Control Structures and Controller design

Beyond PID, Loop-shaping and Pole-placement

- A number of further design techniques and control structures exists, most of which are based on state-space descriptions of the system
- We will briefly introduce
  - Optimal control (Linear quadratic regulator)
  - Kalman filtering

Optimal Control: Linear Quadratic Regulator (LQR) Controller

- The theory of optimal control is concerned with operating a dynamic system at **minimum cost**.
- When the system dynamics are described by a set of linear differential equations and the cost is described by a quadratic function is called the **linear quadratic** (LQ) problem.
- This leads to the **linear–quadratic regulator (LQR)** as a way to choose the feedback gain vector **K**. The LQR is an important part of the solution to the LQG (linear–quadratic–Gaussian) problem.

The Linear Quadratic Regulator (LQR)

Problem: (*The LQR problem*). Consider a linear time-invariant system having a state space model, as defined below:

$$\frac{dx(t)}{dt} = \mathbf{A}x(t) + \mathbf{B}u(t) \qquad x(t_o) = x_o$$

$$y(t) = \mathbf{C}x(t) + \mathbf{D}u(t)$$

We aim to drive the initial state x_o to the **smallest possible value* as soon as possible** in the interval $[t_o, t_f]$, but **without spending too much control effort**.

*With a reference input this corresponds to driving the system to the reference.

Optimal Control: LQR

➤ Linear Quadratic Regulator (LQR) Controller

$$\text{Minimise}_K J = \int_{t_0}^{\infty} x^T Qx + u^T Ru + 2x^T Nu \, dt$$

subject to $\dot{x} = Ax + Bu$

and control error $y = x$

where $u = -Kx$

$$Q \geq 0, \; R > 0, \; N \geq 0$$

System equation at closed-loop: $\dot{x} = (A - BK)x$

➤ Nominal Stability (NS):

$$\text{Re}[\lambda_i(A - BK)] < 0 \;\forall i$$

Without loss of generality, for *N* = 0, control gain is given by $K = R^{-1}B^T X$

where X is the solution to the following **Algebraic Riccati Equation** (ARE):

$$\boxed{A^T X + XA - XBR^{-1}B^T X + Q = 0}$$

(*A*, *B*) must be **stabilisable**!

Optimal control: LQR

$$\text{Minimise}_K J = \int_{t_0}^{\infty} x^T Q x + u^T R u + 2x^T N u \; dt$$

The choice of *Q*, *R* and *N* defines the closed loop characteristics*:*

- *Q* – state weighting matrix: How fast are the states being driven to zero? Which of the states is most important?
- *R* – input weighting matrix: How much control effort should be spent? Which of the control inputs has the highest cost?
- Cross-term *N* is usually set to zero.

Observer design LQG

The LQ design procedure can be reformulated for observer design (using the duality property)

$$\text{Minimise}_L J = \int_{t_0}^{\infty} \tilde{x}^T \boldsymbol{Q_o} \tilde{x} + \tilde{y}^T \boldsymbol{R_o} \tilde{y} + 2\tilde{x}^T \boldsymbol{N_o} \tilde{y}$$

where $\tilde{x} = x - \hat{x}$ is the state estimation error, and $\tilde{y}$ is the output error

Kalman filter

- **Kalman filtering**, also known as **linear quadratic estimation**(**LQE**), is an algorithm that uses a series of measurements observed over time, containing statistical noise and other inaccuracies, and produces estimates of unknown variables that tend to be more accurate than those based on a single measurement alone, by estimating a joint probability distribution over the variables for each timeframe.
- Kalman filters are commonly used as **optimal state estimators (observers).**
- The filter is named after Rudolf E. Kálmán, one of the primary developers of its theory.

Kalman Filter

It can be shown that an **optimal-filter design** can be setup as a quadratic optimisation problem.

This shows that the filter is *optimal* under certain assumptions regarding the signal-generating mechanism (**noise**).

In practice, this property is probably less important than the fact that the resultant filter has the right kind of ***tuning knobs*** so that it can be flexibly applied to a large range of problems of practical interest.

