[Unit 5 & 6 : Fuzzy Controllers/ Non-Linear System and Adaptive Controllers]

Fuzzy Systems (CSc 613)

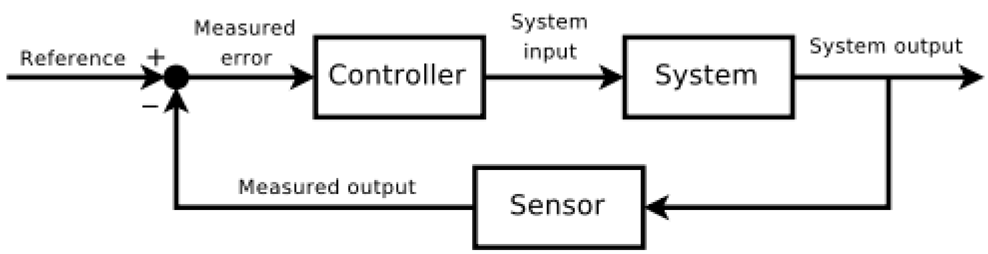
Jagdish Bhatta

Central Department of Computer Science & Information Technology

Tribhuvan University

Control Theory:

Control theory is an interdisciplinary branch of engineering and mathematics that deals with the behavior of dynamical systems with inputs. The external input of a system is called the *reference*. When one or more output variables of a system need to follow a certain reference over time, a controller manipulates the inputs to a system to obtain the desired effect on the output of the system.



The concept of the feedback loop to control the dynamic behavior of the system: this is negative feedback, because the sensed value is subtracted from the desired value to create the error signal, which is amplified by the controller.

Control theory is;

- a theory that deals with influencing the behavior of dynamical systems
- an interdisciplinary subfield of science, which originated in engineering and mathematics, and evolved into use by the social sciences, such as psychology, sociology, criminology and in the financial systems as well.

Control systems may be thought of as having four functions:

- **Measure,**
- **Compare,**
- **Compute, and**
- **Correct.**

These four functions are completed by five elements: Detector, Transducer, Transmitter, Controller, and Final Control Element. The measuring function is completed by the detector, transducer and transmitter.

The usual objective of a control theory is to calculate solutions for the proper corrective action from the controller that result in system stability, that is, the system will hold the set point and not oscillate around it.

Control Loop Basics

A familiar example of a control loop is the action taken when adjusting hot and cold tap to fill a container with water at a desired temperature by mixing hot and cold water. The person touches the water in the container as it fills to sense its temperature. Based on this feedback they perform a control action by adjusting the hot and cold faucets until the temperature stabilizes as desired.

The sensed water temperature is the process variable (PV). The desired temperature is called the setpoint (SP). The input to the process (the water valve position), and the output of the PID controller, is called the manipulated variable (MV) or the control variable (CV). The difference between the temperature measurement and the setpoint is the error (e) and quantifies whether the water in the container is too hot or too cold and by how much.

After measuring the temperature (PV), and then calculating the error, the controller decides how to set the tap position (MV). **The obvious method is proportional control: the tap position is set in proportion to the current error.** A more complex control may include **derivative** action. **This also considers the rate of temperature change: adding extra hot water if the temperature is falling, and less on rising temperature. Finally integral action uses the average temperature in the past to detect whether the temperature of the container is settling out too low or too high and set the tap proportional to the past errors.** An alternative formulation of integral action is to change the current tap position in steps proportional to the current error. Over time the steps add up (which is the discrete time equivalent to integration) the past errors.

In theory, a controller can be used to control any process which has a measurable output (PV), a known ideal value for that output (SP) and an input to the process (MV) that will affect the relevant PV. **Controllers are used in industry to regulate temperature, pressure, force, feed, flow rate, chemical composition, weight, position, speed and practically every other variable for which a measurement exists.**

Similarly, for a car cruise control, a primitive way to implement cruise control is simply to lock the throttle position when the driver engages cruise control. However, if the cruise control is engaged on a stretch of flat road, then the car will travel slower going uphill and faster when going downhill. This type of controller is called an open-**loop controller because no measurement of the system output (the car's speed) is used to alter the control (the throttle position.)** As a result, the controller cannot compensate for changes acting on the car, like a change in the slope of the road.

