Chapter 2

BASIC KNOWLEDGE

This chapter provides the details on the basic knowledge used in this thesis. It is divided into three parts :

1. Sensor-relating problems and the implementation of the Kalman filter.
2. Linear state-feedback controller.
3. LPV controller synthesis.

2.1 Sensor Fusion and Kalman Filtering

2.1.1 Sensor Processing

Both the gyroscope and accelerometer are setup inside the MIP for testing the signal. Under this setup, a test program is written to capture the signal from the sensors through a number of set points. The actual angle of accelerometer can be obtained based on the swinging of the pendulum. Next, a comparison is done between the actual angle of the MIP rotation captured by the encoder and the angle recorded by the accelerometer. This test platform ensures that the accelerometer manipulation is as exact as possible.The digital rate gyroscope installed can only provide a measure of the instantaneous angular change. Furthermore, the gyroscope has a rest average value (value of the gyroscope when it is not moving) which has to be reset at every measurement to get an accurate velocity measurement. It is found that the gyroscope's rest average value drifts with time. This introduces significant errors in the velocity and angular measurements (Figure 2.2). The liquid based digital accelerometer installed on the robot gives a good measurement of absolute tilt angle. The accelerometer data is processed almost the same way as the gyroscope. Several problems observed while testing the accelerometer are the changer of the rest value of the accelerometer every time when it is initialized and the output signal tends to exhibit a significant amount of noises (Figure 2.3).

2.1.2 The Need for Sensor Fusion

Initial tests performed as described above on the gyroscope and the accelerometer showed that the use of either one of the sensors alone is unable to provide sufficient and more impor-

Figure 2.1: Servo test platform.

tantly reliable information in order to balance the robot. The gyroscope provides a measure of instantaneous angular change but it produces a significant drift when the gyroscope is operating. This may be due to the operating temperature or inherent characteristics of the gyroscope itself. On the other hand, the accelerometer provides an absolute measure of tilt angle, but the output signal is often corrupted with noises.

To overcome these problems, a signal level sensor fusion technique using the Kalman filter is proposed. Signal level fusion refers to the combination of signals of a group of sensors with the objective of providing a signal that is usually of the same form as the original signals but of greater quality. In this case, the accelerometer is used to eliminate the drift from the gyroscope signal via the Kalman filter. As a result, an accurate estimate of the angle and its derivative term are obtained.

2.1.3 The Kalman Filter

The Kalman filter is a set of mathematical equations for optimal recursive data processing algorithm that provides the solution of the least squares method. It incorporates all information that can be provided and processes all available measurements, regardless of their precision to estimate the current value of the variables of interest with the use of a knowledge of the system and measurement dynamics, the statistical description of the system noises, mea-

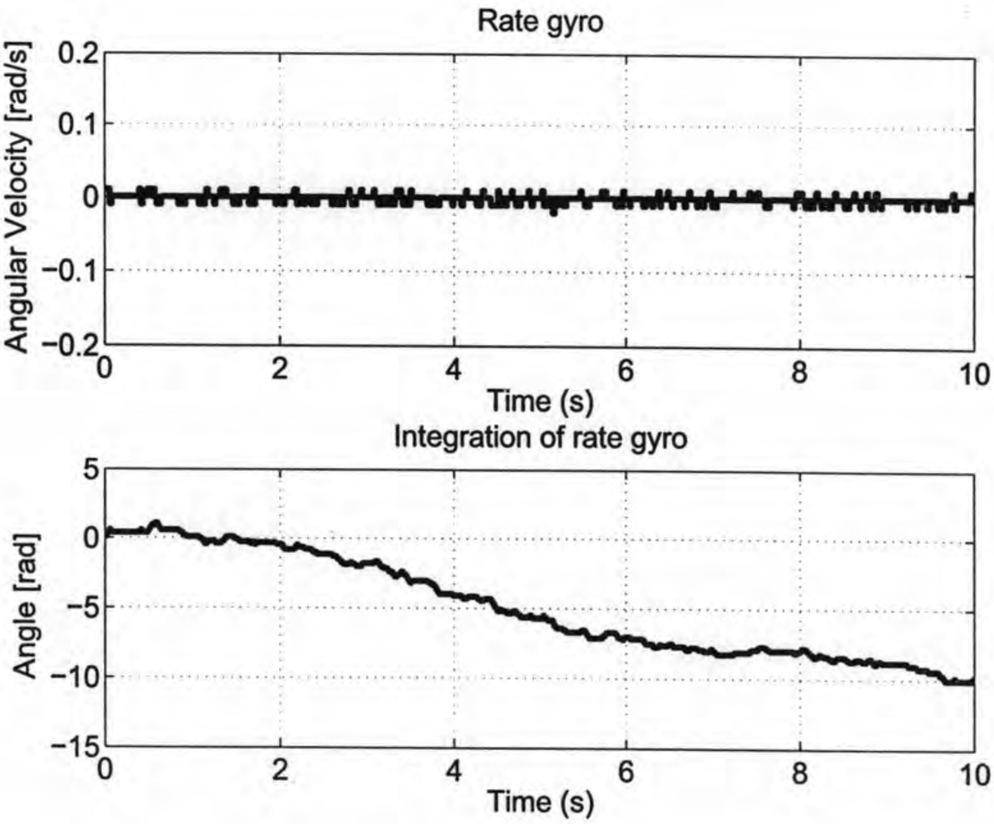

Figure 2.2: Rate gyro and its drift after integration.

surement errors, uncertainty in the model dynamics, and any available information about the initial condition of the variables of interest.

Under the assumptions that the model for system of interest is linear and the noise values are not correlated in time, the Kalman filter is optimized effectively to a criterion that makes sense.