Variations include

- Extended Kalman filter
- Unscented Kalman filter

Kalman Filter

Expectation

Delta-function

$$\delta(x) = \begin{cases} 1, & \text{for } x = 0 \\ 0, & \text{otherwise} \end{cases}$$

- Kalman filter (estimator)
  - For the following system:

$$\dot{x} = Ax + Bu + w_d$$
$$y = Cx + Du + w_n$$

$$E\{w_d(t)w_d^T(\tau)\} = W\delta(t-\tau)$$
$$E\{w_n(t)w_n^T(\tau)\} = V\delta(t-\tau)$$

Design an optimal estimator in order to estimate all states using the measurement output. The estimator is given by

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - \hat{y})$$
$$\hat{y} = C\hat{x} + Du$$

where the Kalman gain is solution of the following:

$$\text{Minimise}_{L(t)} J = \|E\left\{[x(t) - \hat{x}(t)][x(t) - \hat{x}(t)]^T\right\}\|_F^2$$

and the Kalman gain is given by

$$L(t) = P(t)C^T V^{-1}$$

and $P(t)$ is the solution of the following matrix Riccatti Equation:

$$\dot{P}(t) = P(t)A^T + AP(t) - P(t)C^T V^{-1} C P(t) + W$$

Kalman Filter

- Steady-State Kalman Filter

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - \hat{y})$$

$$\hat{y} = C\hat{x} + Du$$

where

$$L(t) = P_{ss}C^T V^{-1}$$

$$\dot{P}(t) = P_{ss}A^T + AP_{ss} - P_{ss}C^T V^{-1} C P_{ss} + W = 0$$

```
>> [msd_kalman,L,P_ss] = kalman(msd_sys, 0.01, 0.1, 0);
```

$$L \approx \begin{bmatrix} 2.41 & 2.90 \end{bmatrix}^T$$

```
zeta=0.01;
omega_n=0.5;

A = [0 1; -omega_n^2  -2*zeta*omega_n];
B = [0; 1];
C = [1 0];
D = 0;
msd_sys = ss(A,B,C,D);
```

$$E\{w_d(t)w_d^T(\tau)\} = 0.01\delta(t - \tau)$$
$$E\{w_n(t)w_n^T(\tau)\} = 0.1\delta(t - \tau)$$

Control structures

- Model predictive control

- Adaptive control

Model predictive control

Model Predictive Control (MPC) Motivation

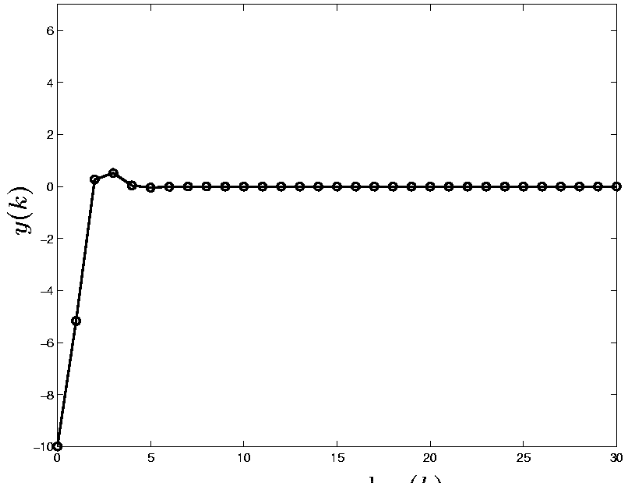
All real world control problems are subject to **constraints** of various types

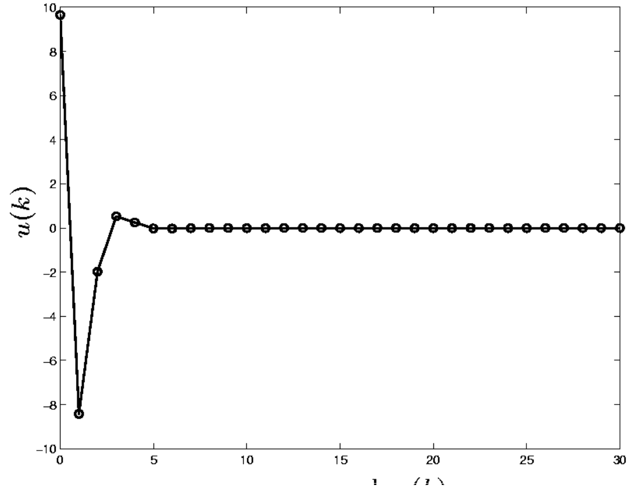
- actuator constraints: amplitude and slew rate limits
- constraints on state variables: e.g. maximal pressures that cannot be exceeded, minimum tank levels, etc.

