7. Adaptive Control

7.1 What is Adaptive Control?

In classical control engineering one has the choice of:

- an open-loop system, where the controller generates the control signal in response to the command input (e.g. an automatic toaster operates this way). See Figure 7-1.

![Figure 7-1 Open-loop control system](image)

- a closed-loop system, where the controller generates the control signal in response to both the command input and the feedback output response (e.g. a thermostat operates this way). See Figure 7-2.

![Figure 7-2 Closed-loop control system](image)

In classical control systems the controller parameters are fixed and thus the controller response is predictable when presented with the same command input and output response. However the closed-loop control system is in a sense “adaptive” since the control signal adapts to different output response that may arise with changes in the plant.

Closed-loop control systems based on linear feedback will not work well when:
1. there are non-linear actuators present in the system.
2. the process dynamics change making the controller either “sluggish” to respond or overcompensate (i.e. overshoot) in response.
3. the plant process is not well understood making the design of the controller difficult.

An adaptive control system uses a controller with adjustable parameters and includes a mechanism for adjusting the parameters. (See Figure 7-3.)
An adaptive control system can be thought of as having two loops. One loop is the normal closed-loop feedback with the plant and controller. The other loop is the parameter adjustment loop which takes a combination of the command input, control signal and output response and adjusts the controller parameters accordingly.

Thus an adaptive control system can cope with:
1. non-linear or unknown plant processes
2. changes in the process dynamics and environment
   by changing the controller parameters accordingly to the new operating region.

**Note** In the following analysis continuous-time systems will be assumed leading to the use of continuous-time functions, differentiation and the Laplace transform in place of discrete-time functions, sample delays and the z-transform. This has been done to be compatible with the discussion and notation from Astrom and Wittenmark[1].

### 7.2 Model-Reference Adaptive System (MRAS)

The MRAS adjusts the controller parameters based on the error, \( e = y - y_m \), between the plant output response, \( y \), and the output from a *reference model* for the desired/expected response of the plant, \( y_m \). This is shown in **Figure 7-4**.

**Figure 7-4** Model-Reference Adaptive System (MRAS)
Example Application: In flight control the MRAS reference model describes the desired response of the aircraft to joystick motions.

7.2.1 The MIT Rule

The mechanism for adjusting the parameters can be obtained by using a gradient descent method as exemplified by the MIT rule.

Consider a controller with one adjustable parameter, $\theta$. The parameter is adjusted such that the loss function defined by:

$$ J = \frac{1}{2} e^2 = \frac{(y - y_m)^2}{2} $$

is minimised. $J$ can be minimised by the gradient descent method where the change in parameter, $\frac{d\theta}{dt}$, is made in the direction of the negative gradient of $J$, that is

$$ -\gamma \frac{\partial J}{\partial \theta} = -\gamma \frac{\partial e}{\partial \theta} $$

where $\gamma$ is the adaptation gain and $\frac{\partial e}{\partial \theta}$ is the sensitivity derivative of the system.

MIT Rule

$$ \frac{d\theta}{dt} = -\gamma \frac{\partial J}{\partial \theta} = -\gamma \frac{\partial e}{\partial \theta} $$

Equation 7.1

**Example 7.1**

**Problem:** Consider the problem of adjusting a feedforward gain for a linear plant process with transfer function, between input $u_c$ and output $y$, of $kG(s)$, where $G(s)$ is known but $k$ is unknown. The controller is required to apply a feedforward gain, $\theta$, to the input, $u_c$ such that the required system transfer function is $G_m(s) = k_0G(s)$, where $k_0$ is a given constant. The required MRAS setup is shown in [Figure 7-5](#). The parameter adjustment block is derived in the solution.