The Kalman filter does not require all previous data to be kept in storage and reprocessed every time a new measurement is taken. With this behavior, the Kalman filter can be implemented as a computer program in a small micro-controller despite the usual implication that a filter is a connection of electrical networks in a box.

2.1.4 The Discrete Kalman Filter Algorithm

- **State Equation**

  The Kalman filter can be applied to estimate the states when the system of interest is adequately modeled in the form of a linear stochastic differential equation. In other words, the Kalman filter can be utilized to estimate the states of a system when the

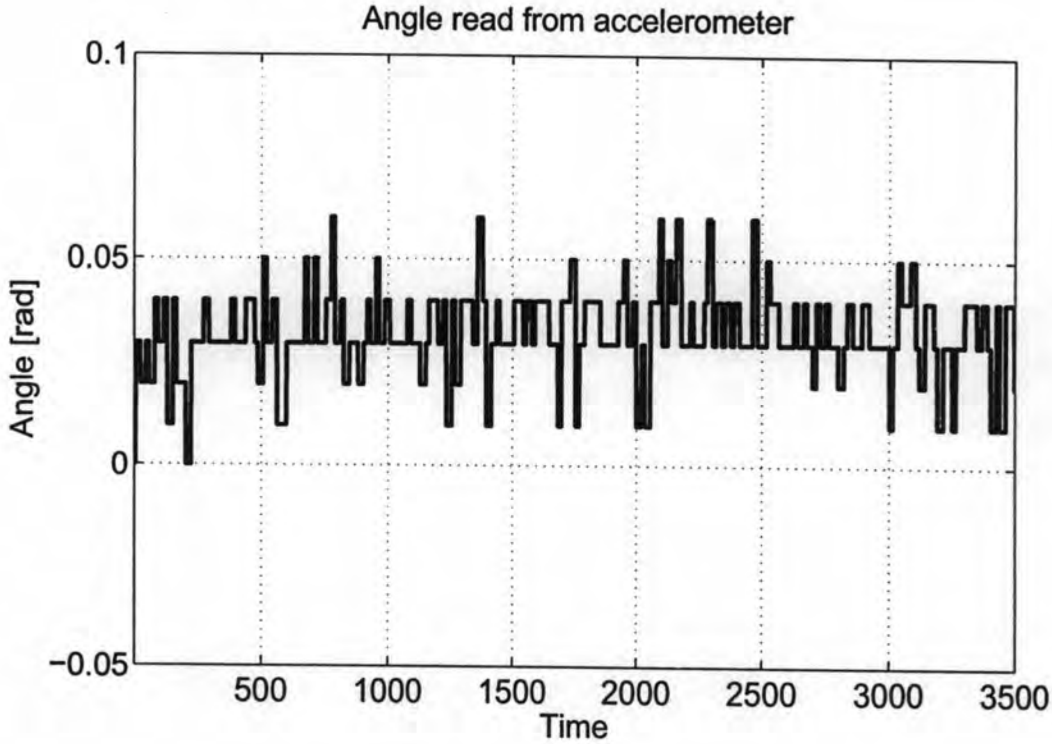

Figure 2.3: Noisy angle read from accelerometer.

system of interest can be modeled in state space form. Equation (2.1) represents the state process of the system in white noise notation.

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \tag{2.1}$$

where $x \in R^{n\times 1}$ is process state vector, $u \in R^{l\times 1}$ is deterministic control input, $A \in R^{n\times n}$ is system dynamics matrix, $B \in R^{n\times l}$ is deterministic input matrix and w is process noise.

- **Measurement Equation**

Measurements for the system is taken at discrete time points, these measurements are included in the measurement vector and can be modeled by the relation,

$$z_k = Hx_k + v_k \tag{2.2}$$

Where $z \in R^m$ is discrete time measurement process, $H \in R^l$ is measurement matrix and v is discrete time white Gaussian noise.

Equation (2.2) states that the measurements are dependent on the state of the system and are related by the measurement matrix with an addition of noise into the measurements. The measurements for the system are often obtained at equally spaced time, but this is not obligatory.

- **System and Measurement Noise**

  It is assumed that the noise vector w_k and v_k are white Gaussian noise with the following statistics

$$
\begin{aligned}
E\{w_k\} &= 0 \\
E\{w_i \times w_j^T\} &= Q_i\delta(i-j) \\
E\{v_i \times v_j^T\} &= R_i\delta(i-j)
\end{aligned}
$$

  where R_i is a positive definite matrix, which means that all components of the measurement vector are corrupted with noise, and there is no linear combination of these components that would be noise free. Since the distributions of w_k and v_k are assumed Gaussian, this is equivalent to assuming that they are uncorrelated with each other.

- **Initial Conditions**

  The state differential equation (2.1) is propagated from the initial condition x_0 and for any particular operation of the real system, the initial state assumes a specific value x_0. However, because this value may not be known precisely in advance, it would be modeled as a random vector that is normally distributed. Thus, the description of x_0 is completely specified by the mean $\hat{x}_0$ and covariance P_0 as

$$
\begin{aligned}
E\{x_0\} &= \hat{x}_0 \\
E\{[x_0 - \hat{x}_0][x_0 - \hat{x}_0]^T\} &= P_0
\end{aligned}
$$

  where P_0 is a symmetric and positive semi-definite matrix. This matrix provides the expected value of the difference between the true state and the estimated state. The diagonal elements of this matrix provide the variance of each state variable from its true value.

- **Kalman Filter Equations**

  The Kalman filter estimates a process by using a form of feedback control. The states of the process are estimated by the filter at a certain point. Therefore, the equations for the Kalman filter fall into two groups, the time update equations and measurement update equations.