In many design problems, these constraints can be ignored, at least in the initial design phase.

However, in other problems, these constraints are an inescapable part of the problem formulation since the system operates near a constraint boundary.

Unconstrained Responses



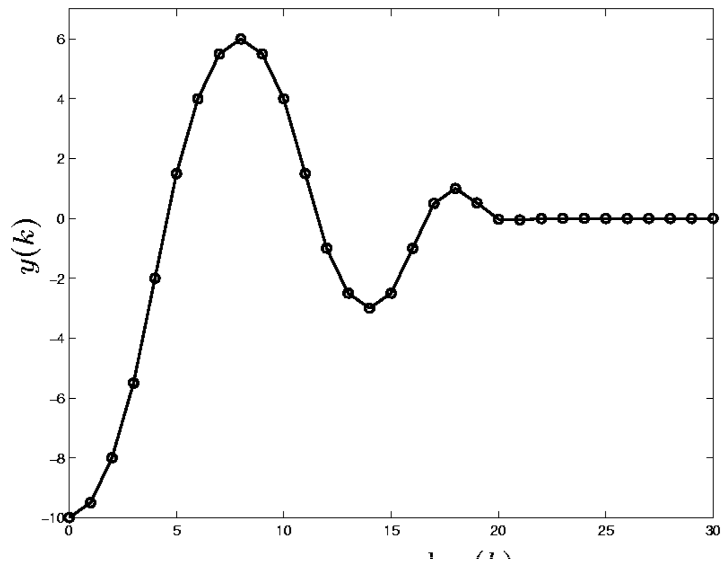


Output response without constraints

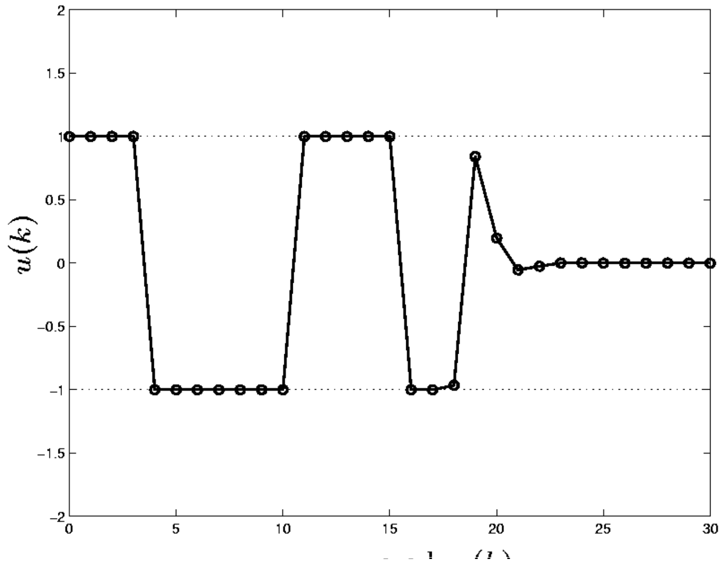
Input response without constraints

Making the constraint more severe

Consider the same set up as above, save that the input is now required to satisfy a more ***severe constraint |u(k)|≤ 1 (****this constraint is 10% of the initial unconstrained input).*



Output response
with constraint |u(k)|≤ 1



Input response
with constraint |u(k)|≤ 1

Discussion of Example with Severe Constraints

We see that the initial input steps have caused the velocity to build up to a large value. **If the control were unconstrained**, this large velocity would help us get to the origin quickly. However, because of the **limited control authority**, the system *braking capacity* is restricted and hence large overshoot occurs. In conclusion, it seems that the control policy has been too *shortsighted* and has not been able to account for the fact that future control inputs would be constrained as well as the current control input.

