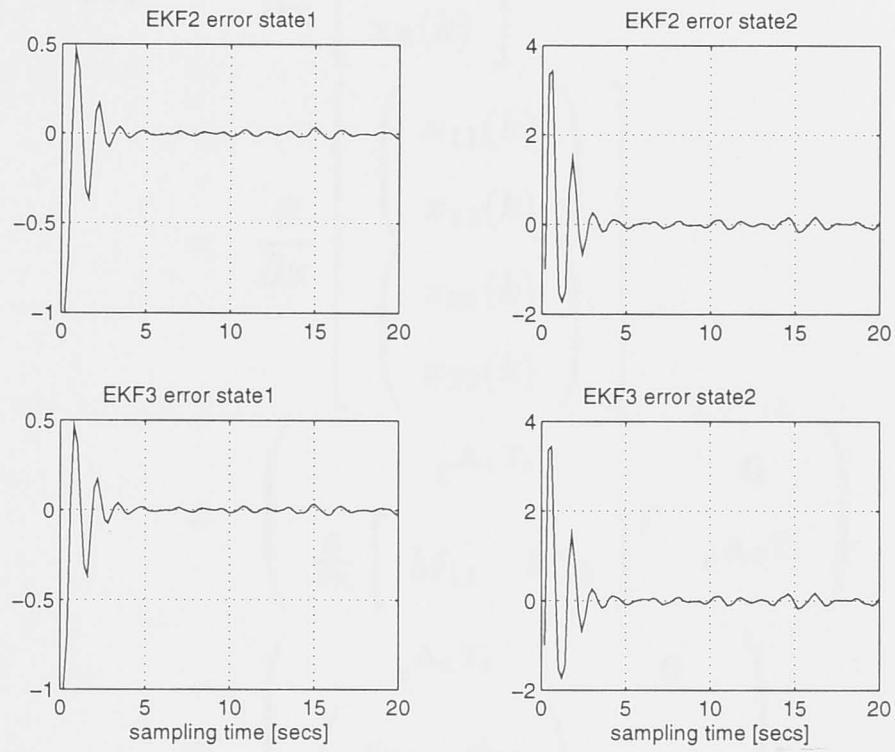


Figure 3.8: EKF2 and EKF3 state estimation errors with encoder measurements

Table 3.14: Estimation error variances in the EKF2 and EKF3 cases

Sensor	Est. error variance [EKF2]	Est. error variance [EKF3]
Encoder	0.0219, 0.3287	0.0229, 0.3296
Rate-gyro	0.0220, 0.3115	0.0232, 0.3200
Encoder & rate-gyro	0.0217, 0.3114	0.0220, 0.3120
Tilt-sensor	0.0222, 0.3268, 0.0000	0.0232, 0.3298, 0.0000
Rate-gyro & tilt-sensor	0.0220, 0.3114, 0.0000	0.0231, 0.3200, 0.0001

3.3.3 Computational issues

EKF2 gain in the pendulum case

To compute the Kalman gain, the Jacobian Ψ is to be computed. Define

$$\mathbf{M}\hat{\mathbf{x}}_1^{[3]} = \begin{pmatrix} M_{11} & M_{12} \end{pmatrix}^T$$

when all the elements of the $\mathbf{M}$ matrix are considered.

Thus

$$\begin{aligned}
\Psi_{k+1} &= \frac{\partial}{\partial \mathbf{x}} \begin{bmatrix} \mathbf{x_1}(k) \\ \mathbf{x_2}(k) \end{bmatrix} \\
&= \frac{\partial}{\partial \mathbf{x}} \begin{bmatrix} \begin{pmatrix} x_{11}(k) \\ x_{12}(k) \end{pmatrix} \\ \begin{pmatrix} x_{21}(k) \\ x_{22}(k) \end{pmatrix} \end{bmatrix} \\
&= \begin{pmatrix} e^{\mathbf{A_1}T_s} & \mathbf{0} \\ \frac{\partial}{\partial \mathbf{x}} \begin{bmatrix} M_{11} & M_{12} \end{bmatrix}^T & e^{\mathbf{A_2}T_s} \end{pmatrix} \\
&= \begin{pmatrix} e^{\mathbf{A_1}T_s} & \mathbf{0} \\ \begin{pmatrix} \psi_{31} & \psi_{32} \\ \psi_{41} & \psi_{42} \end{pmatrix} & e^{\mathbf{A_2}T_s} \end{pmatrix}
\end{aligned}$$

where

$$\begin{aligned}
\psi_{31} &= \frac{\partial}{\partial x_{11}} (M_{11}) \\
&= 3m_{1,1}\ x_{11}^2(k) + 2\sqrt{3}m_{1,2}\ x_{11}(k)x_{12}(k) + 2\sqrt{3}m_{1,3}\ x_{11}(k)u(k) \\
&+ \sqrt{3}m_{1,4}\ x_{12}^2(k) + \sqrt{6}m_{1,5}\ x_{12}(k)u(k) + \sqrt{3}m_{1,6}\ u^2(k) \\
\psi_{32} &= \frac{\partial}{\partial x_{12}} (M_{11}) \\
&= \sqrt{3}m_{1,2}\ x_{11}^2(k) + 2\sqrt{3}m_{1,4}\ x_{11}(k)x_{12}(k) + \sqrt{6}m_{1,5}\ x_{11}(k)u(k) \\
&+ 3m_{1,7}\ x_{12}^2(k) + 2\sqrt{3}m_{1,8}\ x_{12}(k)u(k) + \sqrt{3}m_{1,9}\ u^2(k) \\
\psi_{41} &= \frac{\partial}{\partial x_{11}} (M_{12}) \\
&= 3m_{2,1}\ x_{11}^2(k) + 2\sqrt{3}m_{2,2}\ x_{11}(k)x_{12}(k) + 2\sqrt{3}m_{2,3}\ x_{11}(k)u(k) \\
&+ \sqrt{3}m_{2,4}\ x_{12}^2(k) + \sqrt{6}m_{2,5}\ x_{12}(k)u(k) + \sqrt{3}m_{2,6}\ u^2(k) \\
\psi_{42} &= \frac{\partial}{\partial x_{12}} (M_{12}) \\
&= \sqrt{3}m_{2,2}\ x_{11}^2(k) + 2\sqrt{3}m_{2,4}\ x_{11}(k)x_{12}(k) + \sqrt{6}m_{2,5}\ x_{11}(k)u(k) \\
&+ 3m_{2,7}\ x_{12}^2(k) + 2\sqrt{3}m_{2,8}\ x_{12}(k)u(k) + \sqrt{3}m_{2,9}\ u^2(k)
\end{aligned}$$

where the corresponding matrices are defined in (3.31) and thus the Kalman gain can

be determined. To compute the Jacobian Ψ in the EKF3 case (i.e., with $\mathbf{M_a}$ matrix), (3.33) is utilised.

EKF2 gain in the pendulum and tilt-sensor case

Define

$$\mathbf{M}\hat{\mathbf{x}}_{\mathbf{1}}^{[3]}(k) \;=\; \begin{pmatrix} M_1 & M_2 & M_3 \end{pmatrix}^T$$

when all the elements of the $\mathbf{M}(T_s)$ matrix are considered in the combined pendulum and tilt-sensor model. Thus

$$\begin{aligned}
\Psi_{k+1} &= \frac{\partial}{\partial \mathbf{x}} \begin{bmatrix} \mathbf{x_1}(k) \\ \mathbf{x_2}(k) \end{bmatrix} \\
&= \frac{\partial}{\partial \mathbf{x}} \begin{bmatrix} \begin{pmatrix} x_{11}(k) \\ x_{12}(k) \\ x_{13}(k) \end{pmatrix} \\ \begin{pmatrix} x_{21}(k) \\ x_{22}(k) \\ x_{23}(k) \end{pmatrix} \end{bmatrix} \\
&= \begin{pmatrix} e^{\mathbf{A_1}T_s} & \mathbf{0} \\ \frac{\partial}{\partial \mathbf{x}} \begin{bmatrix} M_1 & M_2 & M_3 \end{bmatrix}^T & e^{\mathbf{A_2}T_s} \end{pmatrix} \\
&= \begin{pmatrix} e^{\mathbf{A_1}T_s} & \mathbf{0} \\ \begin{pmatrix} \psi_{41} & \psi_{42} & \psi_{43} \\ \psi_{51} & \psi_{52} & \psi_{53} \\ \psi_{61} & \psi_{62} & \psi_{63} \end{pmatrix} & e^{\mathbf{A_2}T_s} \end{pmatrix}
\end{aligned}$$

where

$$\begin{aligned}
\psi_{41} &= \frac{\partial}{\partial x_{11}}(M_1)\,;\; \psi_{42} = \frac{\partial}{\partial x_{12}}(M_1)\,;\; \psi_{43} = \frac{\partial}{\partial x_{13}}(M_1) \\
\psi_{51} &= \frac{\partial}{\partial x_{11}}(M_2)\,;\; \psi_{52} = \frac{\partial}{\partial x_{12}}(M_2)\,;\; \psi_{53} = \frac{\partial}{\partial x_{13}}(M_2) \\
\psi_{61} &= \frac{\partial}{\partial x_{11}}(M_3)\,;\; \psi_{62} = \frac{\partial}{\partial x_{12}}(M_3)\,;\; \psi_{63} = \frac{\partial}{\partial x_{13}}(M_3)
\end{aligned}$$

where the corresponding matrices are defined in (3.36) and thus the Kalman gain can be determined. To compute the Jacobian Ψ in the EKF3 case (i.e., with $\mathbf{M_a}$ matrix), (3.38) is utilised.

