

# Automatic Control Techniques Review: Part 1

Withit Chatlatanagulchai

Control of Robot and Vibration Laboratory (CRVLAB),  
Department of Mechanical Engineering, Faculty of Engineering,

Kasetsart University, Bangkok, Thailand

E-mail: fengwtc@ku.ac.th

## ABSTRACT

Existing control techniques in the engineering literature are diverse. This paper attempts to classify control techniques of which the author is aware from his literature review and experience with industrial control projects. Due to the vast number of techniques, most of the presentations will be brief; however, the reader will be pointed toward further excellent references. This paper should act as a starting point for readers, who may or may not have already become familiar with control theory, but are eager to see an overview of the control techniques, to choose techniques that suit their needs and to study them deeper. The work is divided into Part 1 and Part 2. In this first part, techniques that will be discussed are those of basic control, adaptive control, and robust control. In Part 2, techniques that will be discussed are those of nonlinear control, optimal control, and control supplements. The reader who is interested in the field of control and would like to know further details, is referred to an excellent control handbook and references therein.

## Keywords:

control techniques, basic control, adaptive control, robust control

## 1. BASIC CONTROL

This class of control design techniques consists of some basic algorithms. Usually, no robustness is explicitly accounted for during the design, and there is no special structure that can adapt with system change or can act fast to attenuate the disturbances.

### 1.1 Proportional Integral Derivative (PID) Control

PID is the XE "PID" most well-known controller because it does not require mathematical modeling in its design. The user adjusts PID gains by trial and error. It is very convenient and is the first controller to be tried out but usually has marginal performance.



Figure 1 shows a typical implementation of PID in a closed-loop system. In the time domain, PID has the form

$$u = K_p e + K_i \int_0^t e dt + K_d \frac{de}{dt}, \quad (1)$$

where  $k_p$ ,  $k_i$ , and  $k_d$  are proportional, integral, and derivative gains, respectively. In the frequency domain, PID has the form "PID:Basic formula" the form

$$U(s) = \frac{K_d s^2 + K_p s + K_i}{s} E(s), \quad (2)$$

where  $s$  is the Laplace transform variable.

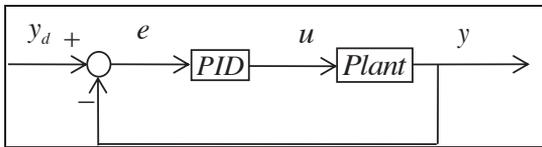


Figure 1: PID Control

There are two other variations (2) mostly seen in practice. The first is the ideal or parallel form

$$U(s) = \left[ K_c \left( 1 + \frac{1}{\tau_I s} + \tau_D s \right) \right] E(s). \quad (3)$$

The second is the cascade form

$$U(s) = \left[ \tilde{K}_c \frac{(\tilde{\tau}_I s + 1)(\tilde{\tau}_D s + 1)}{\tilde{\tau}_I s} \right] E(s). \quad (4)$$

There are two considerations to bear in mind in practice. First, it can be seen that the PID controllers (2) - (4) are in fact improper (the order of the numerator is greater than the order of the denominator), which may lead to stability problems. In practice, one needs to multiply the PID controller with a filter. The filter is typically of the form  $1/(\tau_D s + 1)$  with  $\tau_D$  about 0.1. In most cases, the filter does not change the closed-loop response noticeably. Second, to avoid derivative kick due to the differentiation

of a step reference input, the reference input is usually not differentiated, resulting in a practical two-degrees-of-freedom implementation of the PID controller

$$U(s) = K_c \left[ \left( 1 + \frac{1}{\tau_I s} \right) (R(s) - Y(s)) - \frac{\tau_D s}{\varepsilon \tau_D s + 1} Y(s) \right], \quad (5)$$

where  $R(s)$  and  $Y(s)$  are the Laplace transformations of the reference input  $r(t)$  and the measured output  $y(t)$

There exist some workable PID tuning rules, which are based on the plant model or the required response. PID tuning rules [2] which are claimed by their authors to be "the best simple PID tuning rules in the world" are as follows. For a plant with a first-order plus time delay model

$$G(s) = \frac{k}{\tau s + 1} e^{-\theta s},$$

the PID gains are given by

$$K_c = \frac{1}{k} \frac{\tau}{\tau_c + \theta}, \quad \tau_c = \min(\tau, 4(\tau_c + \theta)),$$

where  $\tau_c$  is the only tuning parameter and is recommended to set at  $k_c = \theta$ . Note that there is no derivative term ( $\tau_D = 0$ ) since it is uncommon to use derivative in process control applications, where most plants are stable with simple overdamped responses. The performance improvement of adding a derivative is usually too small compared to the risk of increased sensitivity to high-frequency measurement noise. For a plant with dominant second-order model,

$$G(s) = k \frac{e^{-\theta s}}{(\tau_1 s + 1)(\tau_2 s + 1)},$$

where  $\tau_2 > \theta$ , the recommended tuning rule is



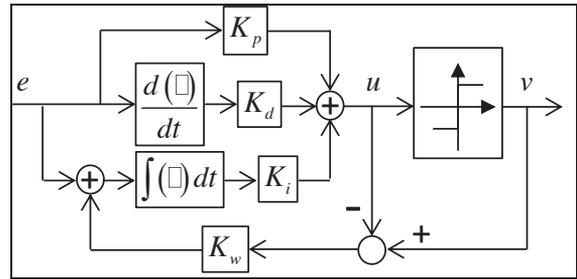
$$\tilde{K}_C = \frac{1}{k} \frac{\tau_1}{\tau_c + \theta}, \tilde{\tau}_I = \min(\tau_1, 4(\tau_c + \theta)),$$

$$\tilde{\tau}_D = \tau_2.$$

When implemented, the controller (1) can simply use some numerical algorithms for the integral of the error and the derivative of the error and the controller can be directly written as Matlab or Labview codes without complications. For the controllers (2)- (5), which are in s-domain transfer functions, some conversions to z-domain transfer functions must first be performed.