In a **closed-loop control system**, a sensor monitors the system output (the car's speed) and feeds the data to a controller which adjusts the control (the throttle position) as necessary to maintain the desired system output (match the car's speed to the reference speed.) Now, when the car goes uphill, the decrease in speed is measured, and the throttle position changed to increase engine power, speeding up the vehicle. Feedback from measuring the car's speed has allowed the controller to dynamically compensate for changes to the car's speed. It is from this feedback that the paradigm of the control *loop* arises: the control affects the system output, which in turn is measured and looped back to alter the control.

To overcome the limitations of the open-loop controller, control theory introduces feedback. A **closed-loop controller uses feedback to control states or outputs of a dynamical system. Its name comes from the information path in the system: process inputs (e.g., voltage applied to an electric motor) have an effect on the process outputs (e.g., speed or torque of the motor), which is measured with sensors and processed by the controller; the result (the control signal) is "fed back" as input to the process, closing the loop. The output of the system *y(t)* is fed back through a sensor measurement *F* to the reference value *r(t)*.** The controller *C* then takes the error *e* (difference) between the reference and the output to change the inputs *u* to the system under control *P*.

Fuzzy Control System

A **fuzzy control system** is a control system based on fuzzy logic—a mathematical system that analyzes analog input values in terms of logical variables that take on continuous values between 0 and 1, in contrast to classical or digital logic, which operates on discrete values of either 1 or 0.

Fuzzy controllers are very simple conceptually. They consist of an input stage, a processing stage, and an output stage. The input stage maps sensor or other inputs, such as switches, thumbwheels, and so on, to the appropriate membership functions and truth values. The processing stage invokes each appropriate rule and generates a result for each, then combines the results of the rules. Finally, the output stage converts the combined result back into a specific control output value.

The processing stage is based on a collection of logic rules in the form of IF-THEN statements, where the IF part is called the "antecedent" and the THEN part is called the "consequent". Typical fuzzy control systems have dozens of rules.

If temperature is cold the heater is high

Types of controllers:

1. **On-Off Controller**

In control theory, a bang–bang controller (on–off controller), also known as a hysteresis controller, is a feedback controller that switches abruptly between two states. These controllers may be realized in terms of any element that provides hysteresis. Hysteresis is the dependence of the output of a system not only on its current input, but also on its history of past inputs. The dependence arises because the history affects the value of an internal state. They are often used to control a plant that accepts a binary input, for example a furnace that is either completely on or completely off. Most common residential thermostats are bang–bang controllers.

Simple on–off feedback control systems like these are cheap and effective. In some cases, like the simple compressor example, they may represent a good design choice.

In most applications of on–off feedback control, some consideration needs to be given to other costs, such as wear and tear of control valves and perhaps other start-up costs when power is reapplied each time the PV drops. Therefore, practical on–off control systems are designed to include hysteresis: there is a deadband, a region around the setpoint value in which no control action occurs. The width of deadband may be adjustable or programmable.

Process Gain:

The **Process Gain(K)** is the ratio of change of the output variable(responding variable) to the change of the input variable(forcing function). It specifically defines the sensitivity of the output variable to a given change in the input variable.

$$K = \frac{\Delta Output}{\Delta input}$$

Gain can only be described as a steady state parameter and give no knowledge about the dynamics of the process and is independent of the design and operating variables. **A gain has three components that include the sign, the value, the units. The sign indicates how the output responds to the process input. A positive sign shows that the output variable increases with an increase in the input variable and a negative sign shows that the output variable decreases with an increase in the input variable. The units depend on the process considered that depend on the variables mentioned.**

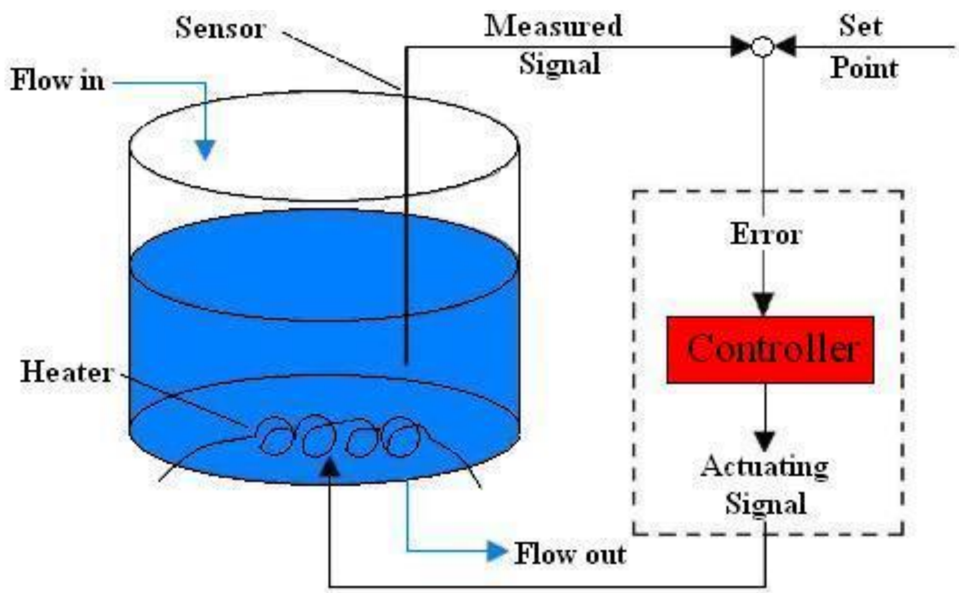
Example:

The pressure was increased from 21psi to 29psi. This change increased the valve position from 30%vp to 22%vp.

$$K = (29-21)\text{psi} / ((22-30)\%\text{vp}) = -1.0\text{psi}/(\%\text{vp})$$

Control:

Process controls are instruments used to control a parameter, such as temperature level, pressure etc. PID controllers are a type of continuous controller because they continually adjust the output vs. an on/off controller, when looking at feed forward or feed backward conditions. An example of a temperature controller is shown below;



Temperature controller in a CSTR

As shown in figure above, the temperature controller controls the temperature of a fluid within a CSTR (Continuous Stirred Tank Reactor). **A temperature sensor first measures the temperature of the fluid. This measurement produces a measurement signal. The measurement signal is then compared to the set point, or desired temperature setting, of the controller. The difference between the measured signal and set point is the error. Based on this error, the controller sends an actuating signal to the heating coil, which adjusts the temperature accordingly.** This type of process control is known **as error-based control because** the actuating signal is determined from the error between the actual and desired setting. **The different types of error-based controls vary in the mathematical way they translate the error into an actuating signal, the most common of which are the PID controllers**.

2. P-Controller

A **proportional control** system is a type of linear feedback control system. In the proportional control algorithm, the controller output is proportional to the error signal, which is the difference between the setpoint and the process variable. In other words, the output of a proportional controller is the multiplication product of the error signal and the proportional gain.

This can be mathematically expressed as

$$P_{out} = K_p\, e(t)+p_0$$

Where,

- p_0 Controller output with zero error.
- P_{out}: Output of the proportional controller
- K_p: Proportional gain
- $e(t)$: Instantaneous process error at time *t*; e(t)=SP-PV
- *SP*: Set point
- *PV*: Process variable

The controller gain is the change in the output of the controller per change in the input to the controller. Therefore, the gain ultimately relates the change in the input and output properties. **If the output changes more than the input, K_c will be greater than 1. If the change in the input is greater than the change in the output, K_c will be less than 1.**

3. PID Controller

A **proportional-integral-derivative controller** (**PID controller**) is a control loop feedback mechanism (controller) widely used in industrial control systems. **A PID controller calculates an *error* value as the difference between a measured process variable and a desired setpoint**. The controller attempts to minimize the *error* by adjusting the process through use of a manipulated variable.

The PID controller algorithm involves three separate constant parameters, and is accordingly **sometimes called three-term control: the proportional, the integral and derivative values, denoted *P, I,* and *D.*** Simply put, these values can be interpreted in terms of time: ***P* depends on the *present* error, *I* on the accumulation of *past* errors, and *D* is a prediction of *future* errors, based on current rate of change.** The weighted sum of these three actions is used to adjust the process via a control element such as the position of a control valve, a damper, or the power supplied to a heating element.

In the absence of knowledge of the underlying process, a PID controller has historically been considered to be the best controller. By tuning the three parameters in the PID controller algorithm, the controller can provide control action designed for specific process requirements. The response of the controller can be described in terms of the responsiveness of the controller to an error, the degree to which the controller overshoots the setpoint, and the degree of system oscillation. The use of the PID algorithm for control does not guarantee optimal control of the system or system stability.

Some applications may require using only one or two actions to provide the appropriate system control. This is achieved by setting the other parameters to zero. **A PID controller will be called a PI, PD, P or I controller in the absence of the respective control actions.** PI controllers are fairly common, since derivative action is sensitive to measurement noise, whereas the absence of an integral term may prevent the system from reaching its target value due to the control action.