![Figure 7-5 Block diagram of MRAS setup](image)
Solution: The controller function is:

\[ u = \theta u_c \]

where \( \theta \) is the adjustable parameter of the controller and the system transfer function is given by:

\[ G_m(s) = \theta k G(s) \]

If the Process parameter \( k \) was known then the controller parameter would simply be:

\[ \theta = \frac{k_o}{k} \]

However since \( k \) is unknown the MIT rule can be used to adjust the parameter based on the error between the plant response, \( y = k G(p) \theta u_c \), and the reference model response, \( y_m = k_o G(p) u_c \).

**Note** the use of the differential operator \( p = \frac{d}{dt} \) to indicate that the Laplace transform variable, \( s \), in the expression, \( G(s) \), implies differentiation of the corresponding time-varying input and output functions (e.g. if \( G(s) = s \), then \( G(p)u_c = pu_c \) implies \( \frac{du_c}{dt} \)).

Thus:

\[ e = y - y_m = k G(p) \theta u_c - k_o G(p) u_c \]

and

\[ \frac{\partial e}{\partial \theta} = k G(p) u_c = \frac{k}{k_o} y_m \]

giving the MIT rule for the parameter adjustment as:

\[ \frac{d\theta}{dt} = -\gamma \frac{k}{k_o} y_m e = -\gamma y_m e \]

or

\[ s\theta = -\gamma y_m e \quad \Rightarrow \quad \theta = -\frac{\gamma}{s} (y_m e) \]

where \( \gamma = \gamma k / k_o \) is the adaptation gain in place of \( \gamma \). It should be noted that to have the correct sign for \( \gamma \) it is necessary to know the sign of \( k \).

Simulation Study: We consider a simulation of the MRAS for a system with transfer function:

\[ G(s) = \frac{1}{s + 1} \]

excited by a sinusoidal input, \( u_c \), with frequency of 1 rad/s and parameter values, \( k = 1 \) and \( k_o = 2 \). **Figure 7-6** shows how the parameter converges toward the correct value reasonably fast for \( \gamma = 1 \) (top plot) and the convergence rate for different adaptation gains \( \gamma = 0.5, 1, 2 \) (bottom plot).
7.3 Self-Tuning Regulators (STR)

The STR provides adaptive control by adjusting the control parameters based on an underlying design problem and required input specification. Since such design problems can only be solved with knowledge of the process (plant) parameters an STR also includes the estimation of the process parameters (e.g., RLS adaptive filtering or LS estimation). The block diagram of the STR is shown in Figure 7-7 where the Estimation block estimates the process parameters from knowledge of the input and output response and the Controller design block adjusts the control parameters based on meeting the specification of the underlying design problem given the process parameters.

![Block diagram of Self-Tuning Regulator (STR)](image)

**Figure 7-7** Block diagram of Self-Tuning Regulator (STR)
Example 7.2
Problem: Consider the general linear controller shown in Figure 7-8. The design problem of interest is for the controller to provide desired closed-loop poles. One known technique to achieve this is the minimum-degree pole placement (MDPP) algorithm [1, pages 94-97].

Figure 7-8 General linear controller with two degrees of freedom

The MDPP algorithm calculates the required polynomials $R$, $S$ and $T$ of the controller given specification of the polynomial $A_0$ for the desired closed-loop poles and specification of the input-output transfer function:

$$A_m y(t) = B_m u_c(t)$$

via the polynomials $A_m$ and $B_m$. To do this knowledge of the process parameters, $A$ and $B$, are required and these are unknown.

Solution: Formulate the STR adaptive algorithm as follows:

1. Specifications: Desired closed-loop transfer function $B_m/A_m$ and a desired observer polynomial $A_0$ with the desired closed-loop poles.
2. Estimation: Estimate the co-efficients of the process polynomials $A$ and $B$ using the appropriate RLS adaptive filter.
3. Controller design: Given the above specification and estimated process parameters use the MDPP algorithm to calculate the required $R$, $S$ and $T$ polynomials for the controller.

Repeat steps 2 and 3 at each sampling instant.

7.4 References