The solution would seem to be to **try to *look-ahead* (i.e. predict the future response)** and to **take account of *current* and *future* constraints** in deriving the control policy. This leads to the idea of **model predictive control**.

Model Predictive Control (MPC)

Model Predictive Control has actually been a major success story in the application of modern control. More than 2,000 applications of this method have been reported in the literature - predominantly in the petrochemical area. Also, the method is being increasingly used in electromechanical control problems.

Model Predictive Control – Basic Idea

The **model of a process or plant** can be used to **predict its future** values of interesting variables.

Thus, the model can be used to calculate **expected future values of the controlled variables** as a function of **possible control actions**. It is then possible to choose a control action which is **optimal according to some criterion** (e.g. some constraints).

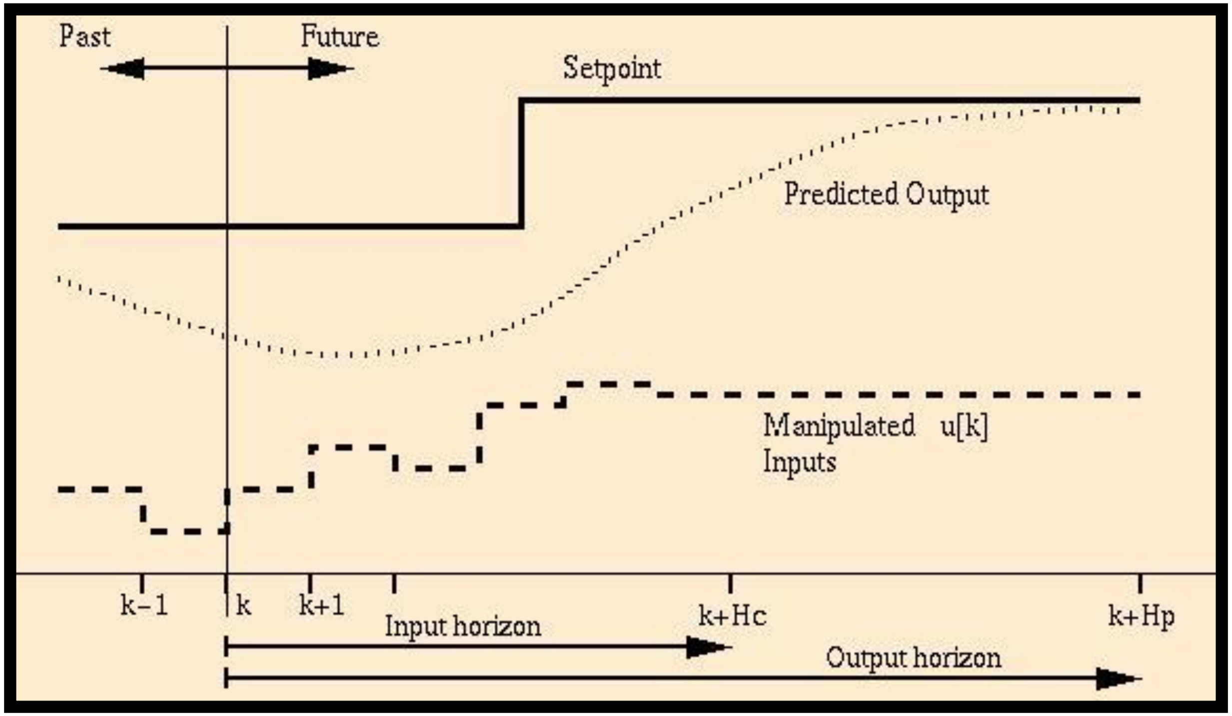
Model Predictive Control: Receding Horizon Control Policy

1) At time *t*, compute or predict the outputs *Hp* time steps into the future, *y(t_k)|t),k=1,2,…Hp,* depending on *Hc* time steps of future control signals *u(t+j), j=0,1,..Hc*, and on measurements known at time *t.*
2) Choose a criterion based on these variables and optimise it with respect to the sequence *u(t+j),j=0,1,…Hc,* taking account of possible constraints.
3) Apply *u(t)* to the plant
4) Wait for the next sampling instant *t+1* and go to 1)

Steps 1) and 2) require to solve, on-line, an **open-loop optimal control problem** over some future interval taking account of the current and *future* constraints, i.e. a **constraint optimisation problem.**

The algorithm is often called a *receding horizon* approach.