State estimate extrapolation in the EKF2 case

The state estimate extrapolation in the discrete extended Kalman filtering case is done as follows:

$$\begin{aligned} \text{Define } \mathbf{s}(k) &\triangleq \begin{pmatrix} \mathbf{x_1}(k) \\ \mathbf{x_2}(k) \end{pmatrix} \\ \mathbf{s}(k+1) &= \mathbf{f_1}\left[\mathbf{s}(k), u(k)\right] + \omega(k) \\ &= \begin{pmatrix} e^{\mathbf{A}_1 T_s}\mathbf{x_1}(k) + \mathbf{u}(k)\mathbf{A_1}^{-1}\left[e^{\mathbf{A}_1 T_s} - \mathbf{I}\right]\mathbf{B_1} + \omega(k) \\ e^{\mathbf{A}_2 T_s}\mathbf{x_2}(k) + \mathbf{M}\hat{\mathbf{x}}_1^{[3]}(k) \end{pmatrix} \end{aligned} \tag{3.39}$$

(3.39) is used for the state estimate extrapolations for both the pendulum and the tilt-sensor cases. Note that the matrices $\mathbf{A_1}$, $\mathbf{A_2}$, $\mathbf{B_1}$ and $\mathbf{B_2}$ pertaining to each of these cases are different and are defined in (3.31) and (3.36) respectively. Also note that $\mathbf{M}\hat{\mathbf{x}}_1^{[3]}$ is different for both the cases and $\omega(\cdot)$ indicates the process noise that is added to the linear part.

Performance of the extended Kalman filters

In estimating the time required by matrix computations, it is traditional to estimate the number of multiplications. The computational complexities of numerical algorithms can also be expressed in terms of *flops*, which are roughly the numbers of multiply-and-accumulate operations required for execution. It gives a rough idea of the relative complexities of alternative algorithms. The complexities will be functions of the problem size, which can be represented by the dimensions of the matrices involved.

The number of multiplications are used as a measure of the computational speed for the comparison of two filters. The following rules were observed in calculating the number of multiplications:

i. The number of multiplications denoted by $n(muls)$, in multiplying two square matrices of order $n \times n$ is n^3.

ii. In multiplying two matrices of order $n \times n$ and $n \times 1$, $n(muls) = n^2$.

ii. In multiplying a constant and a matrix of order $n \times n$, $n(muls) = n^2$.

ii. In multiplying a constant and a matrix of order $n \times 1$, $n(muls) = n$.

ii. $n(muls) = 1$, when multiplying two constants.

Following the rules outlined above, the number of multiplications in the EKF1, EKF2 and EKF3 approaches are calculated and tabulated in Tables 3.15 and 3.16. For the state estimate extrapolation, (3.39) is utilised and from Table 3.7, the error covariance extrapolation can be computed with the Jacobian Ψ being computed as outlined in section 3.3.3. In the EKF3 case, (3.33) is utilised and thus the computational burden is brought down by a significant factor. For the EKF1 approach, the error covariance propagation is computed as indicated in section 3.2.2. From Tables 3.15 and 3.16, it can be seen that the EKF2 is faster than the EKF1 by a factor of 4.7 and EKF3 is faster than the EKF1 by a factor of ≈ 6.

Table 3.15: Comparison of computational speed in the EKF1 and EKF2 cases

Type of nonlinear filter	Required no. of muls.
EKF1	906
EKF2	192
RATIO	4.7188

Table 3.16: Comparison of computational speed in the EKF1 and EKF3 cases

Type of nonlinear filter	Required no. of muls.
EKF1	906
EKF3	152
RATIO	5.9605

Linear versus nonlinear estimation

Comparing the estimation error variances that were tabulated in Tables 3.4, 3.6 and 3.14, it can be inferred that the extended Kalman filters perform far better than the LKF. This is very much so at higher swing-angles of the pendulum for reasons that were discussed in section 2.2 of chapter 2. This can also be seen from Figures 3.4, 3.5 and 3.8. EKF2 performs as good as EKF1 as demonstrated by the estimation error variances. EKF2 and EKF3 are to be preferred to EKF1 because of its numerical efficiency as shown in Tables 3.15 and 3.16.

3.4 Summary

This chapter presented the state estimation problem. Using the tensor techniques, relevant results were developed for the state estimation of a class of nonlinear systems that fall under the extended generalised Wiener model structure which showed that the tensor techniques and the discrete extended Kalman filtering can serve as powerful tools in the state estimation for such classes of nonlinear systems. Four Kalman filters, LKF, EKF1, EKF2 and EKF3 were employed for the state estimation of the pendulum. Using the Volterra series, the pendulum was modelled to have the extended generalised Wiener model structure as shown in chapter 2 and thus it was demonstrated how these results could be utilised for the state estimation of the pendulum model and extended to the combined pendulum and tilt-sensor model. The corresponding estimation error variances were calculated which showed that, when the nonlinearities are severe, the extended Kalman filtering is to be desired to the linear Kalman filtering. Also in the nonlinear case, EKF2 and EKF3 are to be preferred to EKF1 because of the distinct advantage of increased computational speed. It was shown that EKF2 is 4.7 times faster than EKF1 and EKF3 is ≈ 6 times faster than EKF1. In the next chapter, the estimates that were obtained using LKF, EKF2 and EKF3 will be employed in the tracking control of the pendulum model.

Chapter 4

Tracking Control

In this chapter, the tracking control of the pendulum is considered. The estimates that were obtained using the estimation strategies discussed in chapter 3 are employed for this purpose. Both linear and nonlinear estimates are employed and the corresponding results are presented. The linear estimates are provided by the LKF approach and the nonlinear estimates by the EKF2 and EKF3 approaches. The feedback linearisation technique and the internal model principle are employed for achieving tracking of the reference signals. Also steady-state tracking and estimation error variances for both the linear and nonlinear cases are provided along with the corresponding simulation results.

4.1 Preliminaries

4.1.1 The internal model principle

The problem of controlling a fixed plant in order to have its output track a reference signal is one of the most important problems in control theory [IB90]. The work of [FW76] has shown that, in the case of error feedback, any regulator which solves the problem in question, incorporates a model of the dynamical system generating the reference signal which must be tracked. This property is commonly known as the *internal*

model principle. Several authors, [HW84], [AD87], [Ben96] and [JR88], have considered the corresponding problem in a nonlinear setting. These ideas are used in the tracking of the nonlinear pendulum model in which, the plant output is made to track a time-varying (periodic) reference signal in the presence of sensor and actuator noise sources.

Consider a continuous-time plant governed by

$$\begin{aligned} \dot{\mathbf{x}}_\mathbf{p}(t) &= \mathbf{A_p x_p}(t) + \mathbf{b_p} u_p(t) + \omega_p(t) && (4.1) \\ y(t) &= \mathbf{C_p x_p}(t) && (4.2) \end{aligned}$$

where $\mathbf{x_p}(t) \in \mathcal{R}^{n_p}$ is the state vector, $u_p(t) \in \mathcal{R}^{m_p}$ denotes the input, $y(t) \in \mathcal{R}^{\ell_p}$ is the output which is required to track a reference trajectory r and $\omega_p(t)$ represents the process noise.

The internal model principle [FW76] is used for achieving tracking which possibly involves the design of a pre-compensator. If the plant (4.1) does not contain all the modes of the reference signal, a pre-compensator is added which is represented by

$$\begin{aligned} \dot{\mathbf{x}}_\mathbf{a}(t) &= \mathbf{A_a x_a}(t) + \mathbf{b_a} u(t) && (4.3) \\ u_p(t) &= \mathbf{C_a x_a}(t) \end{aligned}$$

where $\mathbf{x_a}(t) \in \mathcal{R}^{n_a}$ is the state vector, $u_p(t) \in \mathcal{R}^{m_a}$. The combined plant and pre-compensator can be represented as

$$\begin{aligned} \begin{pmatrix} \dot{\mathbf{x}}_\mathbf{p}(t) \\ \dot{\mathbf{x}}_\mathbf{a}(t) \end{pmatrix} &= \begin{pmatrix} \mathbf{A_p} & \mathbf{b_p C_a} \\ \mathbf{0} & \mathbf{A_a} \end{pmatrix} \begin{pmatrix} \mathbf{x_p}(t) \\ \mathbf{x_a}(t) \end{pmatrix} + \begin{pmatrix} \mathbf{0} \\ \mathbf{b_a} \end{pmatrix} u(t) + \begin{pmatrix} \omega_p(t) \\ \mathbf{0} \end{pmatrix} && (4.4) \\ y(t) &= \begin{pmatrix} \mathbf{C_p} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{x_p}(t) \\ \mathbf{x_a}(t) \end{pmatrix} \end{aligned}$$

This can be written more succinctly as

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A x}(t) + \mathbf{b} u(t) + \omega(t) && (4.5) \\ y(t) &= \mathbf{C x}(t) \end{aligned}$$

where the definitions of $\mathbf{A}$, $\mathbf{b}$, $\mathbf{C}$ and $\omega(t)$ follow from (4.4).

Control objective

The desired objective is to make the output of the pendulum model follow a pre-specified reference trajectory in the presence of sensor (measurement) additive white noises in the system. For this simulation study, a ramp reference trajectory is used. This trajectory is chosen as it generates a profile through which a robot link moves and also as this kind of reference trajectory closely emulates the kind of trajectory that is used in industrial robots, for example, a welding robot in a car assembly line.

Reference model

The reference signal r can be modelled as the response of

$$\dot{\mathbf{x}}_\mathbf{r}(t) = \mathbf{A}\mathbf{x}_\mathbf{r}(t);\ \ \mathbf{x}_\mathbf{r}(0) \in \mathcal{R}^n \tag{4.6}$$

$$r(t) = \mathbf{C}\mathbf{x}_\mathbf{r}(t) \tag{4.7}$$

The solution of (4.6) is

$$\mathbf{x}_\mathbf{r}(t) = e^{\mathbf{A}t}\mathbf{x}_\mathbf{r}(0) \tag{4.8}$$

$$r(t) = \mathbf{C}e^{\mathbf{A}t}\mathbf{x}_\mathbf{r}(0)$$

subject to which the initial conditions are calculated.

4.1.2 Selection of control gains

When an error occurs between reference and feedback, the control law calculates the energy that has to be applied to the system to eliminate the error. The robot joint response to a commanded position change is generally required to be critically damped and fast. In a linear setting, this can be easily attained by selecting the controller gains in a suitable way and sufficiently large. In addition to this, it is normally desirable to have a high disturbance rejection ratio or high stiffness which also requires a large proportional gain.

In theory, these design constraints can always be satisfied since one is allowed to select the controller gains arbitrarily large. The actual choice of these gains, is, however limited by practical considerations [Ber94]. For instance, sensor noise generally upper bounds the allowable derivative gain [KK88]. Furthermore, the presence of un-modelled high-frequency dynamics, such as link flexibility, restricts the control system bandwidth and, consequently, the feedback gains. Therefore, in practice, several compromises have to be made in the selection and tuning of the controller gains. Another important aspect to deal with is the possibly highly nonlinear nature of the error dynamics of controlled robotic systems.