Another implementation issue often encountered in practice is the so-called integral wind-up problem. In practice, all actuators have saturation limits. The integral wind-up occurs when the actuator reaches its saturation limit and cannot move further; however, in the control algorithm, the tracking error keeps accumulating in the integral term. Once the actuator moves back to within its saturation limit, the control algorithm needs a long time to adjust its output because of the already-big integral term, resulting in poor transient control performance.

Although the integral wind-up problem cannot be completely avoided, several methods have been proposed to reduce its effect. One of these is called a tracking method, as shown in Figure 2. The tracking method basically adds another integral term with the gain  $K_w$ ,  $K_i$  for the control algorithm output  $u$  to track the actuator output  $v$ . Since  $v$  is the saturated version of  $u$ , this method helps prevent the algorithm output  $u$  from increasing beyond the saturation limit due to the integral term.



**Figure 2:** Tracking Method to Avoid Integral Wind-up

In-depth treatment of the PID controller is given in [3] and [4].

### 1.2 Inverse Dynamics

The Inverse dynamics control is typically implemented as shown in Figure 3. Let  $P$  be the actual plant and  $\hat{P}$  be the plant model. The plant model  $\hat{P}$  maps the control input  $\hat{u}$  to the output  $\hat{y}$ . Suppose this plant model  $\hat{P}$  can be inverted. That is, an inverse model  $\hat{P}^{-1}$  can be found mapping from  $\hat{y}$  to  $\hat{u}$ . If the plant model  $\hat{P}$  represents the actual plant precisely, that is,  $\hat{P}=P$ , by letting the input to  $\hat{P}^{-1}$  be  $y_d$ , we would get  $y=y_d$ , since  $\hat{P}^{-1}P=1$ . The output will track its desired value perfectly without the need of an additional controller  $G$ . However, since we already know that  $\hat{P}$  will never equal  $P$ , hence the term plant uncertainty,  $\hat{P}^{-1}$  will never cancel out  $P$  perfectly, and the controller  $G$  is needed for additional control effort. If the plant model  $\hat{P}$  has good accuracy,  $\hat{P}^{-1}$  will cancel out most of  $P$ , and the control effort from  $G$  would be small. Performance of the inverse dynamics control heavily depends on the accuracy of the plant model. Besides, complicated plant models usually cannot be inverted simply because their inverses do not exist or become non-singular.

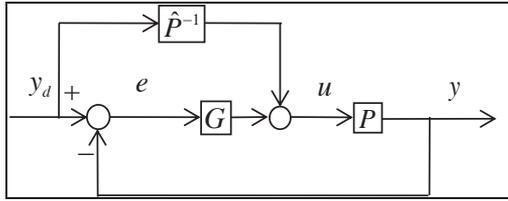


Figure 3: Inverse Dynamics Control

### 1.3 Pole Placement Using State Feedback

Pole placement techniques work by placing poles of the resulting closed-loop system at desired locations.

Consider a linear state-space model

$$\begin{aligned} \dot{x} &= Ax + Bu, \\ y &= Cx + Du. \end{aligned}$$

Normally, the output  $y$  has no connection with the control input  $u$ . Therefore,  $D=0$ . For a simple case, suppose  $y=x_1$ , where  $x$  is a state vector of dimension  $n$ , that is,  $x=[x_1, x_2, \dots, x_n]^T$ . We want to design a gain vector  $K=[K_1, K_2]$  so that a state-feedback control law  $u=-K_x x + K_1 r$  would drive the output  $y$  to the reference input  $r$ . Figure 4 shows a block diagram of this pole placement using state-feedback control.

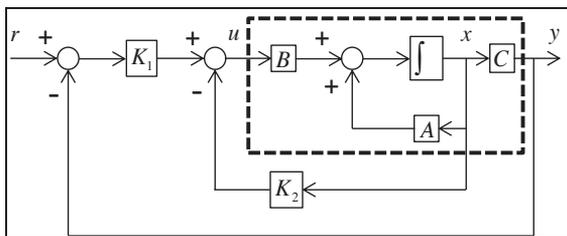


Figure 4: Pole Placement Using State-feedback Control

The controller  $u = -Kx + K_1 r$  would result in a closed-loop system

$$\dot{x} = (A - BK)x + BK_1 r. \quad (6)$$

At steady state, we have

$$\dot{x}(\infty) = (A - BK)x(\infty) + BK_1 r(\infty). \quad (7)$$

Subtracting (7) from (6), letting  $e = x - x(\infty)$  and using the fact that  $r(t) = r(\infty)$  for a step reference input, we then have

$$\dot{e} = (A - BK)e.$$

Therefore, we can choose  $K$  to place the poles of the equation above for EMBED Equation. DSMT4 to approach zero.

### 1.4 Pole Placement Using Root Locus

Consider XE "Pole placement:Root locus" a block diagram in Figure 5. The closed-loop transfer function from  $r$  to  $y$  is

$$\frac{Y(s)}{Y_d(s)} = \frac{G(s)P(s)}{1 + G(s)P(s)}.$$

The closed-loop characteristic equation (CLCE) is

$$1 + G(s)P(s) = 0,$$

whose solutions are the closed-loop poles.