The time update equations estimate the current state and error covariance to obtain a priori estimates for the next step. On the other hand, the measurement update equations are responsible for incorporating a new measurement into a priori estimate to obtain an improved and a posteriori estimate. The time update equations are modeled as the predictor equations, while the measurements update equations are taken as the corrector equations. This concept is illustrated in Figure 2.4. Considering two mea-

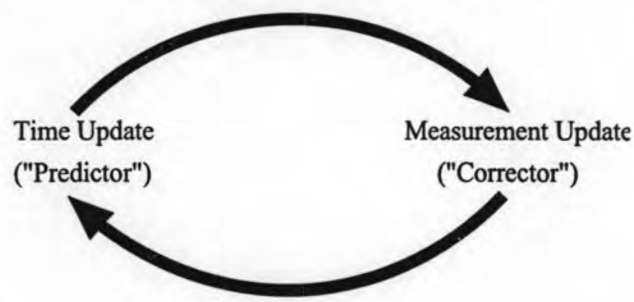

Figure 2.4: Kalman Filter's "Predictor-Corrector" structure.

surement times, $k-1$ and k, and time propagation estimates from the point just after the measurement at time $k-1$ has been incorporated into the estimate, to the point after the measurement at time k is incorporated. This is represented by time $(k-1)^+$ to time k^+. The optimal state estimate is propagated from measurement time $k-1$ to measurement time k by the relations.

$$\hat{x}_k^- = F\hat{x}_{k-1}^+ + Bu_{k-1}^+ \tag{2.3}$$

$$P_k^- = FP_{k-1}^+F^T + Q \tag{2.4}$$

Equation (2.4) defines the conditional covariance matrix of the error in predicting the state x. As the state measurement z_k becomes available at time k, the estimate is updated by defining the Kalman filter gain K_k and employing it in both the mean and covariance relations.

$$K_k = P_k^- H_k^T [H_k P_k^- H_k^T + R_k]^{-1} \tag{2.5}$$

$$\hat{x}_k^+ = \hat{x}_k^- + K_k[z_k - H_k\hat{x}_k^-] \tag{2.6}$$

$$P_k^+ = P_k^- - K_k H_k P_k^- \tag{2.7}$$

The error estimate, that is the errors committed by the estimator $\hat{x}_k^-$ for a particular measurement is given by,

$$E = H_k P_k^- H_k^T + R_k \tag{2.8}$$

Alternatively, equation (2.5) can be rewritten as

$$K_k = P_k^- H_k^T E^{-1} \tag{2.9}$$

The difference between the true measurement z_k and its best prediction before it is actually taken is known as the measurement residual r_k. This is defined by the following equation

$$r_k = z_k - H_k \hat{x}_k^- \tag{2.10}$$

The residual is then passed through an optimal weighing matrix K_k to generate the correction term to be added to $\hat{x}_k^-$ to obtain $\hat{x}_k^+$.

2.1.5 The Extended Kalman Filter

The Extended Kalman Filter provides a method applying the Kalman Filter technique to a nonlinear problem by linearizing the estimation around the current estimate using partial derivatives.

- **State Equation**

  For the Extended Kalman Filter, the process is governed by the nonlinear stochastic difference equations

  $$x_k = f(x_{k-1}) + w \tag{2.11}$$

  where x is a vector of the system states and $f(x)$ is a nonlinear function of those states.

- **Measurement Equation**

  The measurement equation for the Extended Kalman Filter are considered a nonlinear function of the states according to

  $$z_k = h(x_{k-1}) + v \tag{2.12}$$

- **System and Measurement Noise**

  The system and measurement noise v and w for the Extended Kalman Filter are modeled as a random process with zero mean, which is similar to the system and measurement noise statistics modeled for a normal Kalman filter.

- **Kalman Filter Equations**

In order to apply the continuous Ricatti equation, the nonlinear system and measurement equations are linearized with a first order approximation using the Jacobian matrix.

- F is the Jacobian matrix of partial derivatives of f with respect to x, that is

$$F = \left.\frac{\partial f(x)}{\partial x}\right|_{x=\hat{x}} \tag{2.13}$$

- H is the Jacobian matrix of partial derivatives of h with respect to x, that is

$$H = \left.\frac{\partial h(x)}{\partial x}\right|_{x=\hat{x}} \tag{2.14}$$

Now the optimal state estimate propagation from measurement time $k-1$ to measurement time k can be represented with the following equations

$$\hat{x}_k^- = f(x_{k-1}^+) \tag{2.15}$$

$$P_k^- = FP_{k-1}^+F^T + Q \tag{2.16}$$

As before, when the state measurement z_k becomes available at time k, the estimate is updated by the following equations

$$K_k = P_k^- H_k^T [H_k P_k^- H_k^T + R_k]^{-1} \tag{2.17}$$

$$\hat{x}_k^+ = \hat{x}_k^- + K_k[z_k - H_k \hat{x}_k^-] \tag{2.18}$$

$$P_k^+ = P_k^- - K_k H_k P_k^- \tag{2.19}$$

The linearization of the process will be valid and will produce excellent results if the estimate of the states is good. In cases where the state estimates are not good, the filter will diverge quickly and produce a very poor estimate.

2.1.6 Filter Tuning for Performance

The performance of filter could vary greatly if the parameters are not properly adjusted. Therefore, this section intends to provide some insight into the characteristics of the filters parameters. The adjustable parameters are

- The initial covariance matrix P_0.
- The state estimate vector $\hat{x}_0$.
- The Q matrix and its corresponding random noise vector w.
- The R matrix and its corresponding random noise vector v.

2.1.7 Kalman Filter Design for Single Dimensional INS

In this thesis, a Kalman filter is designed to provide an estimate of tilt angle and its derivative for a single dimensional Inertial Navigation System (INS). This filter will attempt to use the data from an accelerometer to eliminate the measurement drift from the gyroscope signal via the Kalman filter. In the process, corruptive noises in the accelerometer reading will also be minimized.