Pole-placement technique

For a closed-loop continuous-time linear system with system matrices $\mathbf{A_c}$ and $\mathbf{b_c}$ where $\mathbf{A_c}$ is $n \times n$ and a gain matrix $\mathbf{K}$, the resulting closed-loop characteristic equation is given by

$$det\left[s\mathbf{I} - (\mathbf{A_c} - \mathbf{b_c}\mathbf{K})\right] \;=\; 0 \tag{4.9}$$

where *det* refers to the determinant. The control law design consists of picking the gains $\mathbf{K}$ so that the roots of (4.9) are in desirable locations. Let the desired locations be $s = s_1, s_2, s_3, \cdots$ and then the corresponding desired control characteristic equation is

$$\alpha_c(s) \;=\; (s - s_1)\,(s - s_2)\;\cdots\;(s - s_n) = 0 \tag{4.10}$$

Hence the required elements of $\mathbf{K}$ are obtained by matching coefficients in (4.9) and (4.10), thus forcing the system-characteristic equation to be identical with the desired formula [F$^+$91].

4.1.3 Feedback linearisation

The first step in designing a control system for a given physical plant is to derive a meaningful model of the plant. i.e., a model that captures the key dynamics of the plant in the operational range of interest. Models of physical systems come in various

forms, depending on the modelling approach and assumptions. These issues were addressed in chapter 2. Some forms, however, lend themselves more easily to controller design. Feedback linearisation deals with techniques for transforming original system models onto equivalent models of a simpler form [SL91].

Feedback linearisation is a method of designing control for a class of feedback linearisable nonlinear systems [Kha92, SL91]. It can be used as a nonlinear methodology. The basic idea is to first transform the nonlinear system (either completely or partially) into a linear system, and then use the well-known and powerful linear design techniques to complete the control design for the class of nonlinear systems. This approach has been used to solve a number of practical nonlinear control problems. The main advantage of feedback linearisation over point-wise linearisation is that once such a control is found, linearisation is achieved independently of the operating point. The conditions under which a nonlinear system is feedback linearisable are complicated, that is, the nonlinear feedback control and the nonlinear coordinate transformation may be very difficult to find. Fortunately, feedback linearisation can often be easily achieved for robotic systems [QD96].

Controller-estimator design via feedback linearisation

Consider the pendulum model equation

$$J\ddot{\theta}(t) + K_1\dot{\theta}(t) + K_0\ sin\ \theta(t) \quad = \quad u(t) + \omega(t)$$

where $u(t)$ refers to the control torque applied to the pendulum. Let

$$u(t) \quad \equiv \quad K_0\ sin\ \hat{\theta}(t) + V(t) \tag{4.11}$$

where $\hat{\theta}(t)$ refers to the estimate obtained by either the LKF or the EKF2 or the EKF3 approach and $V(t)$ to the control law to be proposed. Substituting (4.11) into the pendulum model equation,

$$\begin{aligned} J\ddot{\theta}(t) + K_1\dot{\theta}(t) + K_0\ sin\ \theta(t) &= K_0\ sin\ \hat{\theta}(t) + V(t) + \omega(t) \\ \text{where } V(t) &= -k_1\left(\hat{\theta}(t) - \theta_r(t)\right) - k_2\left(\hat{\dot{\theta}}(t) - \dot{\theta}_r(t)\right). \end{aligned}$$

$\theta_r(t)$ and $\dot{\theta}_r(t)$ are generated by the reference model based on the internal model principle which generates the necessary reference signals and k_1 and k_2 are the controller gains to be determined.

Assuming that $\hat{\theta}(t) = \theta(t)$, the pendulum equation becomes

$$J\ddot{\theta}(t) + K_1\dot{\theta}(t) \quad = \quad V(t) + \omega(t)$$

The structure of the pendulum model after feedback linearisation is shown in Figure 4.1. Only one integrator is required since the plant itself possesses an integrator.

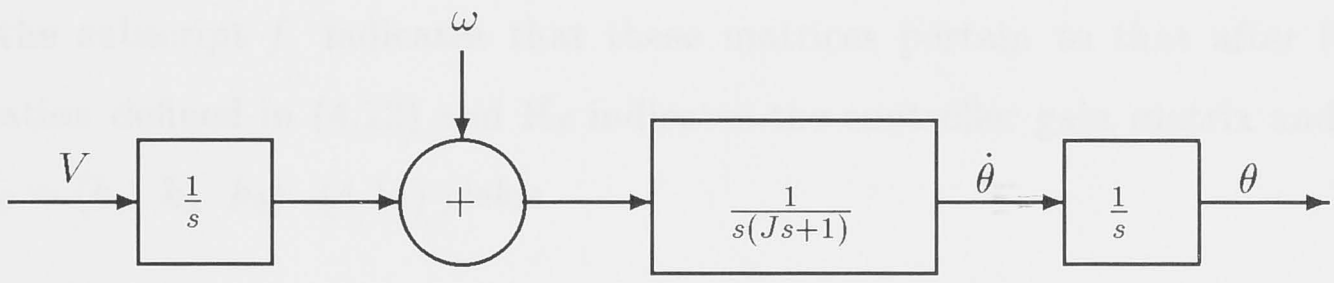


Figure 4.1: Pendulum model structure after feedback linearisation

The assumption $\hat{\theta}(t) = \theta(t)$ is not always satisfied since there is an estimation error. But it holds good as the estimation error is not very large. The state-space representation after feedback linearisation is as follows

$$\dot{\mathbf{x}}(t) \quad = \quad \mathbf{A_f}\ \mathbf{x}(t) + \mathbf{b_f}\ V(t) + \mathbf{d_f}\ \omega(t) \tag{4.12}$$

$$\mathbf{A_f} \quad = \quad \begin{pmatrix} 0 & 1 & 0 \\ 0 & \frac{-K_1}{J} & \frac{1}{J} \\ 0 & 0 & 0 \end{pmatrix};\ \mathbf{b_f} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix};\ \mathbf{d_f} = \begin{pmatrix} 0 \\ \frac{1}{J} \\ 0 \end{pmatrix};$$

A model for the constant ramp function is

$$y_r(t) = \alpha_2 t$$

The initial conditions of the ramp reference signal, $\mathbf{x_r}(0) \in \mathcal{R}^3$ subject to (4.8) are

$$\mathbf{x_r}(0) = \begin{pmatrix} 0 \\ -\alpha_2 \\ -\alpha_2 K_1 \end{pmatrix} \tag{4.13}$$

Control gains

The characteristic control equation after feedback linearisation is

$$det\left[s\mathbf{I} - (\mathbf{A_f} - \mathbf{b_f}\mathbf{K_f})\right] = 0 \tag{4.14}$$

where the subscript f indicates that these matrices pertain to that after feedback linearisation defined in (4.12) and $\mathbf{K_f}$ indicates the controller gain matrix and is such that $\mathbf{K_f} = (k_1 \quad k_2 \quad k_3)$. (4.14) yields

$$s^3 + \left(\frac{K_1}{J}\right)s^2 + \left(\frac{k_2}{J}\right)s + \left(\frac{k_1}{J}\right) = 0;\ k_3 = 0; \tag{4.15}$$

from which the the gains were calculated using the pole-placement technique as

$$k_1 = 10;\ k_2 = 2.5; \tag{4.16}$$

4.2 Results for the pendulum model

The structure of the discrete-estimator based controller structure is shown in Figure 4.2. $\omega(t)$ and $v(t_k)$ refer to the process noise and the sampled measurement noise respectively, as discussed in chapter 3. The gains were selected using pole-placement technique as indicated.

The encoder is the sensor that is used for the measurement of the output (θ) of the pendulum system which is to be tracked. LKF, EKF2 and EKF3 are employed for providing the estimates for the tracking control. The corresponding error variances are tabulated which provides a basis for analysing the errors associated with both the tracking and the estimation. The estimation error is also a means by which the