Model Predictive Control: Receding Horizon Control Policy



MPC - history

The associated literature can be divided into four generations as follows:

- *First generation* (1970's) - used impulse or step response linear models, quadratic cost function, and ad-hoc treatment of constraints.
- *Second generation* (1980's) - linear state space models, quadratic cost function, input and output constraints expressed as linear inequalities, and quadratic programming used to solve the constrained optimal control problem.
- *Third generation* (1990's) - several levels of constraints (soft, hard, ranked), mechanisms to recover from infeasible solutions.
- *Fourth generation* (late 1990's) - nonlinear problems, guaranteed stability, and robust modifications.

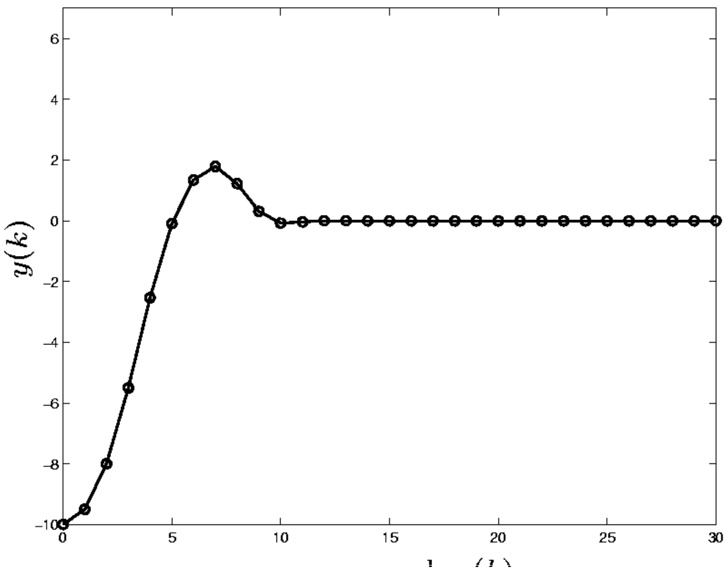
Advantages of Model Predictive Control

The main advantages of MPC are:

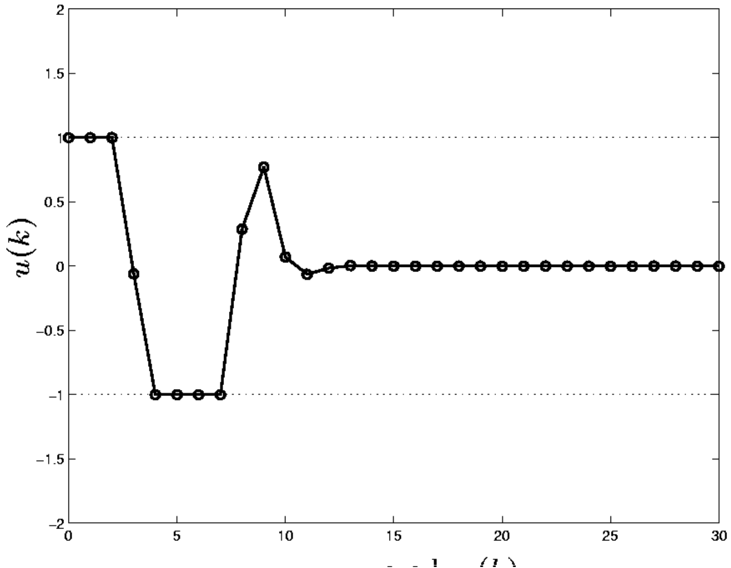
- it provides a one-stop-shop for MIMO control in the presence of constraints,
- it is one of the few methods that allows one to treat state constraints, and
- several commercial packages are available which give industrially robust versions of the algorithms aimed at chemical process control.