An indirect feedback configuration will be used for the single dimensional Inertial Navigation System. This configuration is illustrated in the figure below

The filter compares the data from the sensors and the inertial system and uses this result to

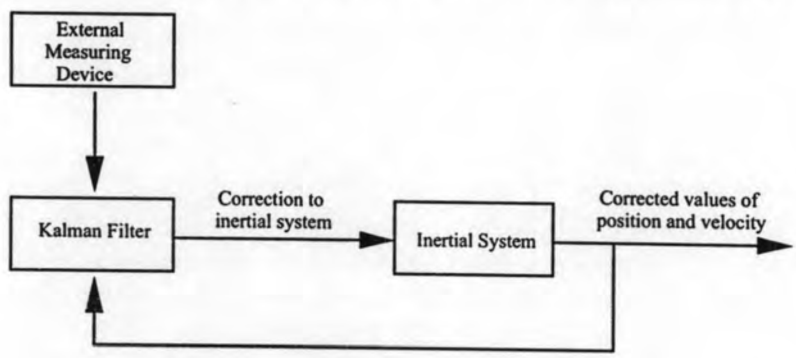

Figure 2.5: Indirect feedback Kalman filter.

estimate the errors in the system. By feeding back these error estimates to the INS to correct it, the inertial errors are not allowed to go unchecked. In this way the adequacy of the model is enhanced.

2.2 Linear State-Feedback Controller

Consider the following linear time invariant system with full state feedback

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{2.20}$$

$$u = -Kx(t) \tag{2.21}$$

We always assume that B is full rank. The matrices (A, B) will be either controllable, stabilizable, or neither. The block diagram for linear state-feedback controller is shown in Figure 2.6.

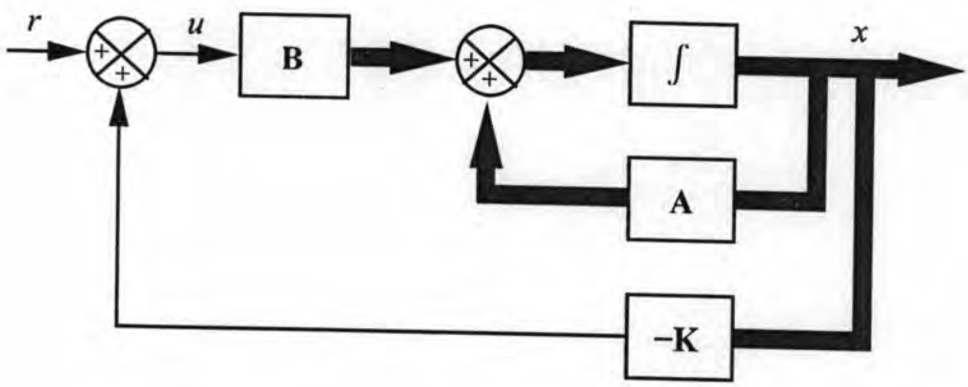

Figure 2.6: Block diagram for linear state-feedback controller.

2.2.1 LQR Controller

The Linear Quadratic Regulator (LQR) control is a modern state-space technique for designing optimal dynamic regulators. It refers to a linear system and a quadratic performance index according to

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{2.22}$$

$$x(0) = x_0 \tag{2.23}$$

The performance of an LQR system can be represented by an integral performance index. It enables a trade off between regulation performance and control effort via the performance index when the initial state x_0 is given.

$$J = \frac{1}{2}\int_0^\infty \left[x^T(t)Qx(t) + u^T(t)Ru(t)\right] dt \tag{2.24}$$

The control law for the LQR is specified as

$$u = -R^{-1}B^T Px \tag{2.25}$$

where $P = P^T \geqslant 0$ solves the following algebraic Ricatti equation

$$0 = PA + A^T P - PBR^{-1}B^T P + Q \tag{2.26}$$

The gain vector $K = R^{-1}B^T P$ determines the amount of control feedback into the system. The matrices R and Q, will balance the relative importance of the control input and state in the cost function (J) being optimized with a condition that the elements in both Q and R matrices are positive definite. The size of Q matrix depends on the size of the system's state matrix and R matrix is dependent on the number of control input to the system.

Using MATLAB, the algebraic Ricatti equation is solved and the control gain K is evaluated for different values of Q and R weighting matrices. The response of the system is simulated as well.

2.2.2 Pole-Placement Controller

The position of poles defines the stability of a system. According to control system theory, the poles of the system can be arbitrarily placed in the complex plane if the controllability matrix is of full rank. This matrix is defined by,

$$C = \begin{bmatrix} B & AB & A^2B & \dots & A^{n-1}B \end{bmatrix} \tag{2.27}$$

The control law for the Pole-placement controller is given as

$$u = -Kx \tag{2.28}$$

where u is the control voltage, x is the state parameters and K is the state feedback gain matrix.

Pole-placement control gives designers the option of relocating all the closed loop poles of the system. This is in contrast with the classical design using Bode plots and frequency response methods, whereby the designer can only hope to achieve a pair of complex conjugate poles that are dominant. The theory of Pole-placement control might look trivial, but since all other poles and zeroes may fall anywhere, meeting the design specification becomes a matter of trial and error. With the freedom of choice rendered by state-feedback comes the responsibility of selecting the poles judiciously.

The poles of the system have to be placed carefully as there are obvious costs associated with shifting pole locations. As a result, several simulation trials using MATLAB have to be performed to attain the best pole location that gives the desired response while not straining the control input.

Stability of the system can be guaranteed as long as all the poles of the system are in the Left-Half Plane (LHP) of the Pole-Zero map, but the question is where in the LHP should it be placed. If a fast response is desired, the poles can be placed further away from the imaginary axis. The further the poles are placed from the imaginary axis, the system will require a faster or a stronger actuator to perform the task.