The PID control scheme is named after its three correcting terms, whose sum constitutes the manipulated variable (MV). The proportional, integral, and derivative terms are summed to calculate the output of the PID controller. Defining u(t) as the controller output, the final form of the PID algorithm is:

$$\mathrm{u}(t) = \mathrm{MV}(t) = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau + K_d \frac{d}{dt} e(t)$$

where

K_p: Proportional gain, a tuning parameter
K_i: Integral gain, a tuning parameter
K_d: Derivative gain, a tuning parameter
e: Error $= SP - PV$
t: Time or instantaneous time (the present)
τ: Variable of integration; takes on values from time 0 to the present t.

The proportional Term:

The proportional term produces an output value that is proportional to the current error value. The proportional response can be adjusted by multiplying the error by a constant *K_p*, called the proportional gain constant.

The proportional term is given by:

$$P_{\mathrm{out}} = K_p\, e(t)$$

A high proportional gain results in a large change in the output for a given change in the error. If the proportional gain is too high, the system can become unstable. In contrast, a small gain results in a small output response to a large input error, and a less responsive or less sensitive controller. If the proportional gain is too low, the control action may be too small when responding to system disturbances.

The Integral Term:

The contribution from the integral term is proportional to both the magnitude of the error and the duration of the error. The integral in a PID controller is the sum of the instantaneous error over time and gives the accumulated offset that should have been corrected previously. The accumulated error is then multiplied by the integral gain (K_i) and added to the controller output.

The integral term is given by:

$$I_{\mathrm{out}} = K_i \int_0^t e(\tau)\, d\tau$$

The integral term accelerates the movement of the process towards setpoint and eliminates the residual steady-state error that occurs with a pure proportional controller. However, since the integral term responds to accumulated errors from the past, it can cause the present value to overshoot the setpoint value.

The integral of a signal is the sum of all the instantaneous values that the signal has been, from whenever you started counting until you stop counting

The Derivative Term:

The derivative of the process error is calculated by determining the slope of the error over time and multiplying this rate of change by the derivative gain K_d. The magnitude of the contribution of the derivative term to the overall control action is termed the derivative gain, K_d.

The derivative term is given by:

$$D_{\mathrm{out}} = K_d \frac{d}{dt} e(t)$$

Derivative action predicts system behavior and thus improves settling time and stability of the system. An ideal derivative is not causal, so that implementations of PID controllers include an additional low pass filtering for the derivative term, to limit the high frequency gain and noise. So derivative is just a mathematical term meaning rate-of-change. That's all there is to it.

Derivative action improves the controller action because it predicts what is yet to happen by projecting the current rate of change into the future. This means that it is not using the current measured value, but a future measured value.

4. Adaptive Controllers

Adaptive control is the control method used by a controller which must adapt to a controlled system with parameters which vary, or are initially uncertain. For example, as an aircraft flies, its mass will slowly decrease as a result of fuel consumption; a control law is needed that adapts itself to such changing conditions. Adaptive control is different from robust control in that it does not need *a priori* information about the bounds on these uncertain or time-varying parameters; robust control guarantees that if the changes are within given bounds the control law need not be changed, while adaptive control is concerned with control law changing themselves.

The foundation of adaptive control is parameter estimation. Common methods of estimation include recursive least squares and gradient descent. Both of these methods provide update laws which are used to modify estimates in real time (i.e., as the system operates).

In general one should distinguish between:

1. Feedforward Adaptive Control
2. Feedback Adaptive Control

as well as between

1. Direct Methods and
2. Indirect Methods

Direct methods are ones wherein the estimated parameters are those directly used in the adaptive controller. In contrast, indirect methods are those in which the estimated parameters are used to calculate required controller parameters

Need of Adaptive Controllers

Most of real- world processes that require automatic control are nonlinear in nature. That is, their partner values alter as the operating point changes, over time, or both. As conventional control schemes are linear, a controller can only be tuned to give good performance at a particular operating point or for a limited period of time. The controller need to retune if the operating point changes, or retuned periodically if the process changes with time. This necessity to retune has driven the need for adaptive controllers that can automatically retune themselves to match the current process characteristics. An excellent introduction to "conventional" adaptive control systems is by Astrom.

FKBC are nonlinearity and so they can be designed to cope with a certain amount controller must cope with nonlinearity over a significant portion of the operating range of the process. Also the rules of the FKBC do not, in general, contain a temporal component, so they cannot cope with process changes over time. So there is need for adaptive FKBC as well.