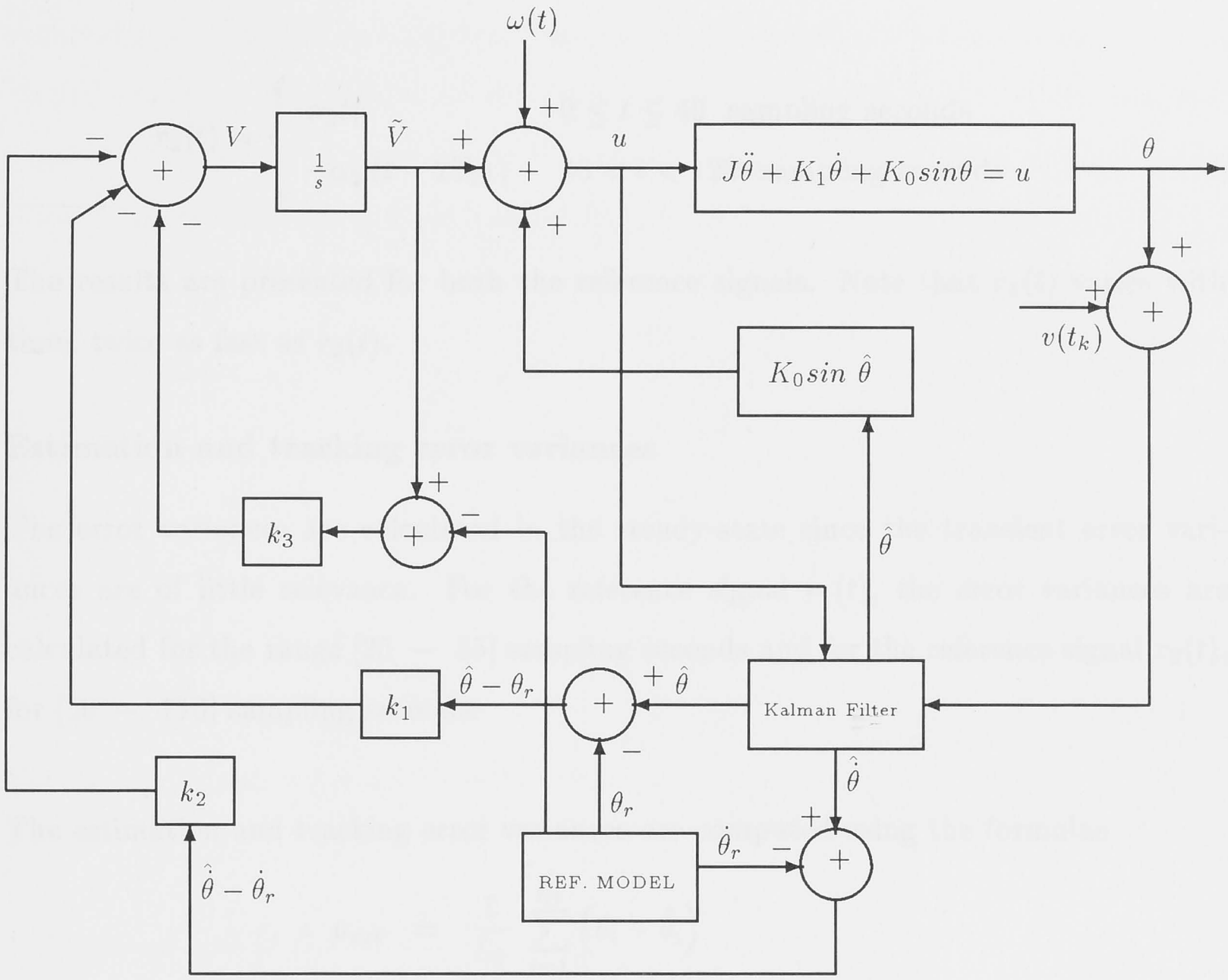


Figure 4.2: Controller-estimator based tracking control structure

effectiveness of the proposed feedback linearisation can be studied. The estimation error is the error that has occured due to the usage of $\hat{\theta}(t)$ instead of $\theta(t)$.

Reference signals

Two reference ramp signals, $r_1(t)$ and $r_2(t)$ are employed for the tracking control. Note that both of these reference trajectories are periodic. i.e., $r_i(t + T_i) = r_i(t)$, $i = 1, 2$ where T_i is the period of the corresponding trajectory.

$$r_1(t) = \begin{cases} \alpha_2 t; & 0 \leq t \leq 20 \ \text{ sampling seconds} \\ -\alpha_2 \left(t - 2T_1\right); & 20 < t < 60 \ \text{ sampling seconds} \end{cases}$$

$$r_2(t) = \begin{cases} \alpha_2 t; & 0 \leq t \leq 40 \text{ sampling seconds} \\ -\alpha_2 \left(t - 2T_2\right); & 40 < t < 120 \text{ sampling seconds} \end{cases}$$

The results are presented for both the reference signals. Note that $r_1(t)$ varies with time, twice as fast as $r_2(t)$.

Estimation and tracking error variances

The error variances are calculated in the steady-state since the transient error variances are of little relevance. For the reference signal $r_1(t)$, the error variances are calculated for the range [25 – 55] sampling seconds and for the reference signal $r_2(t)$, for [50 – 110] sampling seconds.

The estimation and tracking error variances are computed using the formulae

$$\begin{aligned} \mu_{est} &= \frac{1}{L_1} \sum_{i=1}^{L_1} \left(\theta_i - \hat{\theta}_i\right) \\ \mu_{tr} &= \frac{1}{L_1} \sum_{i=1}^{L_1} \left(\theta_i - \theta_{ri}\right) \\ \sigma^2_{est} &= \frac{1}{L_1 - 1} \sum_{i=1}^{L_1} \left(\theta_i^2 - \frac{L_1}{L_1 - 1} \mu^2_{est}\right) \qquad (4.17) \\ \sigma^2_{tr} &= \frac{1}{L_1 - 1} \sum_{i=1}^{L_1} \left(\theta_i^2 - \frac{L_1}{L_1 - 1} \mu^2_{tr}\right) \qquad (4.18) \end{aligned}$$

where L_1 is the number of sampling points considered, μ refers to the sample mean, σ^2 refers to the sample variance. The subscripts *est* and *tr* refer to estimation and tracking respectively.

Tracking control of the pendulum model using LKF

The LKF is employed for the estimation of the states of the pendulum and these estimated states are used for the tracking control. Note that the plant model is still nonlinear, but only the estimates are obtained by the LKF. This will be of help in

comparing the efficiency of the tracking control with nonlinear estimates to that with linear estimates. The results for the tracking control with LKF and for the reference signals, $r_1(t)$ and $r_2(t)$, are shown in Figures 4.3 to 4.8 and the tracking and estimation error variances are tabulated in Table 4.1.

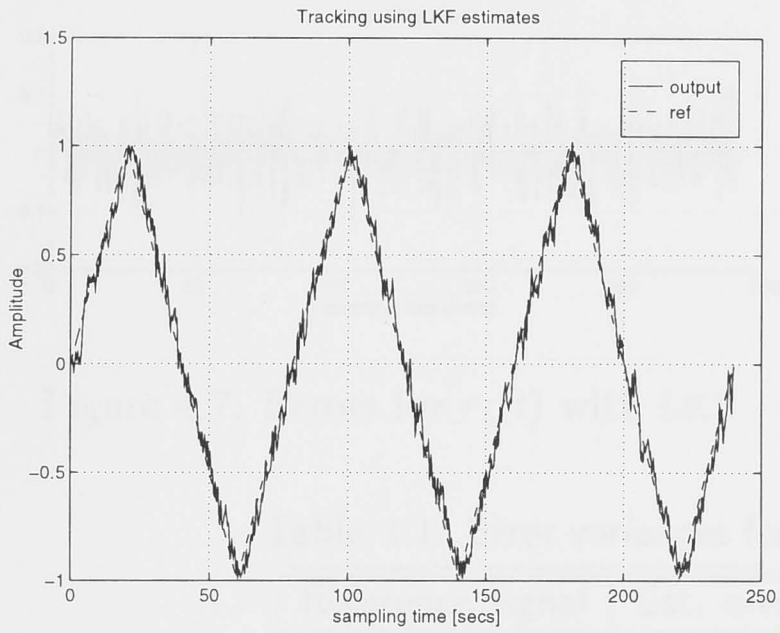


Figure 4.3: Tracking control for $r_1(t)$ with LKF

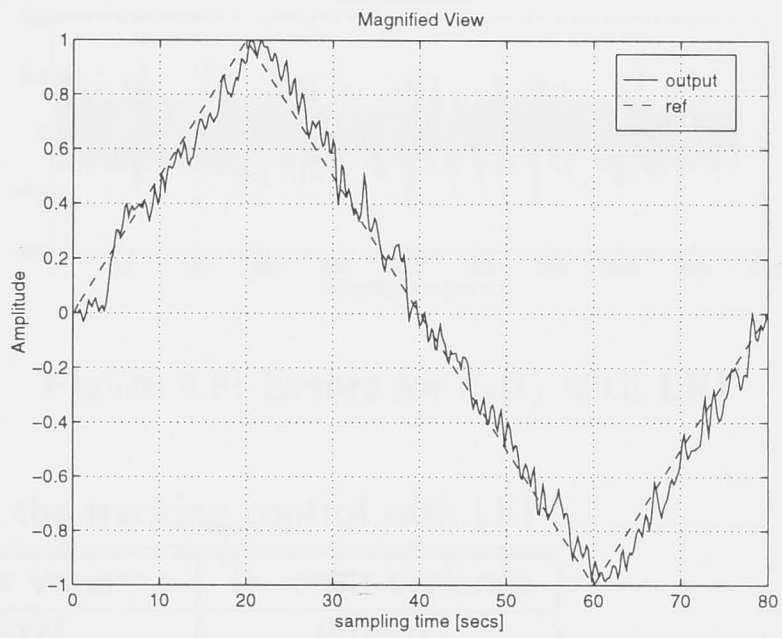


Figure 4.4: Magnified view

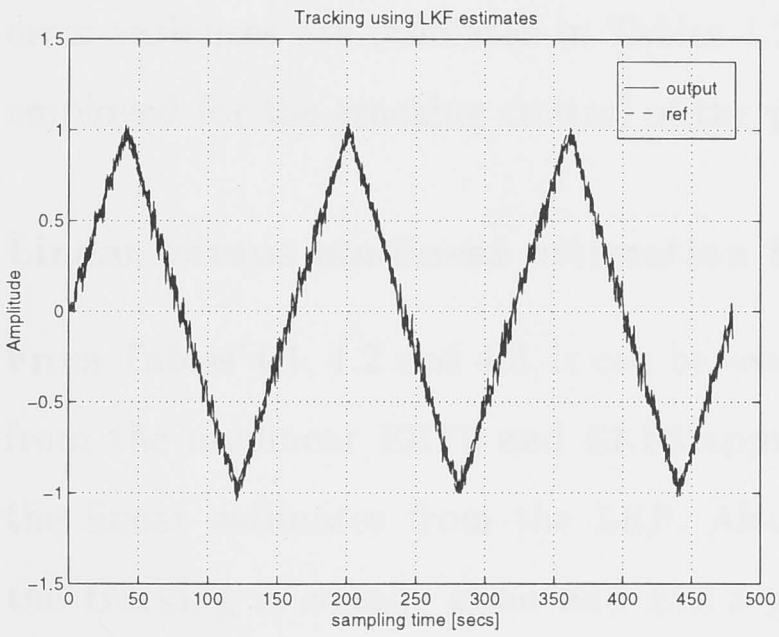


Figure 4.5: Tracking control for $r_2(t)$ with LKF

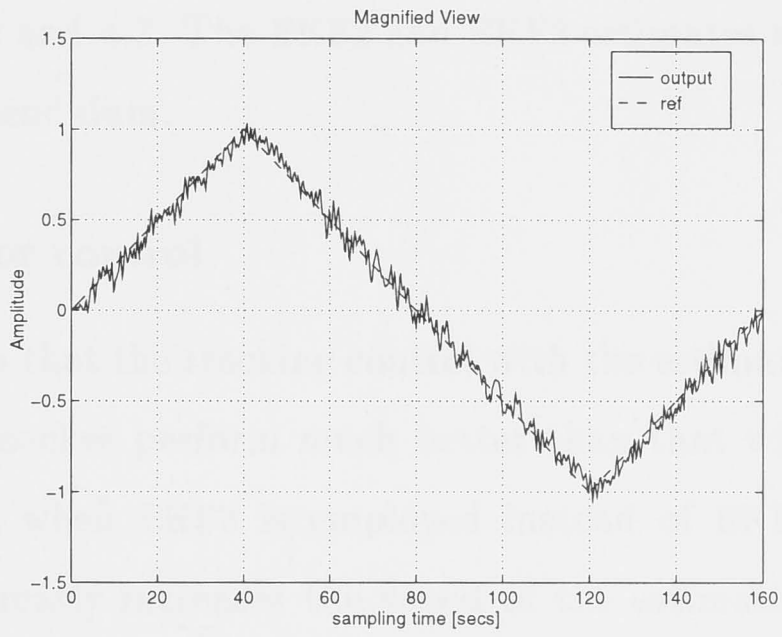


Figure 4.6: Magnified view

Tracking control of the pendulum model using EKF2 and EKF3

The results for the tracking control for the reference signals, $r_1(t)$ and $r_2(t)$, with the nonlinear estimates are shown in Figures 4.9 to 4.20 and the estimation and tracking