2.3 LPV Synthesis Controller

Introducing Linear Parameter-Varying (LPV) systems as an intermediate system description in the controller synthesis enables a systematic way of obtaining the nonlinear controller in a linear-like fashion. The $\mathcal{H}_\infty$ controller synthesis method is the most established method for robust control and the LPV synthesis will result in a controller that is robust in a sense similar to the $\mathcal{H}_\infty$ controller. The linear matrix inequalities are used to compute the LPV controller.

2.3.1 Formulation of an LPV System

Instead of using standard linearization of the nonlinear model in many operating points, the nonlinear model is translated to an LPV system valid in the entire operating range. Such an LPV description is often called a quasi-LPV, this is to emphasize that a nonlinear system is involved in the description. The LPV description is not unique and finding an LPV description that is suitable for the gain scheduling approach is not a trivial task.
Consider the LPV system,

$$\begin{aligned} \dot{x} &= A(\delta(t))x + B_1(\delta(t))w_p + B_2(\delta(t))u \\ z_p &= C_1(\delta(t))x + D_{11}(\delta(t))w_p + D_{12}(\delta(t))u \\ y &= C_2(\delta(t))x + D_{21}(\delta(t))w_p \end{aligned} \tag{2.29}$$

where $x \in \mathbb{R}^n$ is state variable, $y \in \mathbb{R}^{n_y}$ is measured output, $w_p \in \mathbb{R}^{n_w}$ is exogenous input, $u \in \mathbb{R}^{n_u}$ is control signal, $z \in \mathbb{R}^{n_z}$ is exogenous output and $\delta(t) = (\delta_1(t) \cdots \delta_r(t))^T \in \delta \subseteq \mathbb{R}^r$ is varying parameter.
The parameter $\delta(t)$ is used in the LPV controller, such that a nonlinear controller $K(\delta(t)$ is obtained

$$\begin{aligned} \dot{x} &= A_c(\delta(t))x_c + B_c(\delta(t))y \\ u &= C_c(\delta(t))x_c + D_c(\delta(t))y \end{aligned} \tag{2.30}$$

while the closed-loop system is

$$\begin{aligned} \dot{\xi} &= \mathcal{A}(\delta(t))\xi + \mathcal{B}_p(\delta(t))w_p \\ z_p &= \mathcal{C}_p(\delta(t))\xi + \mathcal{D}_{pp}(\delta(t))w_p \end{aligned} \tag{2.31}$$

where $\xi^T = (x^T \quad x_c^T)^T$ is closed-loop state variable.

2.3.2 Linear Fractional Transformation

Consider the matrix P of the dimension $(n_1 + n_2) \times (m_1 + m_2)$ and partition it as follows:

$$P = \begin{pmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{pmatrix} \tag{2.32}$$

Let the matrices Δ and K have dimension $m_1 \times n_1$ and $m_2 \times n_2$, respectively. We adopt the following notation for the lower and upper linear fractional transformations

$$\mathcal{F}_l(P, K) \triangleq P_{11} + P_{12}K(I - P_{22}K)^{-1}P_{21} \tag{2.33}$$

$$\mathcal{F}_u(P, \Delta) \triangleq P_{22} + P_{21}\Delta(I - P_{11}\Delta)^{-1}P_{12} \tag{2.34}$$

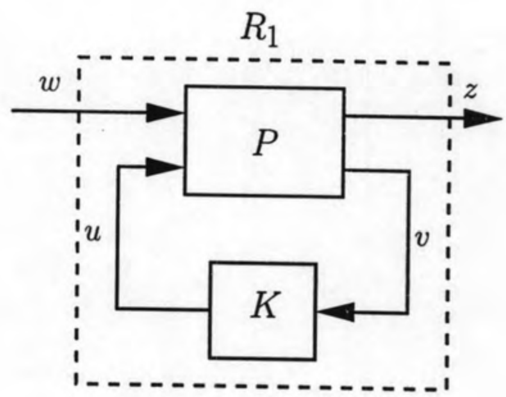

Figure 2.7: R_1 as lower LFT in terms of K.

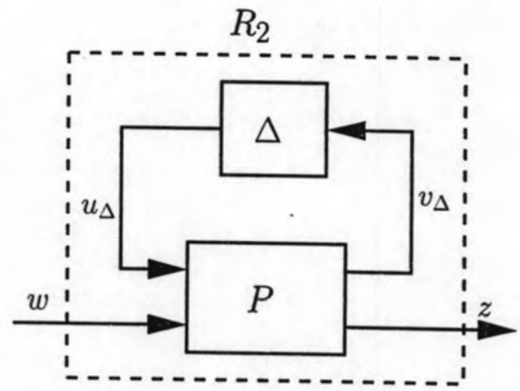

Figure 2.8: R_2 as upper LFT in terms of Δ.

2.3.3 $\mathcal{H}_\infty$ Optimal Control Synthesis

The standard $\mathcal{H}_\infty$ optimal control problem is to find all stabilizing controller K which minimize

$$\|\mathcal{F}_l(P,K)\|_\infty = \max_{w_p \in \mathcal{L}_2, w_p \neq 0} \frac{\|z_p\|_2}{\|w_p\|_2} \tag{2.35}$$

where $\|z_p(t)\|_2 = \sqrt{\int_0^\infty \sum_i |z_{pi}|^2 dt}$ is the 2-norm of the vector signal.
Let γ_{min} be the minimum value of $\|\mathcal{F}_l(P,K)\|_\infty$ over all stabilizing controller K. Then the $\mathcal{H}_\infty$ sub-optimal control problem is: given a $\gamma > \gamma_{min}$, find all stabilizing controllers K such that.

$$\|\mathcal{F}_l(P,K)\|_\infty < \gamma$$

2.3.4 LPV Design with Full Scalings

In LPV control, it is assumed that parameters Δ that enter the system are not unknown but they can be measured online. This allows, among other applications, to approach (robust) gain-scheduling synthesis problems for nonlinear control systems.

