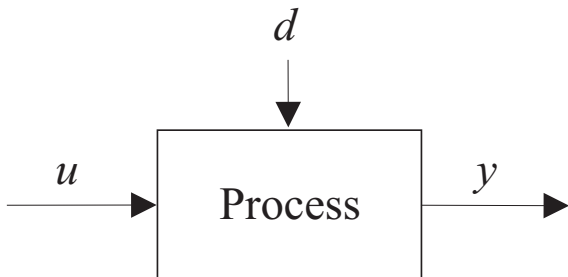


Conventional Control

A Refresher

Process to Be Controlled



y : variable to be controlled (output)

u : manipulated variable (control input)

d : disturbance (input that cannot be influenced)

dynamic system

Examples of “Processes”

- technical (man-made) system

Examples of “Processes”

- technical (man-made) system
- natural environment

Examples of “Processes”

- technical (man-made) system
- natural environment
- organization (company, stock exchange)

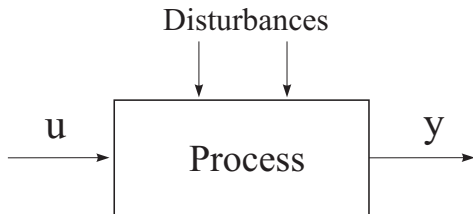
Examples of “Processes”

- technical (man-made) system
- natural environment
- organization (company, stock exchange)
- human body

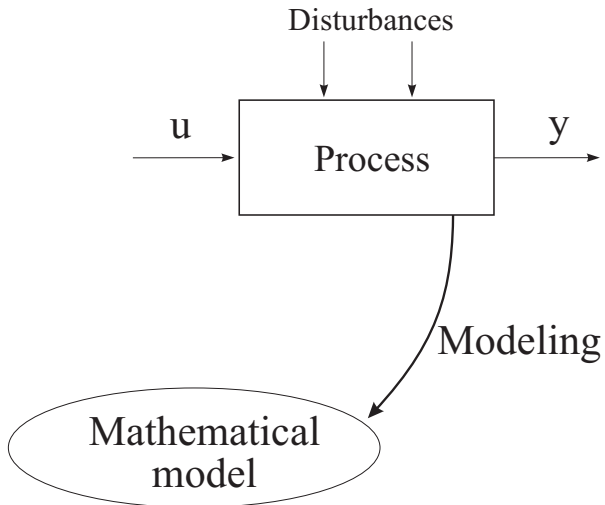
Examples of “Processes”

- technical (man-made) system
- natural environment
- organization (company, stock exchange)
- human body
- ...

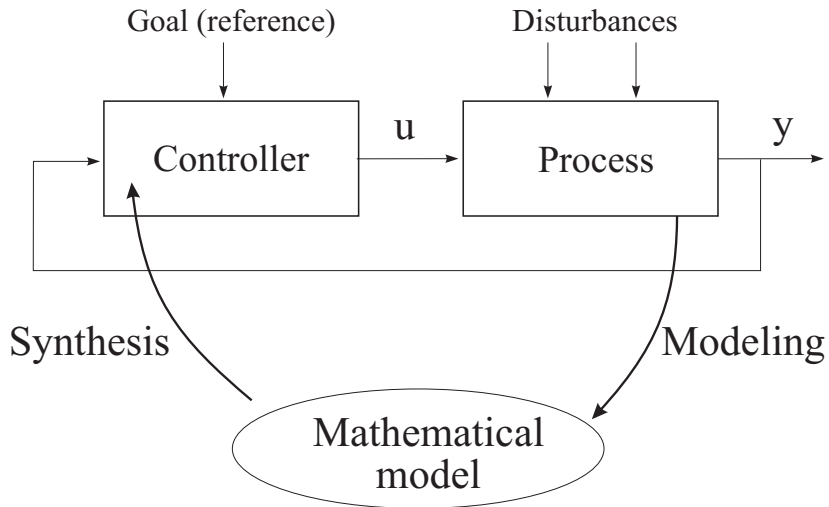
Classical Control Design



Classical Control Design



Classical Control Design



How to Obtain Models?