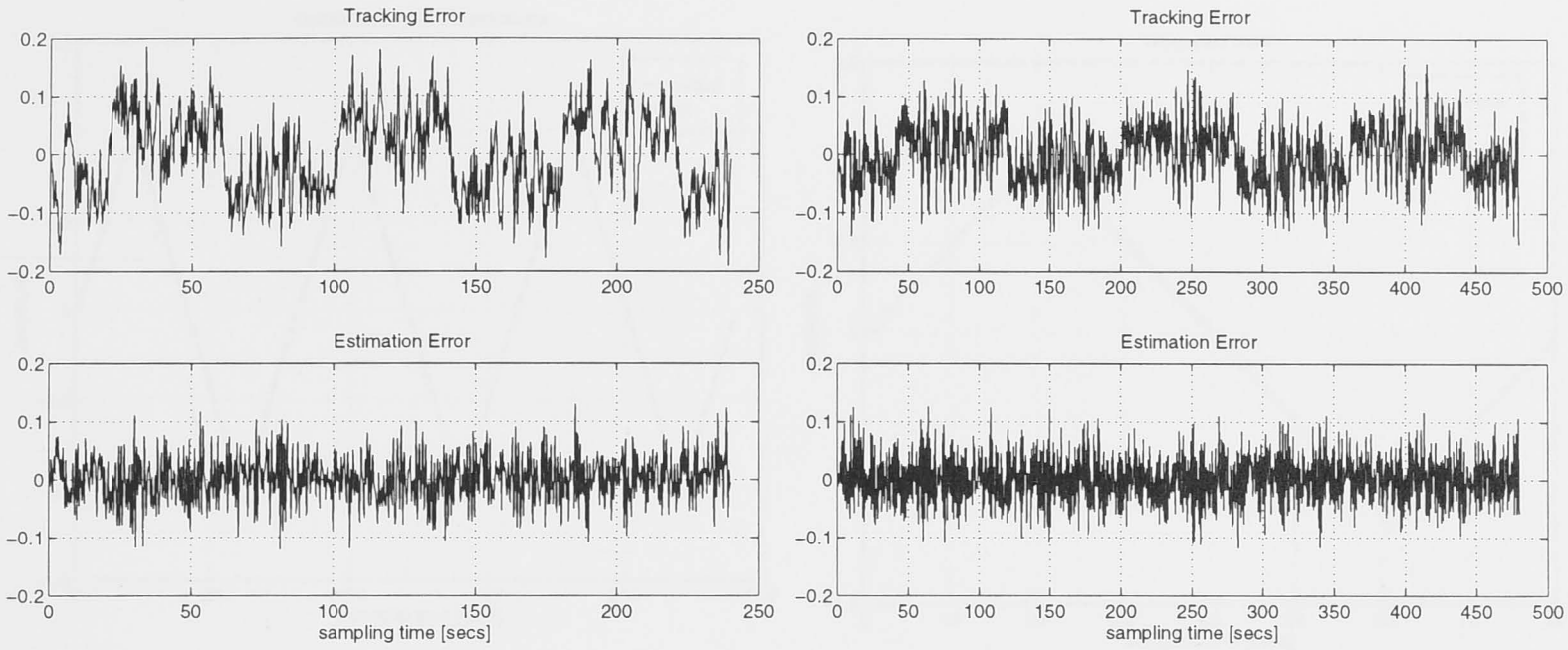


Figure 4.7: Errors for $r_1(t)$ with LKF Figure 4.8: Errors for $r_2(t)$ with LKF

Table 4.1: Error variances for the tracking control with LKF

Reference signal	Est. error variance	Tr. error variance
r_1	0.0021	0.0027
r_2	0.0016	0.0026

error variances are tabulated in Tables 4.2 and 4.3. The EKF2 and EKF3 estimates are employed for the tracking control of the pendulum.

Linear versus nonlinear estimation for control

From Tables 4.1, 4.2 and 4.3, it can be seen that the tracking control with the estimates from the nonlinear EKF2 and EKF3 approaches perform much better than that with the linear estimates from the LKF. Also when EKF3 is employed instead of EKF2, the tracking is equally good and EKF3 greatly increases the speed of the estimation algorithm. The estimation error variance is very small which demonstrates the efficiency of the feedback linearisation approach. Thus the tracking control with the nonlinear estimates is to be preferred to the tracking control with linear estimates.