The CLCE above can be rewritten as

$$1 + K \frac{N(s)}{D(s)} = 0$$

The root locus plot is just the plot of this CLCE when  $K$  varies from 0 to  $\infty$ . In other words, the root locus plot is the closed-loop poles' locations when  $K$  varies from 0 to  $\infty$ . Pole placement using root locus can be done using the following steps. The first step is obtaining CLCE (with  $K$ ). The second step is obtaining CLCE from desired locations of closed-loop poles. The third step is matching both CLCEs and determining  $K$ .

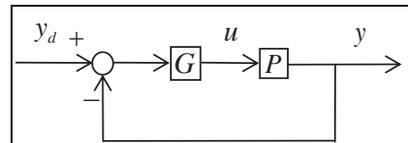


Figure 5: A Simple Unity-feedback Closed-loop System

Details of the pole placement techniques can be found in any basic control textbooks. For example, we have used [5],[6],and [8].



### 1.5 Internal Model Control (IMC)

The idea of IMC is to design a controller so that the resulting closed-loop system matches a desired closed-loop system. This idea is simple and has been proven to be successful in chemical-process applications that involve time delays or a non-minimum-phase plant.

Time delay results in a term  $e^{-\theta s}$ ,  $\theta > 0$  in the plant model, and the non-minimum phase takes place when the plant model has unstable (right-half plane) zero. The time delay and non-minimum phase are essentially the same, since a first-order Taylor approximation gives  $e^{-\theta s} = 1 - \theta s$ , which appears as a right-half-plane zero.

It is a well-known fact that the right-half-plane zero cannot be inverted (since it will become an unstable pole) and cannot be cancelled by the controller (since plant uncertainty will result in imperfect cancellation, which may lead to internal instability – the instability of other internal states that may not be seen from the measured output.)

Consider a plant model, factorized into an invertible minimum-phase part ( $G_m$ ) and a non-invertible all-pass part ( $G_a$ )

$$G(s) = G_m G_a,$$

$$G_a(s) = e^{-\theta s} \prod_i \frac{-s + z_i}{s + z_i}, \text{Re}(z_i) > 0; \theta > 0.$$

For example, a non-minimum-phase transfer function with time delay can be factorized as follows:

$$\begin{aligned} G(s) &= \frac{e^{-3s}(s-1)}{(s+2)(s+3)} \\ &= \frac{(s+1)}{(s+2)(s+3)} \cdot \frac{e^{-3s}(-s+1)}{(s+1)}. \end{aligned}$$

The next step is to specify the desired closed-loop transfer function  $T$  from reference input to output,  $y = Tr$ . Since we cannot prevent  $T$  from including the non-minimum-phase part, we have the desired closed-loop transfer function

$$T(s) = f(s)G_a(s), \quad (8)$$

where  $f(s)$  is a low-pass filter selected by the designer, typically of the form  $f(s) = 1/(\tau_c s + 1)^n$ .

The closed-loop transfer function with a controller  $K$  and a plant model  $G$  is given by

$$T = GK(1 + GK)^{-1}. \quad (9)$$

By equating (9) to (8), we have the controller

$$K = G^{-1} \frac{T}{1 - T} = G_m^{-1} \frac{1}{f^{-1} - G_a}.$$

Note that the controller only contains the inverse of the minimum-phase part of the plant model.

This version of the IMC is obtained from [2]

### 1.6 Decentralized Control

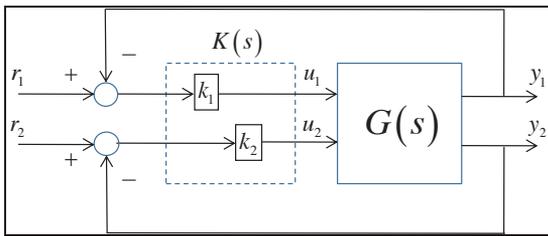
Decentralized control is, in fact, a general name used to call any control schemes that use diagonal controllers to control a square multi-input-multi-output (MIMO) plant. Square means the number of inputs equals the number of outputs. As an example, a 2-input-2-output plant, given as

$$G(s) = \begin{bmatrix} g_{11}(s) & g_{12}(s) \\ g_{21}(s) & g_{22}(s) \end{bmatrix},$$

is controlled by a diagonal controller

$$K(s) = \begin{bmatrix} k_1(s) & 0 \\ 0 & k_2(s) \end{bmatrix}.$$

Figure 6 depicts the decentralized control for the 2-input-2-output plant above.



**Figure 6:** Decentralized Control of a 2-input-2-output Plant

One obvious challenge of a controller of this type is that a smaller number of controllers are used to control a larger number of plants. For example, only two controllers,  $k_1$  and  $k_2$ , are used to control four plants,  $g_{11}$ ,  $g_{12}$ ,  $g_{21}$ , and  $g_{22}$ . As can be expected, the method relies on the ability to decouple the plant (to make  $G(s)$  close to diagonal) or to decouple outputs (to design each controller sequentially.)

There are three main approaches in the decentralized control:

- Fully coordinated design

All the diagonal controller elements  $k_i(s)$  are designed simultaneously, based on the plant  $G(s)$  using some optimization approaches. However, this approach reduces the benefits of decentralized control, which are ease of tuning, reduced modeling effort, and good failure tolerance, and suffers from the complexity of the optimization algorithm used. It is generally viewed that its advantage might not surpass that of the “full” multivariable controller. Therefore, this approach is not normally used in practice.