- **physical (mechanistic) modeling**
 - ① first principles \rightarrow differential equations (linear or nonlinear)
 - ② linearization around an operating point
- **system identification**
 - ① measure input–output data
 - ② postulate model structure (linear–nonlinear)
 - ③ estimate model parameters from data (least squares)

Modeling of Dynamic Systems

$x(t)$... state of the system

summarizes all history such that if we know $x(t)$ we can predict its development in time, $\dot{x}(t)$, for any input $u(t)$

linear state-space model:

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

Modeling of Dynamic Systems

$x(t)$... state of the system

summarizes all history such that if we know $x(t)$ we can predict its development in time, $\dot{x}(t)$, for any input $u(t)$

linear state-space model:

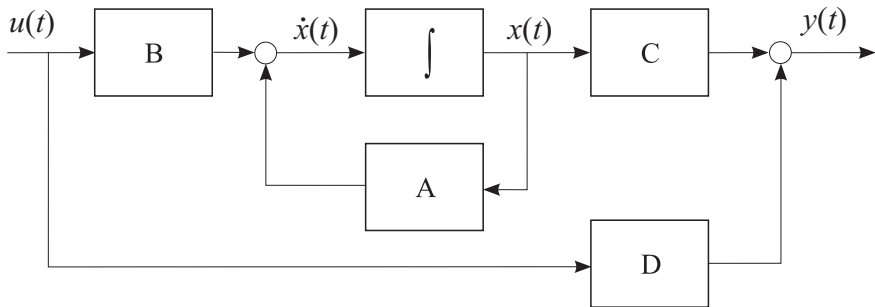
$$\dot{x}(t) = Ax(t) + Bu(t)$$

$$y(t) = Cx(t) + Du(t)$$

Continuous-Time State-Space Model

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

$$y(t) = Cx(t) + Du(t)$$

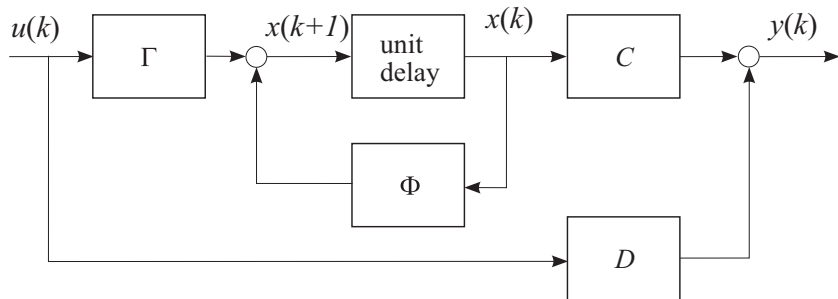


Discrete-Time State-Space Model

$$\begin{aligned}x(k+1) &= \Phi x(k) + \Gamma u(k) \\y(k) &= Cx(k) + Du(k)\end{aligned}$$

Discrete-Time State-Space Model

$$\begin{aligned}x(k+1) &= \Phi x(k) + \Gamma u(k) \\ y(k) &= Cx(k) + Du(k)\end{aligned}$$



Input–Output Models

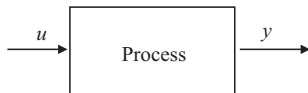
Continuous time:

$$y^{(n)}(t) = f\left(y^{(n-1)}(t), \dots, y^{(1)}(t), y(t), u^{(n-1)}(t), \dots, u^{(1)}(t), u(t)\right)$$

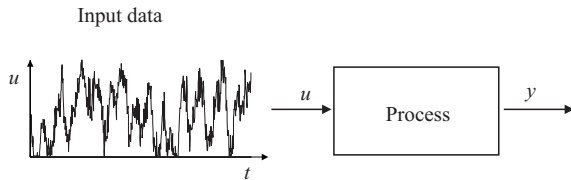
Discrete time:

$$y(k+1) = f\left(y(k), y(k-1), \dots, y(k-n_y+1), \dots, u(k), u(k-1), \dots, u(k-n_u+1)\right)$$