Components of Adaptive Controllers

There is still contention as to what exactly constitutes an adaptive controller and there is no consensus on the terminology to use in describing adaptive controller however, to enable the clear classification of the differing attempts at developing adaptive FKBC, we will introduce some definitions and terminology.

Adaptive controllers generally contain two extra components on top of the standard controller itself. The first is a "process monitor" that detects changes in the process characteristics. It is usually in one of two forms:

- a performance measures that assesses how well the controller is controlling, or
- a parameters estimator that constantly updates a model of the process.

The second component is the adaptation mechanism itself. It use information passed to it by the process monitor to updates the controller parameters and so adapts the controller to the changing process characteristics. Adaptive controllers can be classified as performance – adaptive or parameter-adaptive depending on which type of process monitor will be seen in the adaptive FKBC that follow.

FKBC contain a number of sets of parameter that can be altered to modify the controller performance. These are:

- the scaling factor for each variable,
- the fuzzy set representing the meaning of linguistic values,
- the if- then rules

A non-adaptive FKBC ids one in which these parameters do not change once the controller is being used on line .if any of these parameters are altered on line, we will call the controller an adaptive FBKC. Each of these sets of parameters has been used as the controller parameters to be adapted in different adaptive FBKC. for the purposes of this chapter, adaptive FBKC that modify the fuzzy set definitions or the essentially fine tunes an already working controllers. they can either modify an existing set of rules, in which case they are similar to self-tuning controllers, or they can start with no rules at all and "learn" their control strategy as they go.

Design and performance evaluation of adaptive controller

As described above, the adaptive component of an adaptive controller consists of two parts, namely,

1. the process monitor,
2. the adaption mechanism.

The process monitor looks for changes in process characteristics and adaption mechanism alters the controller parameters on the basis of any detected changes. The nature of the performance monitor is not dictated by whether the underlying controller is fuzzy or not. So identical performance monitors can be used in both adaptive FKBC and non-fuzzy adaptive controllers. The adaptation mechanism, on the other hand, is specifically designed for altering the parameters of a FKBC. These two components will now be described in more detail.

The performance monitor

- Parameter Estimators: Changes in process characteristics can either be detected through on-line identification of process model, or assessment of the response of the process. A process model is a mathematical description of the process that gives values of the process-outputs, given the current process-state and inputs to the process. the process-inputs come from the controller. An on line identifier requires the assumption of a particular form of process model.

- The Adaptation Mechanism: The adaptation mechanism must modify the controller parameters to improve the controller performance on the basis of the output from the process monitor. Adaptation mechanism for FKBC can be classified according to which parameters are adjusted. Parameters that can be adjusted include the scaling factors with which controller input and output values are mapped onto the universe of discourse of the fuzzy set definitions.

5. A Model Based Controller

A self-organizing system that uses in line identification of fuzzy process model for adaption has been developed by Graham and Newell. The structure of the controller is shown in fig.