In this thesis, extend block-diagonal parameter matrices with the corresponding block-diagonal scalings to full block scalings are sketched.

Adjusted to the structure of (2.29)

$$\begin{aligned} \dot{x} &= A(\delta(t))x + B_\delta(\delta(t))w_\delta + B_p w_p + B_u u \\ z_\delta &= C_\delta x + D_{\delta\delta} w_\delta + D_{\delta p} w_p + D_{\delta u} u \\ z_p &= C_p(\delta(t))x + D_{p\delta} w_p + D_{pp} w_p + D_{pu} u \\ y &= C_y x + D_{y\delta} w_p + D_{yp} w_p, \quad w_\delta = \Delta(\delta(t)) z_\delta \end{aligned} \tag{2.36}$$

The measured parameter curve entering the controller is assumed to be also in a linear fractional fashion. Hence an LPV controller is defined by scheduling the LTI system

$$\begin{aligned} \dot{x}_c &= A_c x_c + B_c \begin{pmatrix} y \\ w_c \end{pmatrix} \\ \begin{pmatrix} u \\ z_c \end{pmatrix} &= C_c x_c + D_c \begin{pmatrix} y \\ w_c \end{pmatrix}, \quad w_c = f_c(\Delta(\delta(t))) z_c \end{aligned} \tag{2.37}$$

The controller is hence parameterized through the matrix A_c, B_c, C_c, D_c, and through a possibly nonlinear scheduling function $f_c(\cdot)$ as shown in Figure (2.9). The block diagram in

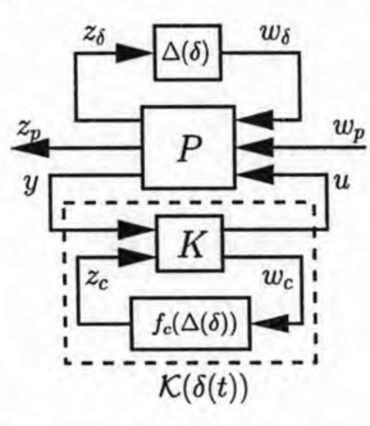

Figure 2.9: LPV controller $K(\delta(t))$

Figure (2.9) may be written as

$$z_p = \mathcal{F}_l(\mathcal{F}_u[P, \Delta(\delta(t))], \mathcal{F}_l[K, f_c[\Delta(\delta(t))]])w_p \tag{2.38}$$

The goal is to construct an LPV controller that renders the QP specification with index P_p for the channel $w_p \longrightarrow z_p$ for all possible parameters curves w_c satisfied. From (2.39) may transform as

$$z_p = \mathcal{F}_u(\mathcal{F}_l(P_a, K), \Delta_e(\delta(t)))w_p \tag{2.39}$$

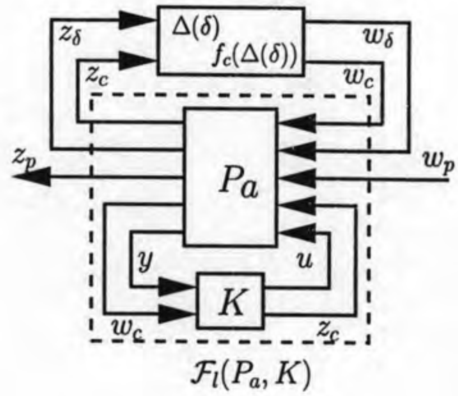

Figure 2.10: LPV augmented system $\mathcal{F}_l(P_a, K)$

From Figure (2.10) P_a is obtained by scheduling the LTI system

$$\begin{pmatrix} \dot{x} \\ \hline z_\delta \\ \hline z_c \\ \hline z_p \\ \hline y \\ w_c \end{pmatrix} = \left(\begin{array}{c|cc|c|cc} A & B_\delta & 0 & B_p & B_u & 0 \\ \hline C_\delta & D_{\delta\delta} & 0 & D_{\delta p} & D_{\delta u} & 0 \\ 0 & 0 & 0 & 0 & 0 & I \\ \hline C_p & D_{p\delta} & 0 & D_{pp} & D_{pu} & 0 \\ \hline C_y & D_{y\delta} & 0 & D_{yp} & 0 & 0 \\ 0 & 0 & I & 0 & 0 & 0 \end{array}\right) \begin{pmatrix} x \\ \hline w_\delta \\ \hline w_c \\ \hline w_p \\ \hline u \\ z_c \end{pmatrix} \tag{2.40}$$

$$\begin{pmatrix} w_\delta \\ w_c \end{pmatrix} = \Delta_e(\delta(t)) \begin{pmatrix} z_\delta \\ z_c \end{pmatrix}$$

with the parameters as

$$\Delta_e(\delta(t)) = \begin{pmatrix} \Delta(\delta(t)) & 0 \\ 0 & f_c(\Delta(\delta(t))) \end{pmatrix} \tag{2.41}$$