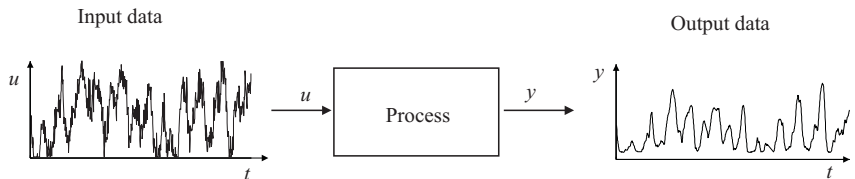
System Identification



System Identification



System Identification



$u(1), u(2), \dots, u(N)$

$y(1), y(2), \dots, y(N)$

System Identification

Given data set

$\{(u(k), y(k)) \mid k = 1, 2, \dots, N\}$:

- 1 Postulate model structure, e.g.:

$$\hat{y}(k + 1) = ay(k) + bu(k)$$

System Identification

Given data set

$\{(u(k), y(k)) \mid k = 1, 2, \dots, N\}$:

- 1 Postulate model structure, e.g.:

$$\hat{y}(k+1) = ay(k) + bu(k)$$

- 2 Form regression equations:

$$y(2) = ay(1) + bu(1)$$

$$y(3) = ay(2) + bu(2)$$

\vdots

$$y(N) = ay(N-1) + bu(N-1)$$

in a matrix form: $\mathbf{y} = \boldsymbol{\varphi}[a \ b]^T$

System Identification

3. Solve the equations for $[a \ b]$ (least-squares solution):

$$\mathbf{y} = \boldsymbol{\varphi} [a \ b]^T$$

System Identification

3. Solve the equations for $[a \ b]$ (least-squares solution):

$$\begin{aligned} \mathbf{y} &= \boldsymbol{\varphi}[a \ b]^T \\ \boldsymbol{\varphi}^T \mathbf{y} &= \boldsymbol{\varphi}^T \boldsymbol{\varphi}[a \ b]^T \end{aligned}$$

System Identification

3. Solve the equations for $[a \ b]$ (least-squares solution):

$$\begin{aligned} \mathbf{y} &= \boldsymbol{\varphi}[a \ b]^T \\ \boldsymbol{\varphi}^T \mathbf{y} &= \boldsymbol{\varphi}^T \boldsymbol{\varphi}[a \ b]^T \\ [a \ b]^T &= [\boldsymbol{\varphi}^T \boldsymbol{\varphi}]^{-1} \boldsymbol{\varphi}^T \mathbf{y} \end{aligned}$$

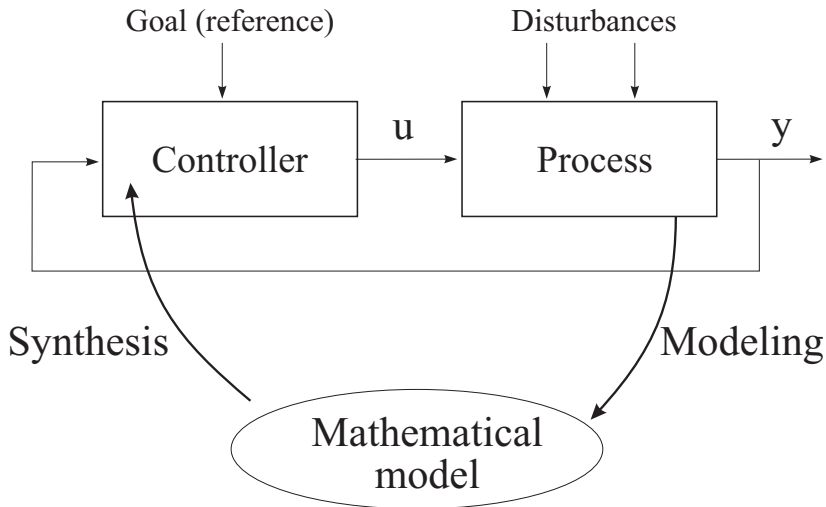
System Identification

3. Solve the equations for $[a \ b]$ (least-squares solution):

$$\begin{aligned} \mathbf{y} &= \boldsymbol{\varphi} [a \ b]^T \\ \boldsymbol{\varphi}^T \mathbf{y} &= \boldsymbol{\varphi}^T \boldsymbol{\varphi} [a \ b]^T \\ [a \ b]^T &= [\boldsymbol{\varphi}^T \boldsymbol{\varphi}]^{-1} \boldsymbol{\varphi}^T \mathbf{y} \end{aligned}$$

Numerically better methods are available
(in MATLAB $[a \ b] = \boldsymbol{\varphi} \setminus \mathbf{y}$).