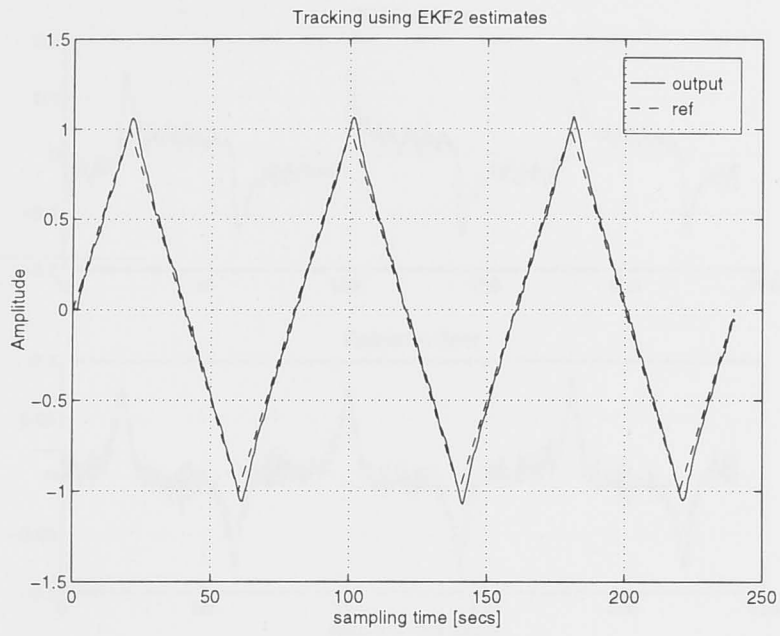


Figure 4.9: Tracking control for $r_1(t)$ with EKF2

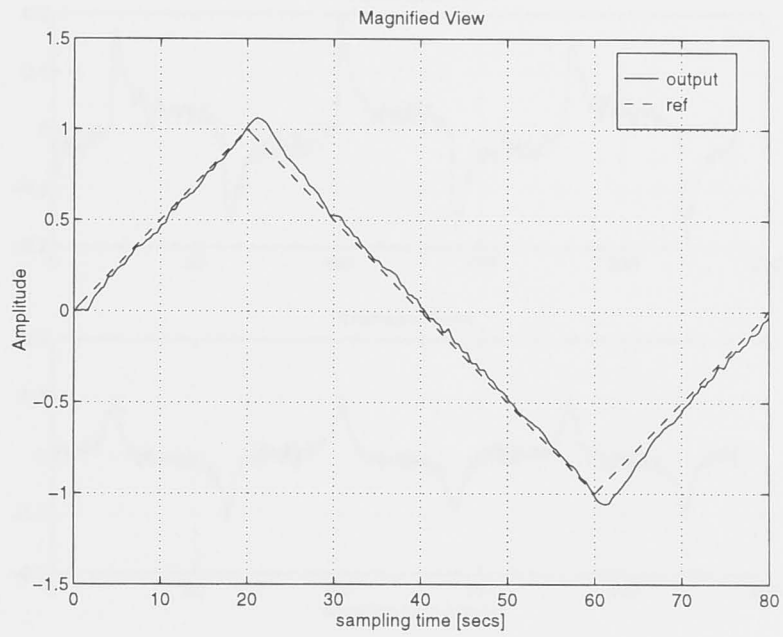


Figure 4.10: Magnified view

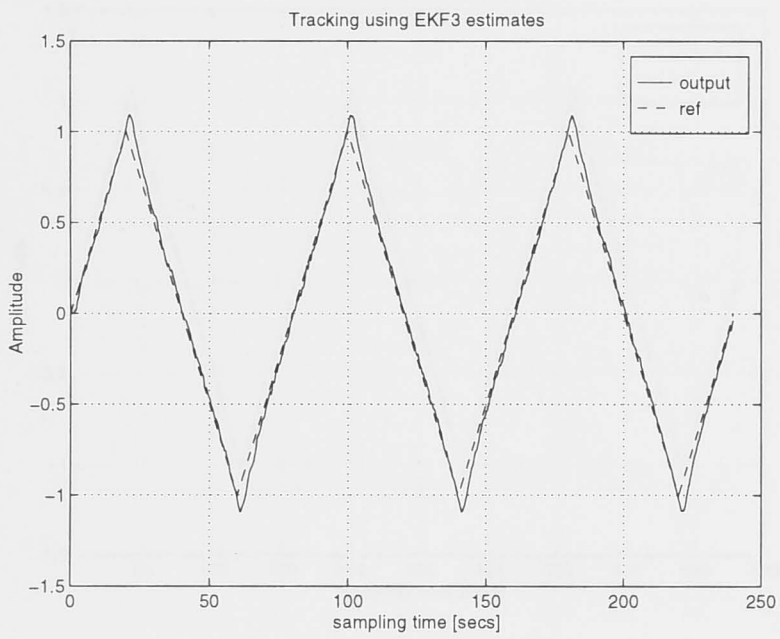


Figure 4.11: Tracking control for $r_1(t)$ with EKF3

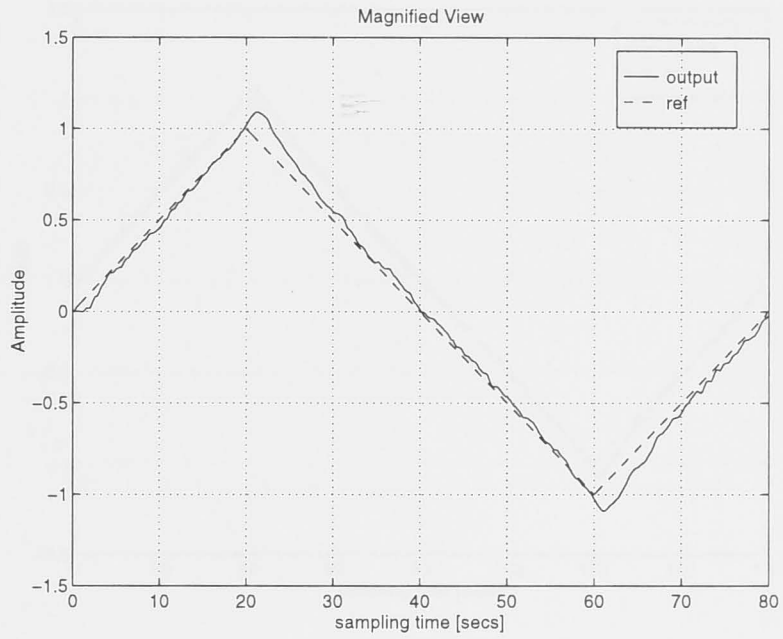


Figure 4.12: Magnified view

Table 4.2: Error variances for the tracking control with EKF2

Reference signal	Est. error variance	Tr. error variance
r_1	9.5997×10^{-5}	2.2774×10^{-4}
r_2	7.5406×10^{-5}	1.5585×10^{-4}

4.3 Summary

This chapter presented the tracking control of the pendulum model. The estimates obtained from the LKF, EKF2 and EKF3 approaches were employed for this purpose.

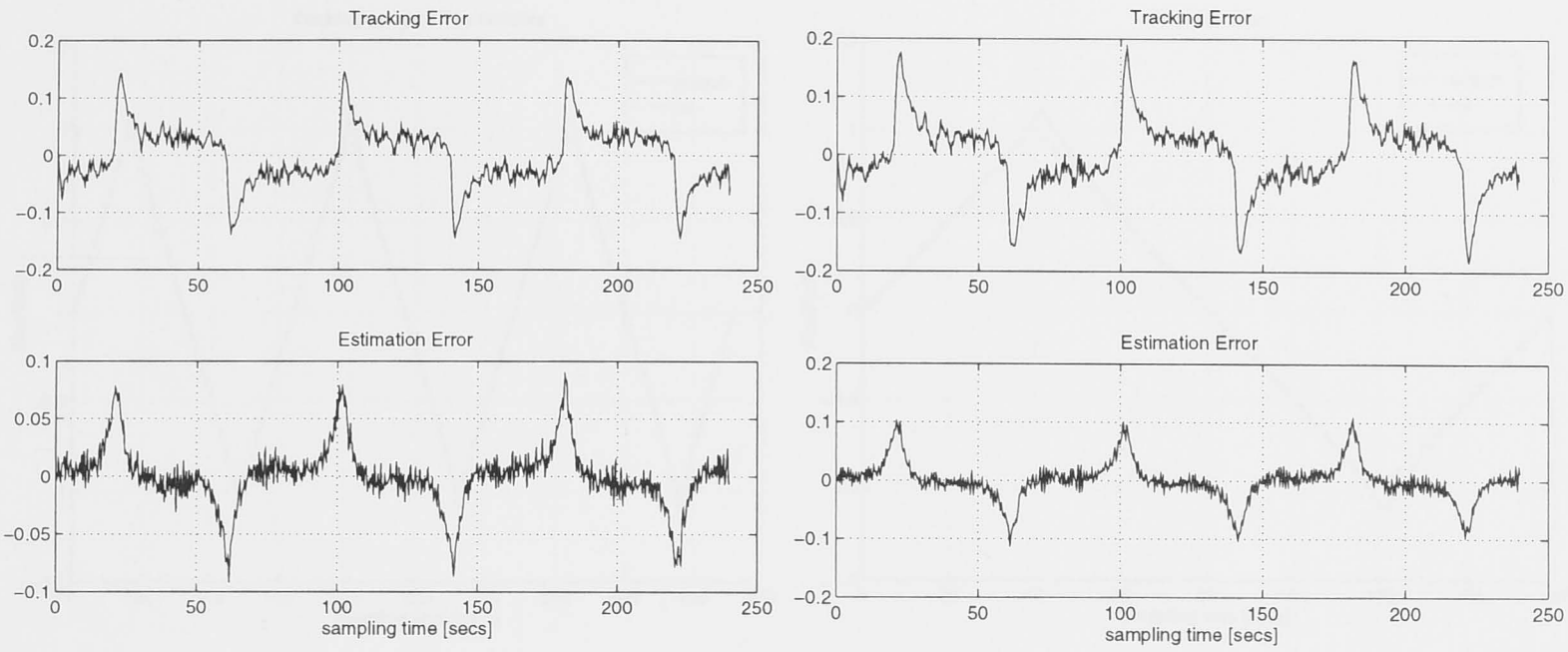


Figure 4.13: Errors for $r_1(t)$ with EKF2

Figure 4.14: Errors for $r_1(t)$ with EKF3

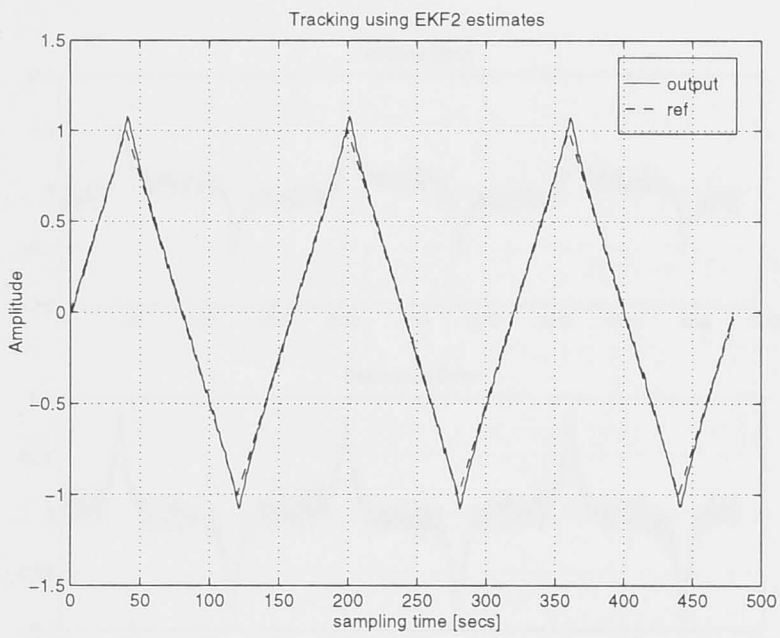


Figure 4.15: Tracking control for $r_2(t)$ with EKF2

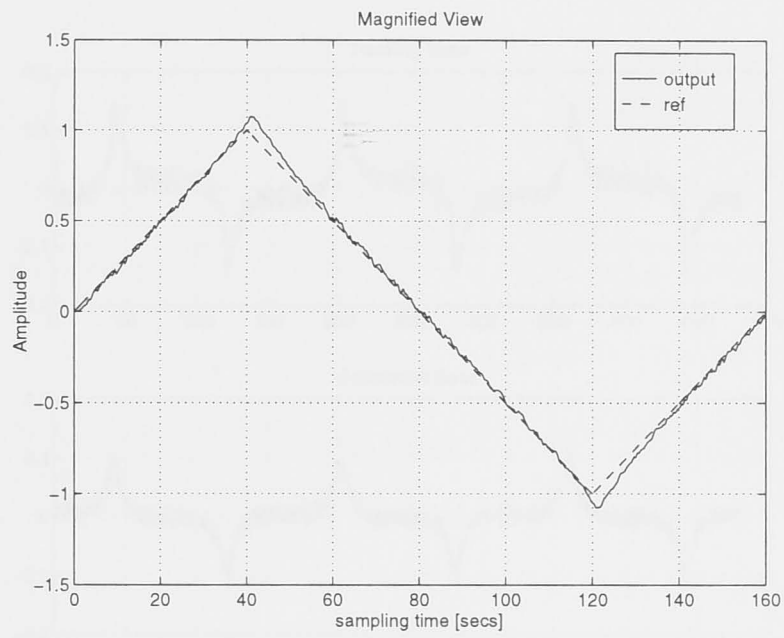


Figure 4.16: Magnified view

Table 4.3: Error variances for the tracking control with EKF3

Reference signal	Est. error variance	Tr. error variance
r_1	1.0773×10^{-4}	4.1290×10^{-4}
r_2	9.9043×10^{-5}	2.3775×10^{-4}

The estimates from either EKF2 or EKF3 are to be preferred to that from the LKF as demonstrated by the estimation and tracking error variances. EKF3, when employed instead of EKF2, increases the computational speed to a significant extent, though the error variances are smaller with EKF2. Thus the trade-off is between the estimation