The LPV problem with which is started out is equivalently reformulated to robust performance design problem as discussed previously : find an LTI controller (2.37) that renders the system (2.40)-(2.41) uniformly exponentially stable is found to satisfy the robust QP specification for $w_p \longrightarrow z_p$ which is nonsingular index P_p. For guaranteeing robust stability and performance of the closed-loop system, extended multipliers are employed and adjusted to the extended uncertainty structure that is given as

$$P_e = \begin{pmatrix} Q_e & S_e \\ S_e^T & R_e \end{pmatrix} = \left(\begin{array}{cc|cc} Q & Q_{12} & S & S_{12} \\ Q_{21} & Q_{22} & S_{21} & S_{22} \\ \hline S^T & S_{21}^T & R & R_{12} \\ S_{12}^T & S_{22}^T & R_{21} & R_{22} \end{array}\right) \quad \text{with} \quad Q_e < 0, \quad R_e > 0, \tag{2.42}$$

and that satisfies

$$\begin{pmatrix} \Delta & 0 \\ 0 & f_c(\Delta) \\ I & 0 \\ 0 & I \end{pmatrix}^T P_e \begin{pmatrix} \Delta & 0 \\ 0 & f_c(\Delta) \\ I & 0 \\ 0 & I \end{pmatrix} > 0, \quad \forall \Delta \in \mathbf{\Delta} \tag{2.43}$$

The corresponding dual multipliers $\widetilde{P}_e = P_e^{-1}$ are partitioned similar to

$$\widetilde{P}_e = \begin{pmatrix} \widetilde{Q}_e & \widetilde{S}_e \\ \widetilde{S}_e^T & \widetilde{R}_e \end{pmatrix} = \left(\begin{array}{cc|cc} \widetilde{Q} & \widetilde{Q}_{12} & \widetilde{S} & \widetilde{S}_{12} \\ \widetilde{Q}_{21} & \widetilde{Q}_{22} & \widetilde{S}_{21} & \widetilde{S}_{22} \\ \hline \widetilde{S}^T & \widetilde{S}_{21}^T & \widetilde{R} & \widetilde{R}_{12} \\ \widetilde{S}_{12}^T & \widetilde{S}_{22}^T & \widetilde{R}_{21} & \widetilde{R}_{22} \end{array}\right) \quad \text{with} \quad \widetilde{Q}_e < 0, \quad \widetilde{R}_e > 0, \tag{2.44}$$

There exists a controller (2.37) anda scheduling function so that the system (2.40)-(2.41) controlled with (2.37) aND the analysis conditions for robust QP with multipliers (2.42)-(2.43) are satisfied if only if there exists X, Y and scalings $P \in \mathcal{P}_1$, $\widetilde{P} \in \widetilde{\mathcal{P}}_1$ that satisfy the LMIs

$$\begin{pmatrix} Y & I \\ I & X \end{pmatrix} > 0, \tag{2.45}$$

$$\begin{pmatrix} \star \\ \star \\ \hline \star \\ \star \\ \hline \star \\ \star \end{pmatrix}^T \left(\begin{array}{cc|cc|cc} X & 0 & 0 & 0 & 0 & 0 \\ 0 & X & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & Q & S & 0 & 0 \\ 0 & 0 & S^T & R & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & -\gamma I & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{\gamma} I \end{array}\right) \left(\begin{array}{ccc} I & 0 & 0 \\ A & B_\delta & B_p \\ \hline 0 & I & 0 \\ C_\delta & D_{\delta\delta} & D_{\delta p} \\ \hline 0 & 0 & I \\ C_p & D_{p\delta} & D_{pp} \end{array}\right) \begin{pmatrix} \Psi_1 \\ \Psi_2 \\ \Psi_3 \end{pmatrix} < 0 \tag{2.46}$$

$$\begin{pmatrix} \star \\ \star \\ \hline \star \\ \star \\ \hline \star \\ \star \end{pmatrix}^T \left(\begin{array}{cc|cc|cc} Y & 0 & 0 & 0 & 0 & 0 \\ 0 & Y & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & \widetilde{Q} & \widetilde{S} & 0 & 0 \\ 0 & 0 & \widetilde{S}^T & \widetilde{R} & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & -\frac{1}{\gamma} I & 0 \\ 0 & 0 & 0 & 0 & 0 & \gamma I \end{array}\right) \left(\begin{array}{ccc} -A^T & -C_\delta^T & -C_p^T \\ I & 0 & 0 \\ \hline -B_\delta^T & -D_{\delta\delta}^T & -D_{p\delta}^T \\ 0 & 0 & I \\ \hline -B_p^T & -D_{\delta p}^T & -D_{pp}^T \\ 0 & 0 & I \end{array}\right) \begin{pmatrix} \Phi_1 \\ \Phi_2 \\ \Phi_3 \end{pmatrix} > 0 \tag{2.47}$$

where

$$\begin{pmatrix} I & 0 & 0 & 0 & 0 \\ 0 & C_y & D_{y\delta} & 0 & D_{yp} \\ 0 & 0 & 0 & I & 0 \end{pmatrix}_\perp = \begin{pmatrix} 0 \\ \Psi_1 \\ \Psi_2 \\ 0 \\ \Psi_3 \end{pmatrix}, \quad \begin{pmatrix} 0 & I & 0 & 0 & 0 \\ B_u^T & 0 & D_{\delta u}^T & 0 & D_{pu} \\ 0 & 0 & 0 & I & 0 \end{pmatrix}_\perp = \begin{pmatrix} \Phi_1 \\ 0 \\ \Phi_2 \\ 0 \\ \Phi_3 \end{pmatrix} \tag{2.48}$$

On the condition that a solution to (2.45)-(2.47) is found:.