Classical Control Design



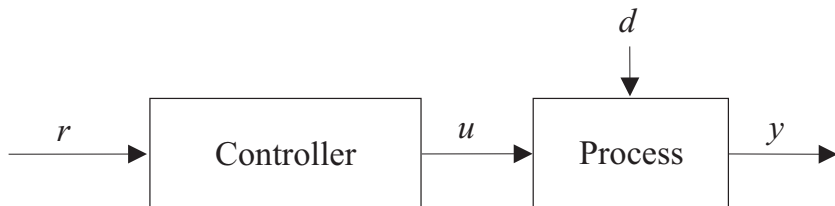
Design Procedure

- **Criterion** (goal)
 - stabilize an unstable process
 - suppress influence of disturbances
 - improve performance (e.g., speed of response)
- **Structure** of the controller
- **Parameters** of the controller (tuning)

Taxonomy of Controllers

- Presence of feedback: feedforward, feedback, 2-DOF
- Type of feedback: output, state
- Presence of dynamics: static, dynamic
- Dependence on time: fixed, adaptive
- Use of models: model-free, model-based

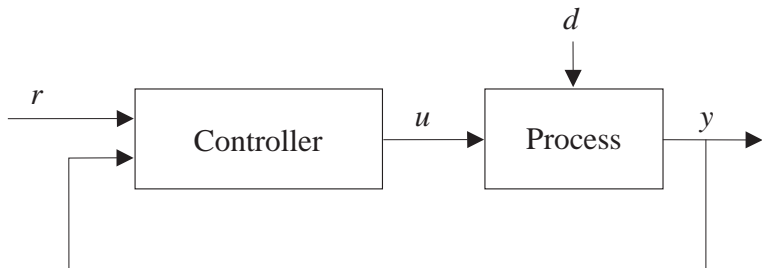
Feedforward Control



Controller:

- (dynamic) inverse of process model
- cannot stabilize unstable processes
- cannot suppress the effect of d
- sensitive to uncertainty in the model

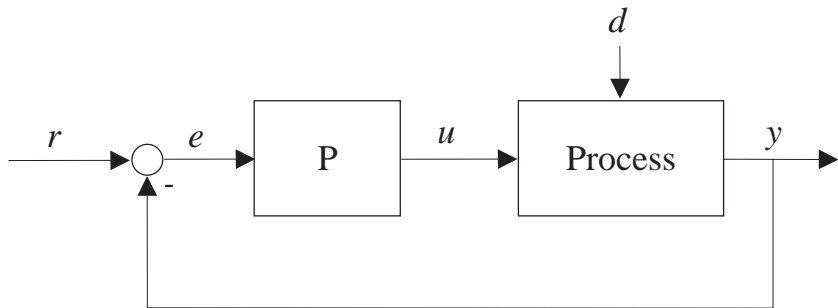
Feedback Control



Controller:

- dynamic or static (\neq inverse of process)
- can stabilize unstable processes (destabilize stable ones!)
- can suppress the effect of d

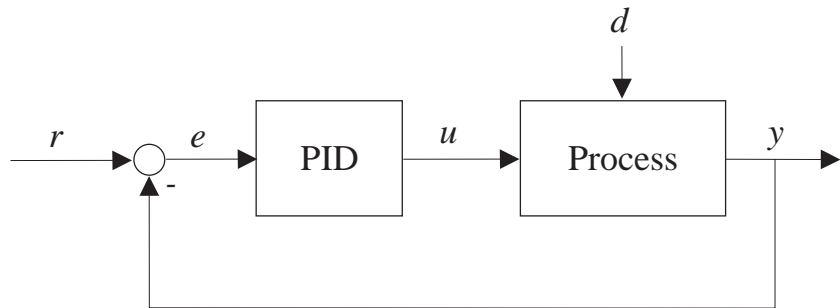
Proportional Control



Controller:

- static gain P : $u(t) = Pe(t)$

PID Control

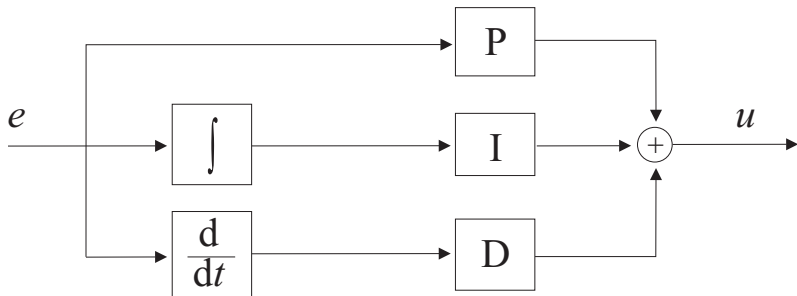


Controller:

- dynamic: $u(t) = Pe(t) + I \int_0^t e(\tau) d\tau + D \frac{de(t)}{dt}$
- P , I and D are the **proportional**, **integral** and **derivative** gains, respectively

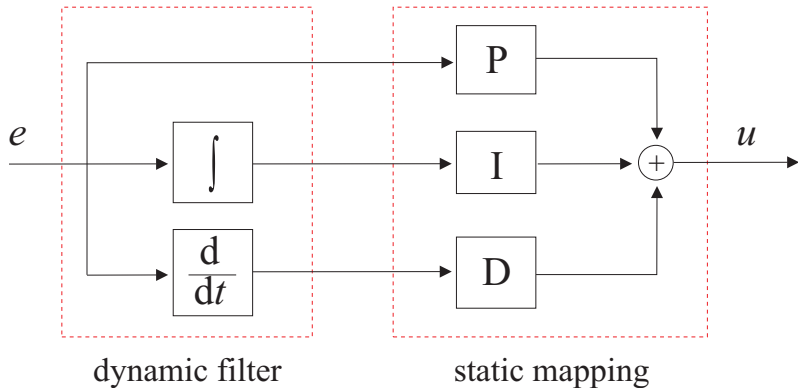
PID Control: Internal View

$$u(t) = Pe(t) + I \int_0^t e(\tau) d\tau + D \frac{de(t)}{dt}$$

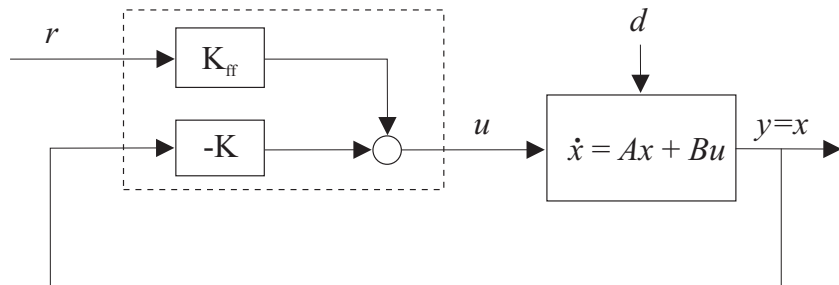


PID Control: Internal View

$$u(t) = Pe(t) + I \int_0^t e(\tau) d\tau + D \frac{de(t)}{dt}$$



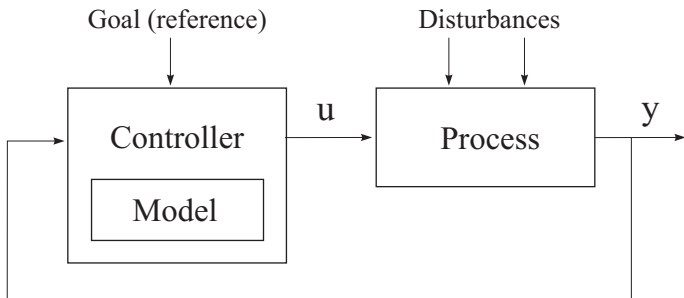
State Feedback



Controller:

- static: $u(t) = Kx(t)$
- K can be computed such that $(A + BK)$ is stable
- K_{ff} takes care of the (unity) gain from r to y

Model-Based Control



- state observer
- model-based predictive control
- adaptive control