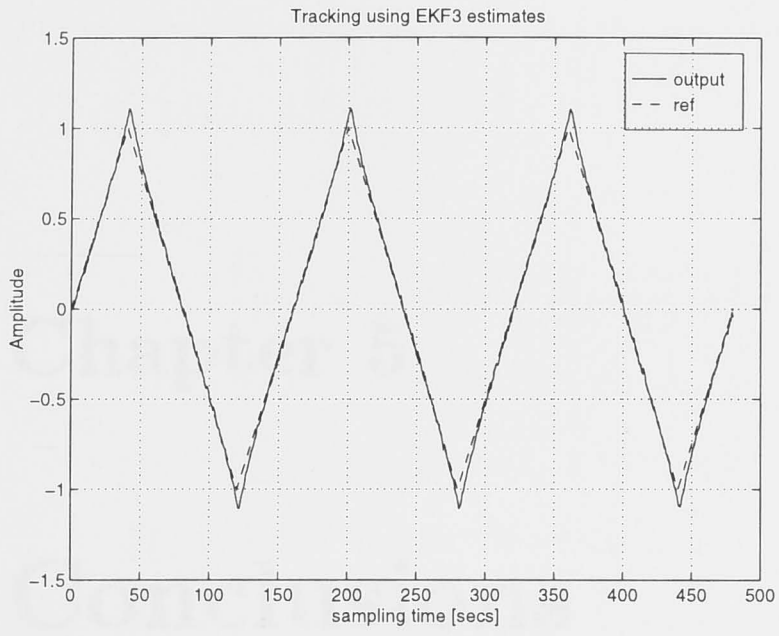


Figure 4.17: Tracking control for $r_2(t)$ with EKF3

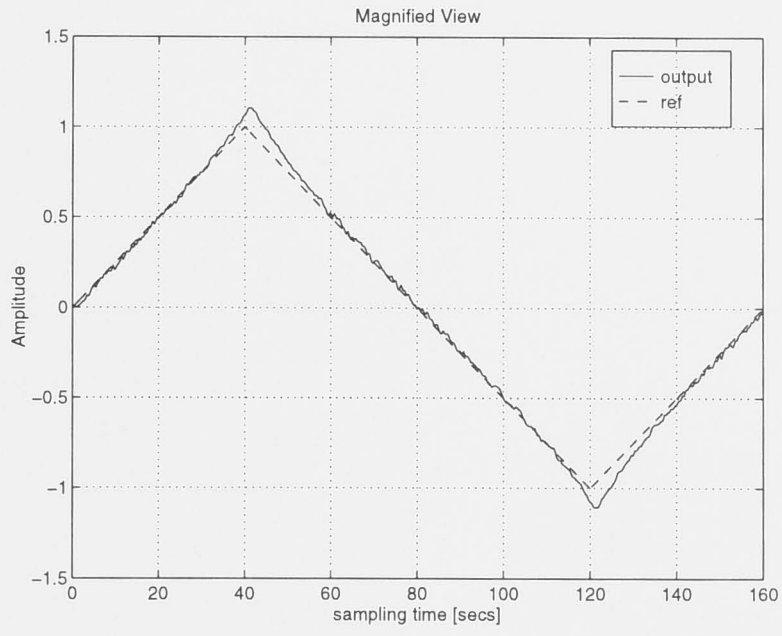


Figure 4.18: Magnified view

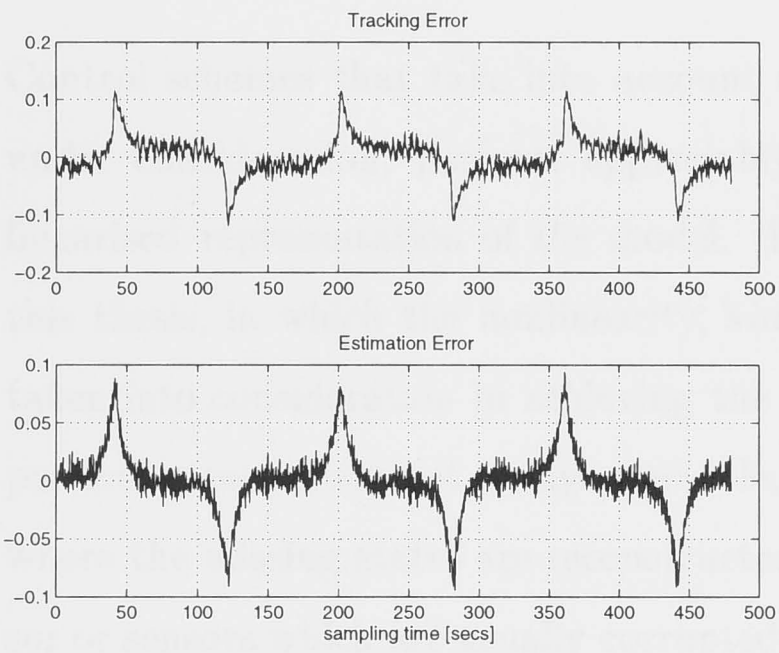


Figure 4.19: Errors for $r_2(t)$ with EKF2

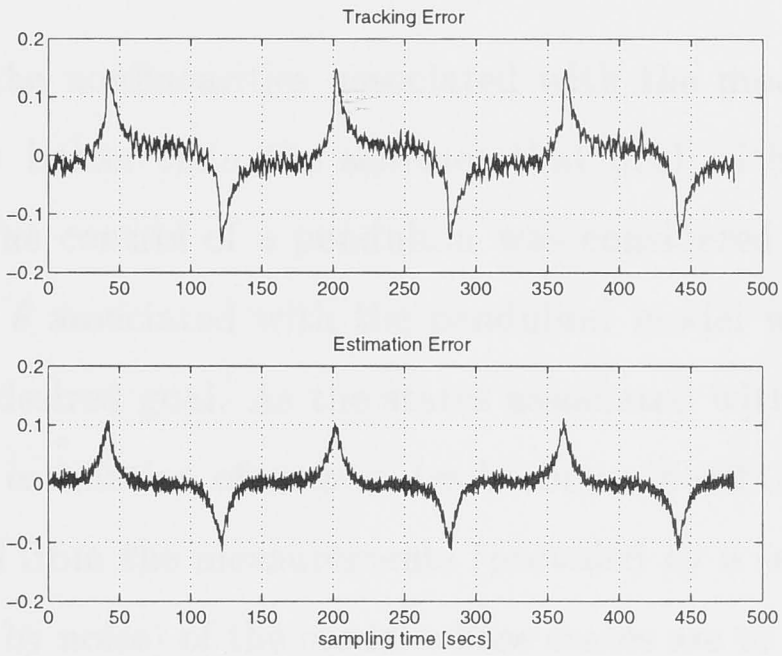


Figure 4.20: Errors for $r_2(t)$ with EKF3

accuracy and the computational speed. The whole discrete-estimator based tracking control is clearly a good method of tracking, in the presence of process and measurement noises, as demonstrated by the tracking error variances. Also the whole discrete estimator based controller has the advantage of increased computational speed and does not demand any additional hardware configurations. Thus the tracking control of the pendulum was accomplished. The next chapter provides the conclusions.

Chapter 5

Conclusions

Control schemes that take into account the nonlinearities associated with the model under consideration, perform appreciably better than the schemes that deal with a linearised representation of the model. The control of a pendulum was considered in this thesis, in which the nonlinearity, $sin\ \theta$ associated with the pendulum model was taken into consideration in achieving the desired goal. As the states associated with a particular model are not always available, estimation of such states becomes inevitable where the missing states are reconstructed from the measurements (provided by a sensor or sensors which are usually corrupted by noise) of the model whose states are to be estimated. Three sensors, the encoder, the rate-gyro and the tilt-sensor, were analysed for providing the measurements for the estimation.

The selection of a particular sensor is dependent on many factors. In a practical setting, no single sensor satisfies all the requirements and criteria as certain sensors are suitable for certain applications. The conclusions are presented under the following categories:

Application type

As discussed in section 2.3, in certain applications like biped locomotion, the encoder is not a feasible option since it complicates the design of the biped with the added weight, which makes the already complicated design [KB93] messier.

The tilt-sensor is ideal in situations like these, as it is compact and provides comparable readings.

Cost

The encoder is much cheaper as compared to the tilt-sensor as the tilt-sensor involves a signal-conditioner as indicated in section 2.1.3. The typical cost of an encoder (HEDS series), a rate-gyro (muRata ENV-05S) and a tilt-sensor with the signal-conditioner (Spectron L211U and SA40012) are $25.00, $350.00 and $650.00 respectively. Thus from a financial point of view, the encoder is the best sensor and with accurate digital encoders, precise position measurements can be obtained.

Operational accuracy

The tilt-sensor has an inherent delay associated with its operation as compared to the encoder. This was demonstrated by means of experiments in section 2.5.3. So the tilt-sensor is not a suitable sensor when instantaneous outputs are required. The encoder is the favourable choice in this situation. Also with the rate-gyro, noise tends to corrupt the measurements much more than compared to the other sensors (see section 2.5.2).

Estimation accuracy

Referring to the estimation error variances that were tabulated in Tables 3.4, 3.6 and 3.14, if a choice is to be made between the encoder and the tilt-sensor, the encoder is to be selected.

Employing the combination of two sensors depends on each of the sensors subject to the conclusions derived under the above categories. Also note that the combination of two sensors is obviously unattractive from a financial point of view as two sensors increase the cost factor significantly. The encoder and rate-gyro combination is the best sensor in this category.

The unknown parameters associated with the pendulum and tilt-sensor models were obtained by the parameter estimation algorithm that was developed in chapter 2 (see section 2.3). This algorithm was based on the assumption of small sampling intervals. The factors detrimental to the operation of the sensors were analysed and it was found that the noise associated with the sensors is the major source of error. The variance of the noise associated with each sensor was determined. The actuator was tested for the presence of dead zone and it was found that the actuator considered for analysis (Precision DC motor RS718-981) has a negligible amount of dead zone.

The Kalman filter (linear) is an optimal estimator that can be employed for the state estimation of a linear model. Unfortunately, nonlinear estimation approaches are more complex as compared to the linear estimation approaches. The extended Kalman filtering is a nonlinear estimation approach in which the linear Kalman filtering techniques are extended to the nonlinear case. When the model is continuous and the measurements are discrete, the standard extended Kalman filtering approach, EKF1, is the most commonly employed. The main computational burdens in this approach are the state estimate propagation and the error covariance propagation. A linear Kalman filter, LKF, and three extended Kalman filters, EKF1, EKF2 and EKF3 were employed for the state estimation of the pendulum model. A class of nonlinear systems that can be represented by the extended generalised Wiener model structure have the form in which, a linear system $H(s)$ is in cascade with a polynomial type nonlinearity followed by another linear system $G(s)$. By using the Volterra series and the approximation $sin\ \theta \approx \theta - (\theta^3/6)$, the approximate pendulum model was represented in such a form for which $H(s) \equiv G(s)$. General discrete extended Kalman filtering results for models that fall under this model structure were developed and these results were extended to the pendulum and the combined pendulum and tilt-sensor models. In EKF2 and EKF3, as opposed to the EKF1 approach, the extrapolations are done once in every sampling interval and thus this type of approach has the distinct advantage of increased computational speed. This was demonstrated for the nonlinear case and the corresponding results were given in Tables 3.15 and 3.16. EKF2 and EKF3 perform as good as EKF1

which was demonstrated in terms of both the estimation accuracies (see Tables 3.4, 3.6 and 3.14) and simulations (see Figures 3.4, 3.5 and 3.8). Comparing the results of the linear Kalman filtering approach with that of the extended Kalman filtering approaches for the pendulum model, it was inferred that, at higher swing-angles where the nonlinearities are severe, the nonlinear filtering approaches have to be followed.