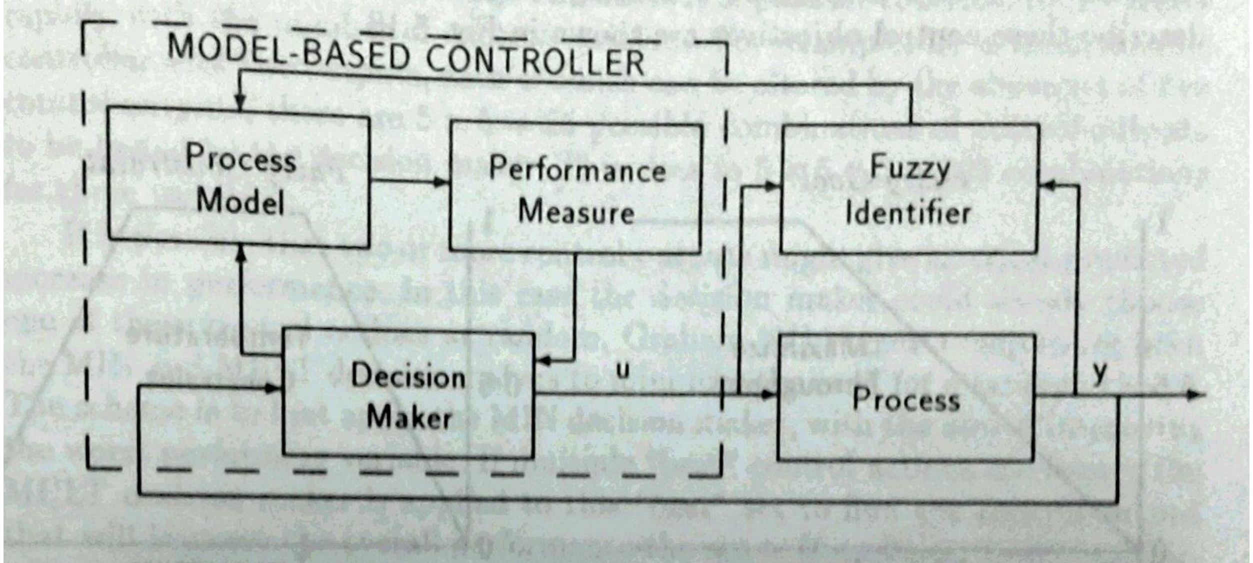


Fig. 5.17. Model-based adaptive fuzzy controller (redrawn from Graham and Newell, [82]).

The controller was originally developed by Hendy(96) and consists of three parts:

1. a fuzzy process modal,
2. a controller performance measure,
3. a decision maker,

The fuzzy process model consists of a set if-then rules that are the inverse of the rules found in a Mamdani FKBC. Instead of specifying the desired control-output for given process state, they predicted the process-output to be expected in time due to the current and previous process-state and control-output. For example, a fuzzy model of a gas furnace may contain a rule such as:

if the current CO2 concentration is MEDEIUM

and the previous methance feedrate was LOW

then the next CO2 concentration will be JUST HIGH.

The controller performance measure consists of group of fuzzy sets. Each fuzzy set describe the required performance of a process variable. The performance of a particular variable is given by the degree of membership of its current value in its performance fuzzy set.

The control-output is calculated by the decision maker, using the fuzzy process model and the performance measure. The aim of the decision maker is to choose, from a predefined, finite set of control-output, which action will maximize the performance of the process if no future control-output are taken. This is achieved by applying each possible control-output to fuzzy process model to obtain prediction of the process-state to be expected due to each control-output. The control-output that produces the predicted process state of highest performance, as specified by the performance measure, is chosen as the current control-output.

The sequence of operations that comprise the model based controller are summarized by the flowchart shown in fig. below;

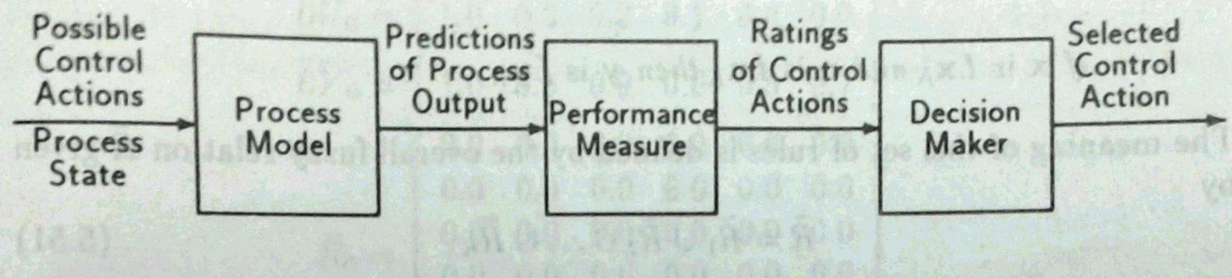
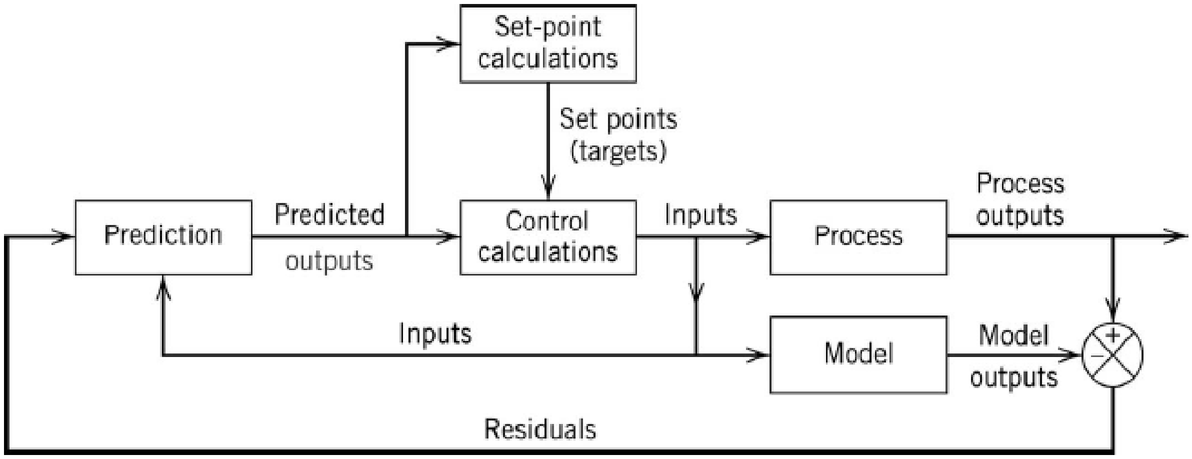