1. **First step:** Extension of scalings. The matrices are defined as

   $$Z = \begin{pmatrix} I \\ 0 \end{pmatrix} \text{ and } \widetilde{Z} = \begin{pmatrix} 0 \\ I \end{pmatrix}$$

   with the row partition of P. For the given P and $\widetilde{P}$, an extension P_e with (2.42) are found such that the dual multiplier $\widetilde{P}_e = P_e^{-1}$ is related to the given $\widetilde{P}$ as in (2.44). After a suitable permutation, this amounts to find an extension

   $$\begin{pmatrix} P & T \\ T^T & T^TNT \end{pmatrix} \text{ with } \begin{pmatrix} \widetilde{P} & T \\ T^T & T^TNT \end{pmatrix} = \begin{pmatrix} P & T \\ T^T & T^TNT \end{pmatrix}^{-1} \tag{2.49}$$

where the specific parametrization of the new blocks in terms of a nonsingular matrix T and some symmetric N will turn out to be convenient. The positivity/negativity constraints in (2.42) is obeyed that amount to

$$\begin{pmatrix} Z & 0 \\ 0 & Z \end{pmatrix}^T \begin{pmatrix} P & T \\ T^T & T^TNT \end{pmatrix} \begin{pmatrix} Z & 0 \\ 0 & Z \end{pmatrix} < 0 \tag{2.50}$$

and

$$\begin{pmatrix} \widetilde{Z} & 0 \\ 0 & \widetilde{Z} \end{pmatrix}^T \begin{pmatrix} P & T \\ T^T & T^TNT \end{pmatrix} \begin{pmatrix} \widetilde{Z} & 0 \\ 0 & \widetilde{Z} \end{pmatrix} < 0 \tag{2.51}$$

The main goal is to adjust T to render (2.50)-(2.51) and it is satisfied.The sub-blocks $T_1 = TZ$ and $T_2 = T\widetilde{Z}$ of $T = (T_1, T_2)$ are conduct and this lead to

$$\begin{aligned} T_1^T[N - Z(Z^TPZ)^{-1}Z^T]T_1 < 0 \\ T_2^T[N - \widetilde{Z}(\widetilde{Z}^TP\widetilde{Z})^{-1}\widetilde{Z}^T]T_2 > 0 \end{aligned} \tag{2.52}$$

If $n_+(S)$ and $n_-(S)$, are denoted respectively, the number of positive and negative eigenvalues of the symmetric matrix S are achieved.
Simple Scur complement arguments reveal that

$$\begin{aligned} n_-\begin{pmatrix} Z^TPZ & Z^T \\ Z & N \end{pmatrix} &= n_-(N - Z(Z^TPZ)^{-1}Z^T) + n_-(Z^TPZ) \\ &= n_-(Z(Z^T\widetilde{P}^{-1}Z)Z^T) + n_-(N) \end{aligned} \tag{2.53}$$

When $N = (P - \widetilde{P}^{-1})^{-1}$ is set. This leads to

$$n_-(N - Z(Z^TPZ)^{-1}Z^T) = n_-(N) \tag{2.54}$$

and

$$n_+(N - \widetilde{Z}(\widetilde{Z}^TP\widetilde{Z})^{-1}\widetilde{Z}^T) = n_+(N) \tag{2.55}$$

This implies that there exists T_1 and T_2 with $n_-(N)$, $n_+(N)$ columns that satisfy (2.52)-(2.53).

2. **Second step:** Construction of the scheduling function. Recall that

$$\begin{pmatrix} \Delta \\ I \end{pmatrix} P \begin{pmatrix} \Delta \\ I \end{pmatrix} > 0 \text{ and } \begin{pmatrix} I \\ -\Delta^T \end{pmatrix}^T \widetilde{P} \begin{pmatrix} I \\ -\Delta^T \end{pmatrix} < 0, \quad \forall \Delta \in \mathbf{\Delta}$$

Due to the structural simplicity, an explicit formula can be provided. This shows that $f_c(\Delta)$ can be selected to depend smoothly on $\mathbf{\Delta}$. Indeed, by a straightforward Schur-complement argument, (2.43) is equivalent to

$$\left(\begin{array}{cc|cc} U_{11} & U_{12} & (W_{11}+\Delta)^T & W_{21}^T \\ U_{21} & U_{22} & W_{12}^T & (W_{22}+f_c(\Delta))^T \\ \hline W_{11}+\Delta & W_{12} & V_{11} & V_{12} \\ W_{21} & W_{22}+f_c(\Delta) & V_{21} & V_{22} \end{array}\right) > 0 \tag{2.56}$$

for $U = R_e - S_e^T Q_e^{-1} S_e > 0, \quad V = -Q_e^{-1} > 0, \quad W = Q_e^{-1} S_e$.
Since there exists a solution, the inequality can be arranged to

$$\begin{pmatrix} U_{22} & (W_{22} + f_c(\Delta))^T \\ W_{22} + f_c(\Delta) & V_{22} \end{pmatrix} \\ - \begin{pmatrix} U_{21} & W_{12}^T \\ W_{21} & V_{21} \end{pmatrix} \begin{pmatrix} U_{11} & (W_{11} + \Delta)^T \\ W_{11} + \Delta & V_{11} \end{pmatrix}^{-1} \begin{pmatrix} U_{12} & W_{21}^T \\ W_{12} & V_{12} \end{pmatrix} > 0 \tag{2.57}$$

Finally, The scheduling function is obtained as:

$$f_c(\Delta) = -W_{22} + \begin{pmatrix} W_{21} & V_{21} \end{pmatrix} \begin{pmatrix} U_{11} & (W_{11} + \Delta)^T \\ W_{11} + \Delta & V_{11} \end{pmatrix}^{-1} \begin{pmatrix} U_{12} \\ W_{12} \end{pmatrix}. \tag{2.58}$$

3. **Third step** LTI controller construction. After having reconstructed the scalings, the LTI part of the controller is designed to solve a nominal QP problem that can be done along standard lines.

2.4 Conclusions

So far, we have introduced the methods that we will use for controller design of an MIP. The first controller is linear state-feedback controller LQR and pole placement. And the second controller is an $\mathcal{H}_\infty$ controller which is designed to use an LPV control technique with a Full Block multiplier. For all controllers mentioned above will be designed for balancing and rotating of the MIP.