Example Revisited

Consider again the problem described earlier with **input constraint $|u(k)| \leq 1$**. Here, we consider the MPC cost function with *Hp =Hc= 2* such that it is the incremental cost associated with the underlying LQR problem, i.e. we consider the constraint on the present and next step. Thus the derivation of the control policy is not quite as *shortsighted* as was previously the case.



Output response using MPC

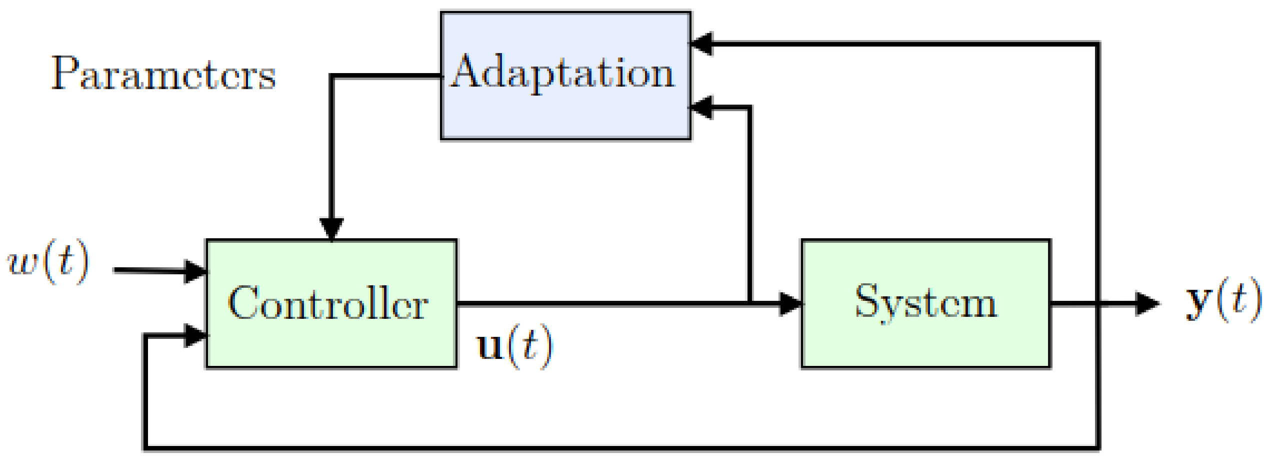


Input response using MPC

Adaptive control

Adaptive Control

Adaptive control is one approach to deal with changing or uncertain plant parameters. Rather than designing a robust controller, the idea is to adjust (**adapt**) the controller to external changes affecting the control feedback loop. This can often have performance advantaged compared to using **robust control** approaches.



Adaptive Control

There are many different approaches to adaptive control. We briefly discuss three main concepts:

- **Gain scheduling:** Switching between pre-defined controller settings, depending on operating conditions.
- **Model reference adaptive system**: Adaptation of the controller parameters, by comparing the current plant outputs to those of a reference plant.
- **Self-tuning regulator:** Estimation of the plant parameters from measurements, and redesign of a controller based on this.

Plasticity – stability dilemma

- **Plasticity**: Controller needs to adapt to new conditions
- **Stability**: Controller needs to “memorise” old patterns

If the adaptation process is biased towards plasticity then noise or transient changes might lead to unwanted parameter changes.

If the adaptation process is biased towards stability then the adaptation algorithm will not be able to react to changes quickly enough.