- Independent design

Each controller element  $k_i(s)$  is designed based on the corresponding diagonal element of  $G(s)$ . There are some considerations of the

off-diagonal interactions when tuning each loop. Therefore, input-to-output pairings are important to reduce these interactions.

- Sequential design

The controllers are designed sequentially, one at a time. Once one controller is implemented, the next one is designed based on the closed-loop system of the previous one. The sequential design approach can be used for interactive problems where the independent design approach fails, provided that it is acceptable to decouple the outputs to have slow and fast outputs in separated loops.

This version of the decentralized control is obtained from [2]

## 2. Adaptive control

Usually the engineer knows the model structure of the plant to be controlled, but the model's parameters may be difficult to obtain precisely. For example, most chemical processes can be modeled as a first-order transfer function with time delay

$$\frac{P}{v} = \frac{a}{bs+1} e^{-cs},$$

where  $a$ ,  $b$  and  $c$  are unknown parameters.

Adaptive control incorporates an on-line learning algorithm to determine these unknown parameters while satisfying other control objectives such as tracking, stability, and robustness. The adaptive control is also suitable for controlling systems that may change over time or have different operating points, such as those in engines.



Figure 7 depicts a general adaptive control system where the plant parameters are unknown. Available signals, which are the control input  $u$  and the measured output  $y$ , are used in a strategy to adjust controller parameters so that a desired closed-loop performance is achieved.

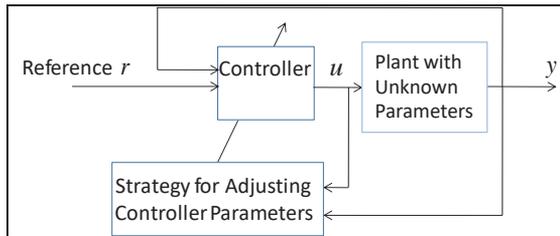


Figure 7: General Adaptive Control System

There are three main approaches in adaptive control: gain scheduling control, adaptive pole placement control, and model reference adaptive control.

### 2.1 Gain Scheduling

In gain scheduling control, the controller parameters are adjusted based on some auxiliary measurements. The adjustment can be either discontinuous or smooth. For example, several PID gains can be adjusted based on load and engine speed operating points. Figure 8 shows a gain scheduling diagram.

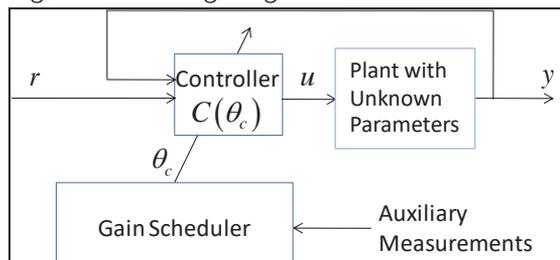


Figure 8 : Gain Scheduling

### 2.2 Adaptive Pole Placement Control (APPC)

In adaptive pole placement control,

the controller is designed to place the closed-loop poles at some fixed desired locations. The controller parameters, however, adapt with the change in the system. There are two categories: indirect and direct adaptive pole placement control. The indirect adaptive control estimates the plant parameters, whereas the direct adaptive control estimates the controller parameters.

Figure 9 contains a diagram of the indirect adaptive pole placement control. Available signals, which are the control input  $u$ , the measured output  $y$ , and the reference input  $r$ , are used in estimating the plant parameters  $\theta^*$  online. Once the estimate  $\theta$  is obtained, it is used to form the controller parameters  $\theta_c$ , which will be used in the control law.

Figure 10 contains a diagram of the direct adaptive pole placement control, where the controller parameters are estimated directly from available signals.

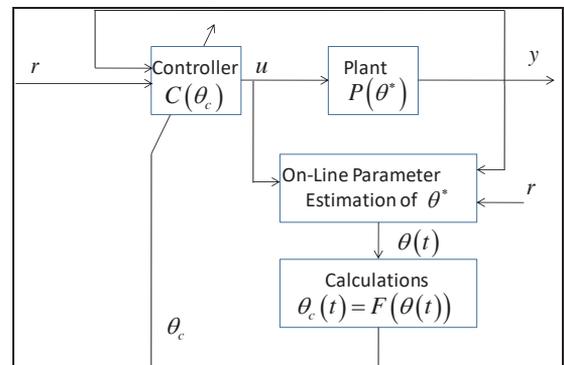


Figure 9: Indirect Adaptive Pole Placement Control

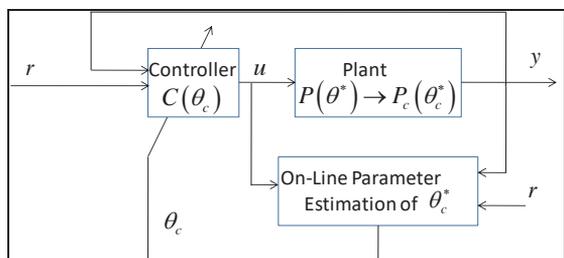


Figure 10: Direct Adaptive Pole Placement Control



### 2.3 Model Reference Adaptive Control (MRAC)

Model reference adaptive control is, in fact, similar to the adaptive pole placement control, but instead of placing the closed-loop poles at desired locations, the MRAC tries to follow a desired reference model by minimizing the difference between the model output and the measured output. Similar to the APPC, the MRAC can be divided into direct and indirect control. Figure 11 depicts a diagram of the indirect MRAC, whereas Figure 12 depicts a diagram of the direct MRAC.