Fig. 5.19. Flowchart of model based controller.

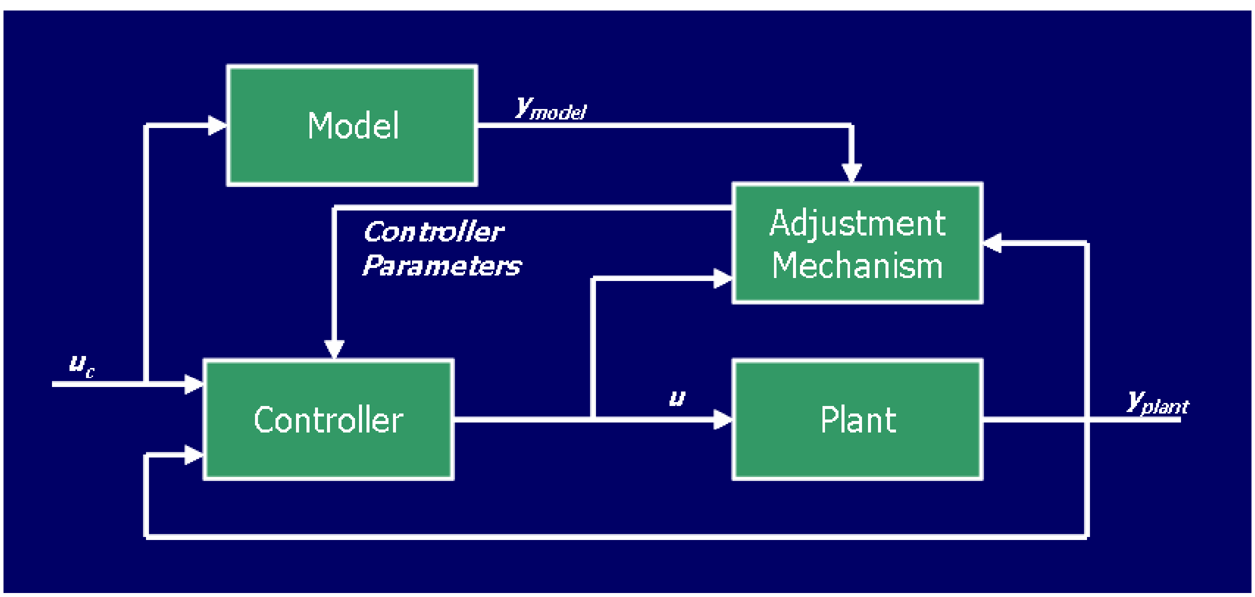
6. Model Predictive Controllers

Future values of output variables are predicted using a dynamic model of the process and current measurements



7. MRAC Controllers

The general idea behind Model Reference Adaptive Control (MRAC, also known as an MRAS or Model Reference Adaptive System) is to create a closed loop controller with parameters that can be updated to change the response of the system. The output of the system is compared to a desired response from a reference model. The control parameters are update based on this error. The goal is for the parameters to converge to ideal values that cause the plant response to match the response of the reference model. For example, you may be trying to control the position of a robot arm naturally vibrates.



Again, the idea behind Model Reference Adaptive Control is to create a closed loop controller with parameters that can be updated to change the response of the system to match a desired model. There are many different methods for designing such a controller. This tutorial will cover design using the MIT rule in continuous time. When designing an MRAC using the MIT rule, the designer chooses: the reference model, the controller structure and the tuning gains for the adjustment mechanism.