Gain scheduling

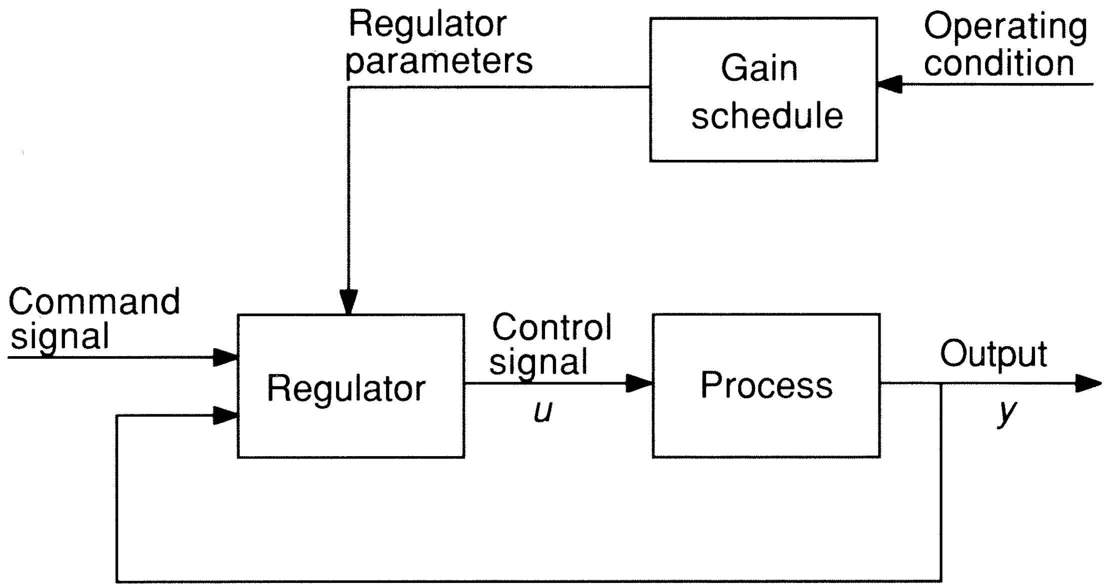


Figure 1.4 Block diagram of a system with gain scheduling.

Model reference adaptive system (MRAS)