Traditional adaptive control is designed with three basic assumptions. First, the system is free from noise and disturbances. Second, the plant structure represents the actual system precisely. Third, the unknown parameters are constant at all times.

In practice, there are noise and disturbances, the actual system may not possess the plant model structure leading to plant model uncertainties, and the plant parameters may vary with time. The traditional adaptive control system must be modified to avoid parameter drift and instability problems, by creating a scheme called robust adaptive control, where adaptive laws are modified to provide robustness for the closed-loop system.

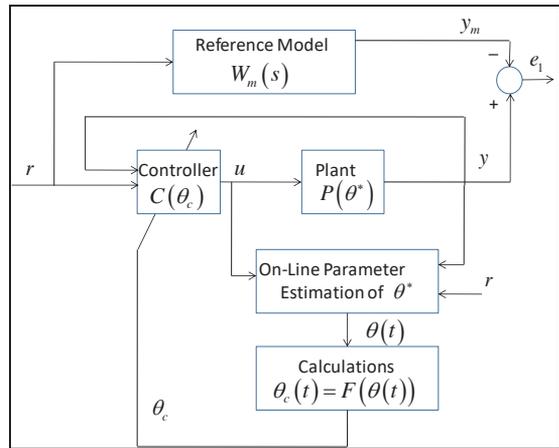


Figure 11: Indirect Model Reference Adaptive Control

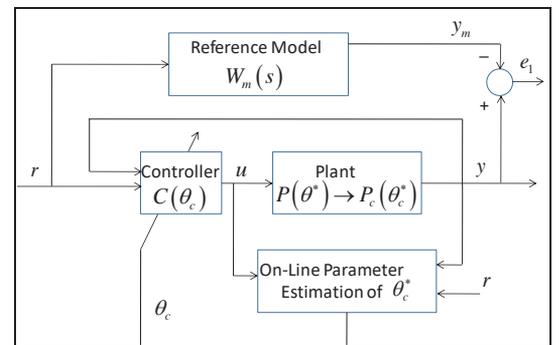


Figure 12: Direct Model Reference Adaptive Control

Details of the adaptive control as well as the robust adaptive control can be found in many excellent textbooks. For example, we have used [9],[10],[11],and [12].

### 2.4 Intelligent Control

In the adaptive control schemes previously mentioned, we must know the structure of the unknown plant model. The adaptive law only estimates the unknown parameters, not the structure of the plant model. Currently, more research has turned toward using intelligent systems to completely replace the unknown plant model. Executing it this way, the plant model is totally eliminated from the control algorithm.



The intelligent systems are obtained by mimicking the human power of reasoning and logic. The mathematical functions, forming the intelligent system, were proved to be a universal approximator that can estimate any continuously differentiable functions with arbitrary accuracy. The intelligent control schemes use the approximation properties of the intelligent system to estimate the unknown plant model. Then, the control system is designed from this estimated plant.

The estimation process of any intelligent system can be depicted as shown in Figure 13. Suppose a function to be estimated is  $f(x_1, x_2)$ , a scalar-valued function of  $x_1$  and  $x_2$ . The inputs  $x_1$  and  $x_2$  are separated into different groups. Each group has its own attribute. Subsequently, there is a mapping from each group to the estimated surface of  $f(x_1, x_2)$ . The system's adjustable parameters can be either group parameters or mapping parameters or both. These parameters are adjusted to minimize the estimation error.

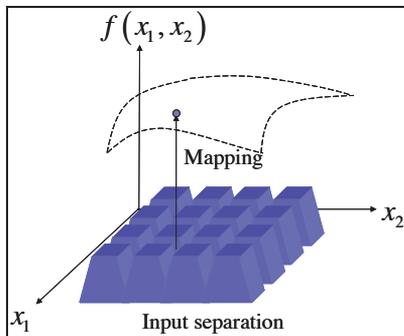


Figure 13: Input Separation and Mapping Processes of the Intelligent System in Estimating a Scalar-valued Function

There are many types of intelligent systems, though their basic estimation processes are all the same. Some intelligent systems possess more advantages over the others in some aspects, such as simpler structure or more estimation

power. The most well-known intelligent systems are neural networks and fuzzy logic.

- Neural networks

Neural networks obtained the idea from mimicking human neurons. An example of a simple neural network is a radial basis function network (RBFN), depicted in Figure 14. A scalar-valued function to be estimated is  $g(z_1, z_2, \dots, z_n)$ . The inputs to the networks are therefore  $z_1$  to  $z_n$ .  $s_i(Z)$  is the Gaussian function of the input vector  $Z$ , given as follows:

$$s_i(Z) = \exp\left[-\frac{\|Z - \mu_i\|^2}{\sigma_i^2}\right]$$

$$= \exp\left[-\frac{(Z - \mu_i)^T (Z - \mu_i)}{\sigma_i^2}\right], i = 1, 2, \dots, l,$$

where  $\mu_i = [\mu_{i1}, \mu_{i2}, \dots, \mu_{in}]^T$  is the center of the receptive field and  $\sigma_i$  is the width of the Gaussian function.

The parameters  $\mu_i$  and  $\sigma_i$ , together with the weights  $w_i$ , are to be adapted in ways that the output of the networks  $\hat{g}(z_1, z_2, \dots, z_n)$  estimates the actual function  $g(z_1, z_2, \dots, z_n)$  closely. In fact, we can view determining  $\mu_i$  and  $\sigma_i$  as the input separation process and determining  $w_i$  as the mapping process.