The feedback linearisation technique which attempts to cancel the nonlinearity associated with the pendulum was employed and with the help of the estimates generated by the discrete filtering approaches, the tracking control of the pendulum was accomplished. Both linear and nonlinear estimation approaches were considered. For the tracking control of the pendulum with linear estimates, the estimates generated by the LKF were employed and for the tracking control with nonlinear estimates, the EKF2 and EKF3 estimates were employed. From the tracking error variances tabulated in Tables 4.1, 4.2 and 4.3, again it was found that the nonlinear tracking control is much superior to the linear tracking control.

Further research

The approximation of $sin\ \theta$ can be extended to include an additional higher order term, namely $\frac{\theta^5}{5!}$. This essentially complicates the analysis using the Volterra series (due to the addition of a kernel) and thus deriving the corresponding input-output relations using the tensor techniques get more involved. Variants of the extended Kalman filter, the iterated EKF and the second-order filters (see section 3.2) can also be employed for state estimation. With such inclusions, the computational burden would be increased significantly. But the bright side of the inclusion may be that this could result in much better estimates, the trade-off being increased accuracy to increased complexity filter algorithms.

Based on the techniques developed in this thesis, it is possible to extend the estimation strategies to a multi-link robot arm by following the derived results (with the appropri-

ate changes) which could prove to be advantageous in the estimation of the associated states and thus would lead to a numerically efficient control algorithm.

References

[AD87] V. Anantharam and C.A. Desoer, *Tracking and Disturbance Rejection of MIMO Nonlinear Systems with a PI or PS Controller*, International Journal of Control (1987), 1023–1034.

[Akb94] S.M. Akbar, *The Inverted Pendulum*, Undergraduate Thesis, Department of Engineering, ANU, Australia (1994).

[AM79] B.D.O. Anderson and J.B. Moore, *Optimal Filtering*, Information and System Sciences Series, Prentice Hall Inc., 1979.

[AM91] B.D.O. Anderson and J.B. Moore, *Kalman Filtering: Whence, What and Whither?*, Mathematical System Theory. The Influence of R.E. Kalman, Springer-Verlag, 1991.

[AW89] K.J. Astrom and B. Wittenmark, *Adaptive Control*, Addison-Wesley Publ. Co., 1989.

[Ben96] M.D.D. Benedetto, *Synthesis of an Internal Model for Nonlinear Output Regulation*, Automatica (1996), 1285–1301.

[Ber94] H. Berghuis, *Model-based Robot Control - from Theory to Practice*, Ph.D. thesis, University of Twente, The Netherlands, 1994.

[BR75] S.L. Baumgartner and W.J. Rugh, *Complete Identification of a Class of Nonlinear Systems from Steady-State Frequency Response*, IEEE Transactions on Circuits and Systems (September 1975), 753–758.

[Bro70] R.W. Brockett, *Finite Dimensional Linear Systems*, John Wiley & Sons Inc., 1970.

[Bro73] R.W. Brockett, *Lie Algebras and Lie Groups in Control Theory*, Proceedings of N.A.T.O. Advanced Study Institute London, Edited by D.Q. Mayne and R.W. Brockett, Dordrecht, Reidel (1973).

[Bro76] R.W. Brockett, *Volterra Series and Geometric Control Theory*, Automatica (1976), 167–176.

[DFC71] V.I. Dotsenko, R.G. Faradzhev, and G.S. Charatisvhilli, *Properties of Maximal Length Sequences with p-levels*, Autom & Telemekh (1971).

[Dor93] R.C. Dorf, *The Electrical Engineering Handbook*, CRC Press, 1993.

[F+91] G.F. Franklin et al., *Feedback Control of Dynamic Systems*, Addison-Wesley Publ. Co., 1991.

[FW76] B.A. Francis and W.M. Wonham, *The Internal Model Principle of Control Theory*, Automatica (1976), 457–465.

[G+94] A. Grace et al., *Control System Toolbox*, 1994.

[GA93] M.S. Grewal and A.P. Andrews, *Kalman Filtering Theory and Practice*, Prentice Hall Inc., 1993.

[Gar73] A.G. Gardiner, *The Identification Testing of Nonlinear Processes*, Proceedings of the International Symposium and Process Identification by Correlation and Spectral Techniques (January 1973), 161–168.

[Gel74] A. Gelb, *Applied Optimal Estimation*, MIT Press, 1974.

[Geo59] D.A. George, *Continuous Nonlinear Systems*, Tech. Report #355, MIT, July 1959.

[Gil78] E.G. Gilbert, *Bilinear and 2-Power Input-Output Maps: Finite Dimensional Realizations and the Role of Functional Series*, IEEE Transactions on Automatic Control (1978), 1418–1425.

[HW84] J.S.A. Hepburn and W.M. Wonham, *Error Feedback and Internal Models on Differentiable Manifolds*, IEEE Transactions on Automatic Control (1984), 397–403.

[IB90] A. Isidori and C.I. Byrnes, *Output Regulation of Nonlinear Systems*, IEEE Transactions on Automatic Control (1990), 131–140.

[Jen96] P. Jensfelt, *Sensory Processing for Control of a Simple Robot Model*, Master's thesis, KHT, Sweden, December 1996.

[JR88] H. Jie and W.J. Rugh, *On a Nonlinear Multivariable Servomechanism Problem*, Tech. report, JHU/ECE-88/04.2, 1988.

[K+89] R.D. Klafter et al., *Robotic Engineering - An Integrated Approach*, Prentice Hall Inc., 1989.

[KB93] J. Kieffer and R. Bale, *Walking Viability and Gait Synthesis for a Novel Class of Dynamically-Simple Bipeds*, Informatica (1993), 145–155.

[Kha92] H. Khalil, *Nonlinear Systems*, MacMillan, 1992.

[KK88] P.K. Khosla and T. Kanade, *Experimental Evaluation of Nonlinear Feedback and Feed-forward Control Schemes for Manipulators*, International Journal of Robotics Research (1988), 18–28.

[Lju87] L. Ljung, *System Identification Theory for the User*, PTR Prentice Hall Inc., 1987.

[Lju89] L. Ljung, *System Identification Toolbox*, 1989.

[M+96] B.K. Maner et al., *Nonlinear Model Predictive Control of a Simulated Multivariable Polymerization Reactor using Second-order Volterra Models*, Automatica (1996), 1285–1301.

[Mac46] C.C. MacDuffee, *The Theory of Matrices*, Chelsea Publ. Co., 1946.

[Mot92] Precision DC Motors, *Precision DC motors, Gear Boxes and Servo Mounting Kit*, November 1992.

[NG69] P.N. Nikifaruk and M.M. Gupta, *A Bibliography of the Properties, Generation and Control Systems Applications of Shift Register Sequence*, International Journal of Control (1969).

[Pac95] Hewlett Packard, *Two Channel Optical Incremental Encoder Module - HEDS-9100 Series*, 1995.

[Pel70] J.B. Peller, *Nonlinear Smoothing Techniques*, Theory and Applications of Kalman Filtering - Edited by C.T. Leondes (1970), 185–201.

[PM82] G.A. Parker and E.L. Moore., *Practical Nonlinear System Identification using a Modified Volterra Series Approach*, Automatica (1982), 85–91.

[QD96] Z. Qu and D.M. Dawson, *Robust Tracking Control of Robot Manipulators*, IEEE Press, 1996.

[Rug81] W.J. Rugh, *Nonlinear System Theory: The Volterra/Wiener Approach*, The Johns Hopkins University Press, 1981.

[San77] J. Sandor, *The Analysis and Design of Polynomic Nonlinear Control Systems*, Ph.D. thesis, UNSW Australia, 1977.

[SJ75] K.S. Shanmugam and M.T. Jong, *Identification of Nonlinear Systems in Frequency Domain*, IEEE Transactions on Aerospace and Electronic Systems (November 1975), 1218–1225.

[SL91] J.J.E. Slotine and W. Li, *Applied Nonlinear Control*, Prentice Hall Inc., 1991.

[SP78] R.J. Simpson and H.M. Power, *Correlation Techniques for the Identification of Non-linear Systems*, Meas & Control (1978), 316.

[Ste86] R.F. Stengel, *Stochastic Optimal Control: Theory and Application*, John Wiley & Sons Inc., 1986.

[STI93] Spectron Systems Technology Inc., *Electrolytic Level Sensors*, 1993.

[SW78] J. Sandor and D. Williamson, *Identification and Analysis of Nonlinear Systems by Tensor Techniques*, International Journal of Control (June 1978), 853–878.

[TA94] G+G Technics AG, *Signal Conditioner SA40012*, Schweiz, Germany, 1994.

[TK96] G. Tao and P.V. Kokotović, *Adaptive Control of Systems with Actuator and Sensor Nonlinearities*, John Wiley & Sons Inc., 1996.

[Vac95] R.J. Vaccaro, *Digital Control: A State-Space Approach*, McGraw-Hill Series in Electrical Engineering, 1995.

[Van79] H.L. VanTrees, *Functional Techniques for the Analysis of the Nonlinear Behavior of Phase-Locked Loops*, IEEE Transactions on Communication (1979), 542–556.

[Wei93] Dipl.-Inform. Gerhard Weiβ, *Usage of a Vibrating Gyroscope for Orientation Estimation*, Tech. report, University of Kaiserslautern - CS Dept., 1993.

[Wil91] D. Williamson, *Digital Control and Implementation Finite Wordlength Considerations*, Prentice Hall International Series in Systems and Control Engineering, 1991.