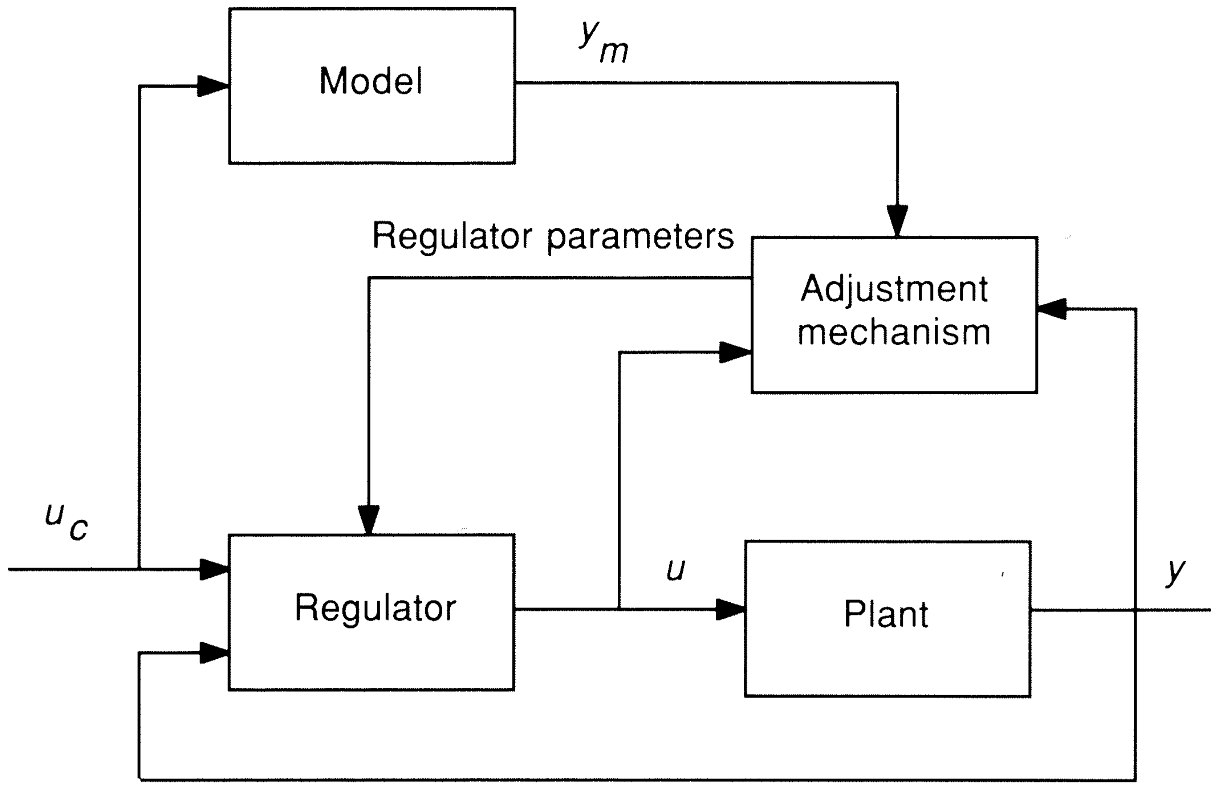


Figure 1.5 Block diagram of a model-reference adaptive system (MRAS).

Self-tuning regulator (STR)

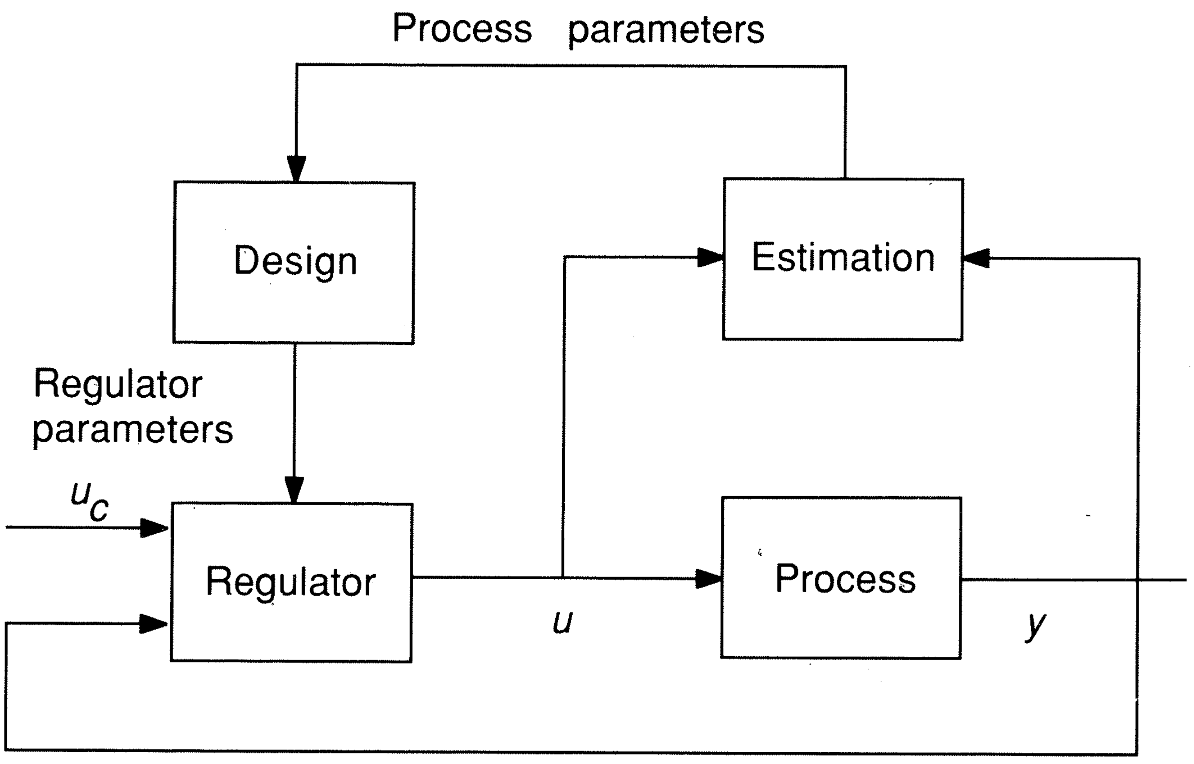


Figure 1.8 Block diagram of a self-tuning regulator (STR).