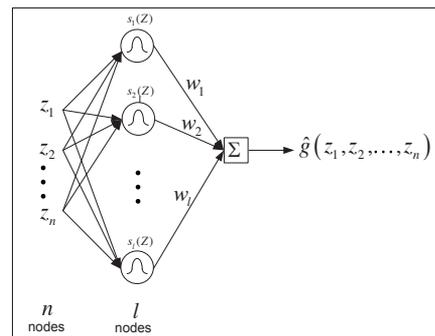
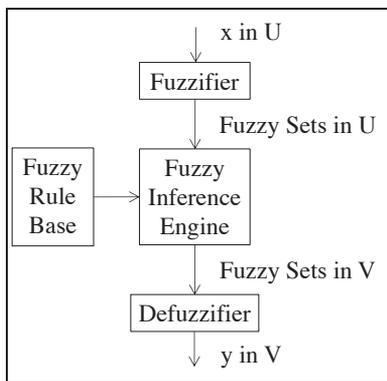


Figure 14: Diagram of the Radial Basis Function Network

- Fuzzy logic

Fuzzy logic uses a fuzzy system as an approximator. Instead of using mathematical functions, the fuzzy system uses human logic in input separation and mapping. Figure 15 depicts a basic configuration of a fuzzy system, where the fuzzifier can be viewed as input separation and fuzzy inference engine and the defuzzifier can be viewed as mapping.



**Figure 15:** Basic Configuration of a Fuzzy System

A fuzzy system can be used directly as a controller or as a plant estimator on which a controller is designed. A fuzzy system can also be used in the adaptive control setting as an estimator of the controller's parameters in the direct method, or as an estimator of the plant parameters in the indirect method.

The intelligent control using neural networks and fuzzy logic can be found in several excellent text books. We have used [13] and [14] Textbooks dedicated to neural networks are [15], [16] and [17]. Textbooks specifically for fuzzy logic control are [18] and [19].

### 3. ROBUST CONTROL

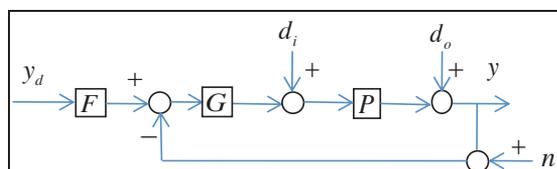
A control system is robust when it can tolerate external disturbances and noise and model uncertainties without becoming unstable or degrading performance. As mentioned before, all controllers are robust. However, in most controllers, we simply do not know how robust the controllers are or whether the robustness can be guaranteed until some simulations or experiments are performed.

In this robust control category, we present various controllers that guarantee both robust performance and stability, because there are some robust specifications taken into account during the design process or possess some structures that can counteract the effects of the disturbances, noise, and uncertainty.

#### 3.1 Quantitative Feedback Theory (QFT)

QFT was developed almost sixty years ago. It is a controller design process that results in a frequency-domain controller.

Figure 16 shows a typical closed-loop system, where  $F$  and  $G$  are the prefilter and the controller designed using QFT.  $P$  is the plant,  $y$  is the output, and  $y_d$  is its desired value.  $d_i$ ,  $d_o$ , and  $n$  are the plant input disturbance, the plant output disturbance, and the measurement noise, respectively.



**Figure 16:** A Two-DOF Closed-loop System



A QFT-based design process has three steps. These are developing frequency-domain specifications, generating bounds, and performing loop shaping.

Specifications may be given in time or frequency domain. If it is given in the time domain, it must be converted to the frequency domain. Specifications are tracking, disturbance rejection, noise rejection, stability margin, and control effort. For example, in tracking, we would want  $|FGP/(1+GP)|$  to be close to one to follow a constant reference command. In plant output disturbance rejection, we would want  $|1/(1+GP)|$  to be less than a small number because we want to attenuate the effect of the disturbance to the output.

Frequency-domain specifications are converted to bounds on the Nichols chart. We all are familiar with the bode gain and phase plots. The Nichols chart has the bode gain as its vertical axis and the bode phase as its horizontal axis.

For a frequency, instead of a one-point plant, the plant can include uncertainty and becomes a region called plant template. Figure 17 shows a one-point plant versus a plant template.

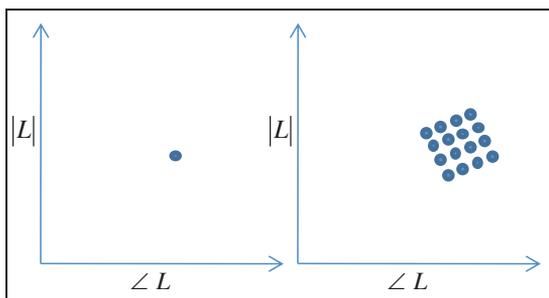


Figure 17: One-point Plant versus Plant Template on the Nichols Chart

The bounds are generated from the frequency-domain specifications by making sure that all points on the plant template are included when generating a bound.

The worst-case bounds are the strictest bounds among all specifications at a given frequency. There is only one worst-case bound per frequency.

The loop shaping involves designing the controller  $G$  and the prefilter  $F$  so that the resulting open-loop shape satisfies all bounds at all frequencies. Figure 18 is an example of worst-case bounds and the open-loop shape.