MRAC begins by defining the tracking error, e. This is simply the difference between the plant output and the reference model output:

$$e=y_{plant} - y_{model}$$

Non linear systems

In mathematics, a linear function (or map) $f(x)$ is one which satisfies both of the following properties:

- additivity (Superposition), $f(x+y) \;= f(x) \;+ f(y);$
- homogeneity, $f(\alpha x) \;= \alpha f(x).$

(Additivity implies homogeneity for any rational *α*, and, for continuous functions, for any real *α*. For a complex *α*, homogeneity does not follow from additivity; for example, an antilinear map is additive but not homogeneous.) The conditions of additivity and homogeneity are often combined in the superposition principle;

$$f(\alpha x + \beta y) = \alpha f(x) + \beta f(y)$$

An equation written as;

$$f(x) = C$$

is called **linear** if $f(x)$ is a linear map (as defined above) and **nonlinear** otherwise. The equation is called *homogeneous* if $C = 0$.

The definition $f(x) = C$ is very general in that x can be any sensible mathematical object (number, vector, function, etc.), and the function $f(x)$ can literally be any mapping, including integration or differentiation with associated constraints (such as boundary values). If $f(x)$ contains differentiation with respect to x, the result will be a differential equation.

The Nonlinearity of controller

The linearity of a controller is characterized by following properties:

1. **Additivity property or superposition property:**

   Let y1 = *f(x)* *and y2 = f(z).* Then for the additivity property to be abeyed it is required that *y1 +y2 = f(x+z). Hence,*

   f(x) +f(z) = f(x+z).

2. **Scaling property or homogeneity property:**

Let $y = f(x)$. Then for the scaling property to be obeyed it is required that

$$a.y = f(a.x)$$
$$a.\ f(x) = f(a.x).$$

Sliding Mode FKBC

This control method can be applied in the presence of model uncertainties, parameters fluctuation and disturbances provided that the upper bounds of their absolute values are known. the sliding mode control is especially appropriate for the tracking control of robot manipulators and also for motors whose mechanical loads changes over a wide range. The disadvantage of this method is the drastic changes of the control variable which leads to high stress, *e.g.* chattering of gears for the plant to be controlled. However, this can be avoided by a small modification: a boundary layer is introduced near the switching line which smoothen out the control behavior and ensure that the system states remain within this layer. Given that the upper bounds of model uncertainties, etc., are known, stability and high performance of the controlled system are guaranteed in principle, FKBC *work like modified sliding mode controller.* Compared to ordinary sliding mode control, however, FKBC have the advantage of still higher robustness. In the following, the structure of FKBC is derived from a nonlinear state vector, choice of the switching condition, scaling (normalization) of the state vector, choice of the switching line and determination of the break frequencies the controller. By choice of an additional boundary control variable can be avoided, especially at the boundary of the normalized phase plane. In this context, the higher robustness of the modified FKBC over the modified sliding mode control is discussed. Finally the method is extended to the n-dimensional case.

Sliding Mode Control

Sliding Mode Control

Let

$$x^{(n)} = f(\mathbf{x}, t) + u + d, \tag{4.60}$$

where

$$\mathbf{x} = (x, \dot{x}, \ldots, x^{(n-1)})^T$$

$\mathbf{x}$ is the state vector, d are disturbances, and u is the control variable. Furthermore, let

$$f(\mathbf{x}, t) = \hat{f}(\mathbf{x}, t) + \Delta f(\mathbf{x}, t) \tag{4.61}$$

be a nonlinear function of the state vector $\mathbf{x}$ and, explicitly, of time t, where Δf are model uncertainties, and $\hat{f}$ is an estimate of f.

Furthermore, let Δf, d and $x_d^{(n)}$ have upper bounds with known values $\tilde{F}$,D and v:

$$|\Delta f| \leq \tilde{F}(\mathbf{x}, t)\ ; \quad |d| \leq D(\mathbf{x}, t)\ ; \quad |x_d^{(n)}| \leq v. \tag{4.62}$$

The control problem is to obtain the state $\mathbf{x}$ for tracking a desired state $\mathbf{x}_d$ in the presence of model uncertainties and disturbances. With the tracking error

$$\mathbf{e} = \mathbf{x} - \mathbf{x}_d = (e, \dot{e}, \ldots, e^{(n-1)})^T, \tag{4.63}$$

a *sliding surface* (*switching line* for second order systems) is defined as follows:

$$s(\mathbf{x}, t) = 0 \tag{4.64}$$

$$s(\mathbf{x}, t) = (\mathrm{d}/\mathrm{d}t + \lambda)^{n-1} e; \quad \lambda \geq 0. \tag{4.65}$$