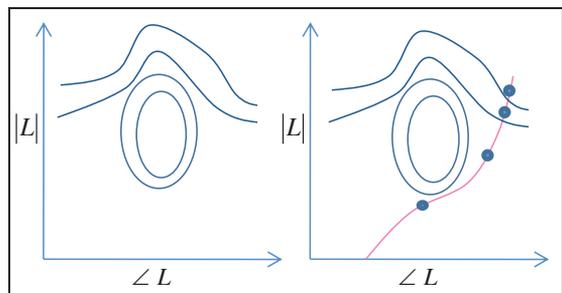


Figure 18: Worst-case Bounds and the Loop Shaping

In summary, a QFT-based controller is robust against plant uncertainty since it is designed from the plant template. The controller is also robust against external disturbances and noise since they are specified in the specifications. The tracking performance and control effort can also be specified. More details of the QFT can be found in [20],[21],[22],and [23]

### 3.2 $H_2$ and $H_\infty$ Control

$H_2$  and  $H_\infty$  controls have gained increasing popularity due to their applicability to the multi-input-multi-output (MIMO) plant and the availability of the optimization algorithms



and software to implement them. The control formulation is based on a general configuration, given in Figure 19, where  $P$  is the plant model;  $K$  is the controller;  $\Delta$  represents uncertainty;  $w$  is exogenous inputs such as reference input, disturbances, and noise;  $z$  is the error signal to be minimized;  $v$  is generalized controller input such as reference input, measured plant output, measured disturbances, or error signal;  $u$  is the control effort;  $y_\Delta$  and  $u_\Delta$  are input and output of the uncertainty, respectively.

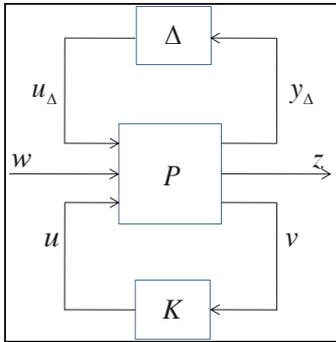


Figure 19: General Control Configuration in  $H_2$ ,  $H_\infty$  Control

The overall control objective is to minimize the magnitude of the transfer function from  $w$  to  $z$ . The magnitude of the transfer function can be measured using  $H_2$  or  $H_\infty$  norms. The controller design problem is then based on the information in  $v$ , finding a controller  $K$  that counteracts the influence of  $w$  on  $z$  therefore minimizing the closed-loop norm from  $w$  to  $z$ .

Most of the control problems, such as reference tracking, disturbance rejection, or even state estimating, can be cast as the general configuration in Figure 19.  $P$  is the generalized plant model, which can include plant model, disturbance model, or designed weighting

functions.  $K$  is the general controller in whatever configuration, which can be broken down into parts; for example, in the two-degree-of-freedom controller  $K = [K_r, K_y]$ , where  $K_r$  is a prefilter and  $K_y$  is a feedback controller.

As an example, consider a helicopter control problem given in [24], where four blade angles were used to control the helicopter pose and movement in the presence of wind disturbances. Figure 20 contains a block diagram of the  $H_\infty$  control of the helicopter. The vector  $y$  contains the outputs to be controlled, which are heave velocity, pitch attitude, roll attitude, heading rate, roll rate, and pitch rate. The vector  $r$  is the reference input. The vector  $d$  contains three components of wind gust that perturb the helicopter's velocity states.  $G$  and  $G_y$  contain plant transfer functions.  $W_1, W_2, W_3, W_4$  and  $K$  are the four weights and controller to be designed.  $z_1$  contains the tracking error.  $z_2$  is the control effort.

The  $H_\infty$  optimization problem is to find a controller  $K$  that minimizes the cost function

$$\left\| \begin{bmatrix} \frac{z_1}{r} & \frac{z_1}{d} \\ \frac{z_2}{r} & \frac{z_2}{d} \end{bmatrix} \right\|_{H_\infty} = \left\| \begin{bmatrix} W_1 S W_3 & -W_1 S G_d W_4 \\ W_2 K S W_3 & -W_2 K S G_d W_4 \end{bmatrix} \right\|_{H_\infty}$$

where  $S$  is the sensitivity function. The  $H_\infty$  norm is used to measure the size of the cost function to be minimized by an optimization numerical method, and hence the name  $H_\infty$  control. By minimizing this cost function, the tracking error  $S_1$  and the control effort  $S_2$ , as a result of the reference input  $r$  and the wind gust



$d$ , are reduced to a small quantity. Note that the weights  $W_1, W_2, W_3$ , and  $W_4$  were selected first to emphasize the minimization of the cost function over various frequencies.

Details of  $H_1, H_\infty$  controls can be found from [2],[25],[26],[27]and[28].

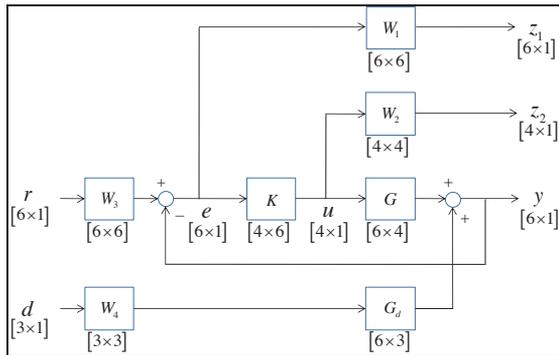


Figure 20: Block Diagram of the  $H_\infty$  Control of a Helicopter

### 3.3 $\mu$ Synthesis

$\mu$  is a structured singular value – a value relates to several conditions used to evaluate robust stability and robust performance of MIMO systems.  $\mu$ -synthesis control is an algorithm to determine the controller that minimizes a given  $\mu$  condition, hence satisfying robust stability and robust performance specifications. The  $\mu$  synthesis also uses  $H_\infty$  norm as a measure.

Details of  $\mu$  synthesis can be found in [2].

### 3.4 Linear Matrix Inequality (LMI)

A linear matrix inequality is an inequality of the form

$$A(x) = A_0 + x_1 A_1 + \dots + x_N A_N < 0,$$

where  $x = (x_1, \dots, x_N)$  is an unknown vector and  $A_0, A_1, \dots, A_N$  are given symmetric matrices.

There are three general LMI problems:

- Feasibility problem  
To find a solution  $x$  to the LMI  $A(x) < 0$
- Linear objective minimization problem  
To minimize  $c^T x$ ; subject to the LMI  $A(x) < 0$ .
- Generalized eigenvalue minimization problem  
To minimize  $\lambda$  subject to  $A(x) < \lambda B(x), B(x) > 0, C(x) < 0$

Many robust control problems and design specifications fall into the three general LMI problems above. The main strength of LMI formulations is the ability to combine various design constraints or objectives in a numerically tractable manner. It can be viewed as combining the strengths of QFT (the ability to quantify uncertainty and various specifications) with  $H_\infty$  (the ability to systematically handle MIMO problem) controls.

Details of LMI can be found in [29].

## REFERENCES

- [1] Levine WS (1996). The Control Handbook. Boca Raton, FL: CRC Press.
- [2] Skogestad S, Postlethwaite I (2005). Multivariable Feedback Control: Analysis and Design. Hoboken, NJ: Wiley.
- [3] Astrom KJ, Hagglund T (1995). PID Controllers: Theory, Design, and Tuning. Research Triangle Park, NC: ISA.
- [4] Astrom KJ, Hagglund T (2005). Advanced PID Control. Research Triangle Park, NC: ISA.
- [5] Ogata K (2002). Modern Control Engineering. Upper Saddle River, NJ: Prentice Hall.
- [6] Dorf RC, Bishop RH (2007). Modern Control Systems. Upper Saddle River, NJ: Prentice Hall.



- [7] Nise N (2008). Control Systems Engineering. Hoboken, NJ: Wiley.
- [8] Franklin GF, Powell JD (2005). Feedback Control of Dynamic Systems. Upper Saddle River, NJ: Prentice Hall.
- [9] Ioannou PA, Sun J (1996). Robust Adaptive Control. Upper Saddle River, NJ: Prentice Hall.
- [10] Astrom KJ, Wittenmark B (2008). Adaptive Control. Mineola, NY: Dover.
- [11] Ioannou PA, Fidan B (2006). Adaptive Control Tutorial. Philadelphia, PA: SIAM.
- [12] Tao G (2003). Adaptive Control Design and Analysis. Hoboken, NJ: Wiley.
- [13] Jang JSR, Sun CT, Mizutani E (1997). Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence. Upper Saddle River, NJ: Prentice Hall.
- [14] Passino KM (2004). Biomimicry for Optimization, Control, and Automation. New York, NY: Springer.
- [15] Ge SS, Hang CC, Lee TH, Zhang T (2001). Stable Adaptive Neural Network Control. New York, NY: Springer.
- [16] Lewis FL, Jagannathan S, Yesildirek A (1998). Neural Network Control of Robot Manipulators and Non-linear Systems. Boca Raton, FL: CRC Press.
- [17] Poznyak AS, Sanchez EN, Yu W (2001). Differential Neural Networks for Robust Nonlinear Control. Farrer Road, Singapore: World Scientific.
- [18] Wang LX (1996). A Course in Fuzzy Systems and Control. Upper Saddle River, NJ: Prentice Hall.
- [19] Passino KM, Yurkovich S (1997). Fuzzy Control. Boston, MA: Addison Wesley.
- [20] Yaniv O (1999). Quantitative Feedback Design of Linear and Nonlinear Control Systems. New York, NY: Springer.
- [21] Houppis CH, Rasmussen SJ, Garcia-Sanz M (2005). Quantitative Feedback Theory: Fundamentals and Applications. Boca Raton, FL: CRC Press.
- [22] Horowitz I (1993). Quantitative Feedback Design Theory. Boulder, CO: QFT Publications.
- [23] Sidi MJ (2001). Design of Robust Control Systems: From Classical to Modern Practical Approaches. Malabar, FL: Krieger.
- [24] Postlethwaite I, Foster NP, Walker DJ (1994). Rotorcraft control law design for rejection of atmospheric turbulence. In: Proceedings of the IEE Conf. Cont., Warwick, UK, pp.1284–1289.
- [25] Zhou K, Doyle JC (1997). Essentials of Robust Control. Upper Saddle River, NJ: Prentice Hall.
- [26] Zhou K, Doyle JC, Glover K (1995). Robust and Optimal Control. Upper Saddle River, NJ: Prentice Hall.
- [27] Doyle JC, Francis BA (2009). Feedback Control Theory. Mineola, NY: Dover.
- [28] Maciejowski JM (1989). Multivariable Feedback Design. Boston, MA: Addison Wesley.
- [29] Boyd S, Ghaoui LEI, Feron E, Balakrishnan V (1997). Linear Matrix Inequalities in System and Control Theory. Philadelphia, PA: SIAM.