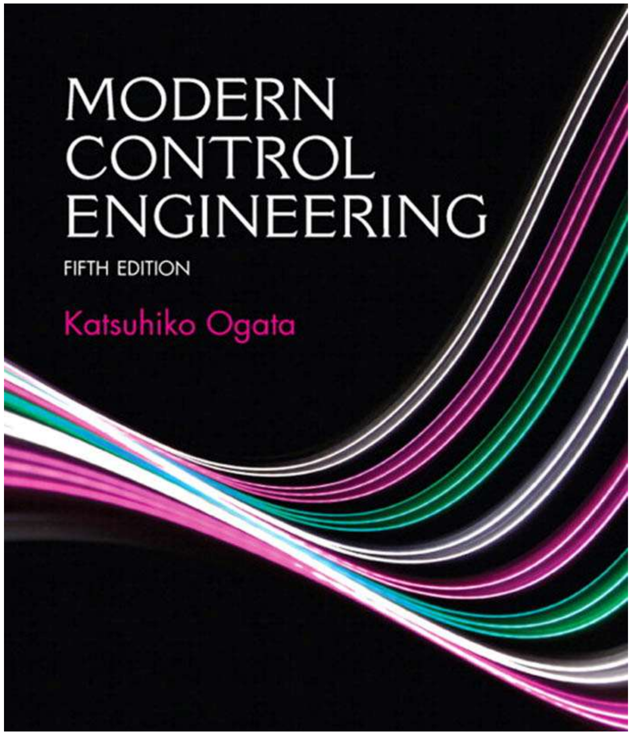
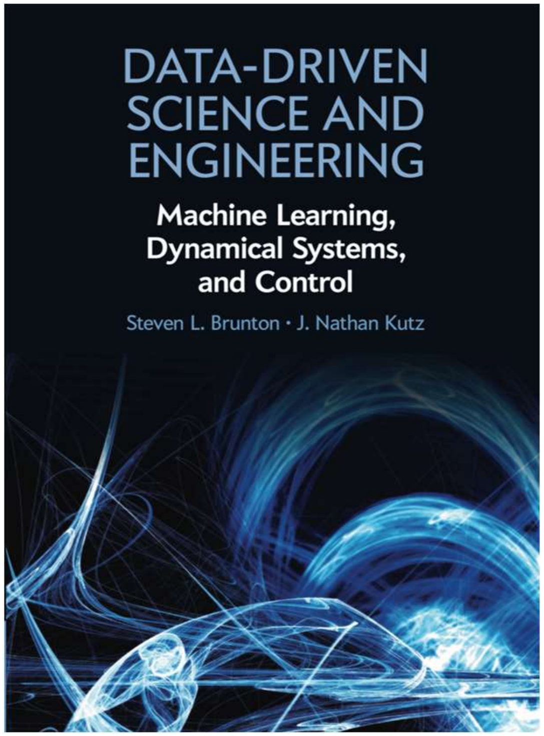
EECE5550 – Mobile Robotics

Closed-loop Feedback Control

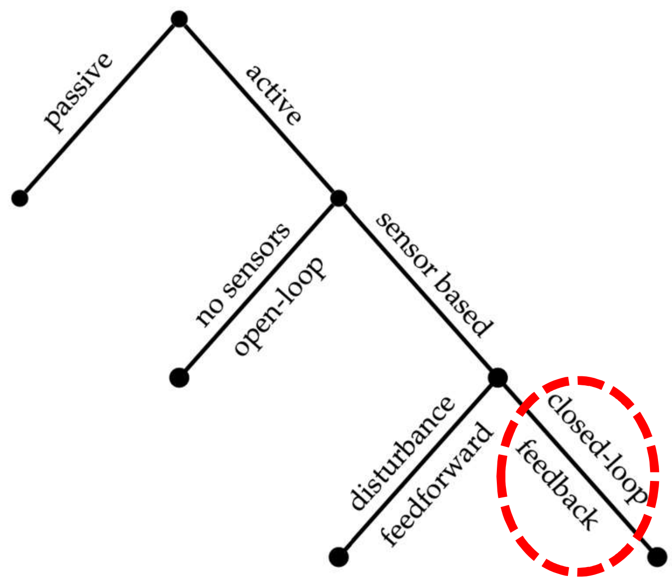
Reference Books

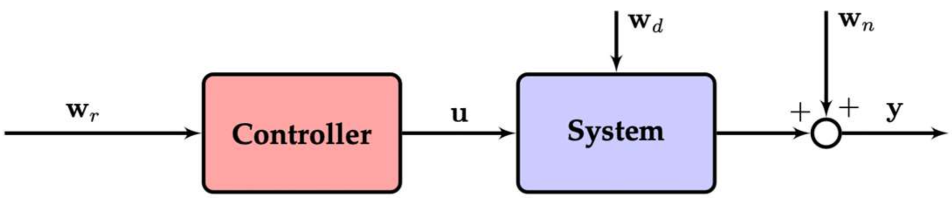




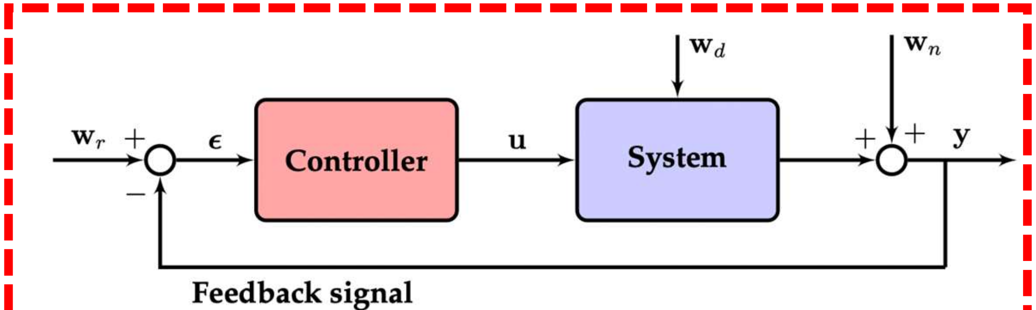
Why close the loop?

- Types of control





Open-loop Control Diagram

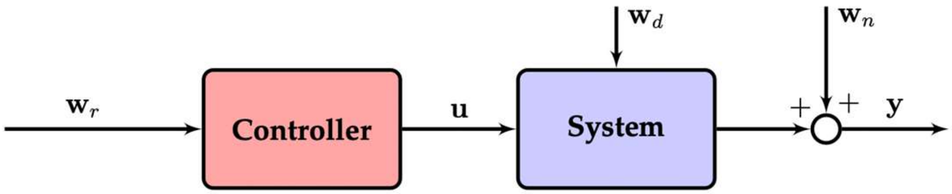


Closed-loop Control Diagram

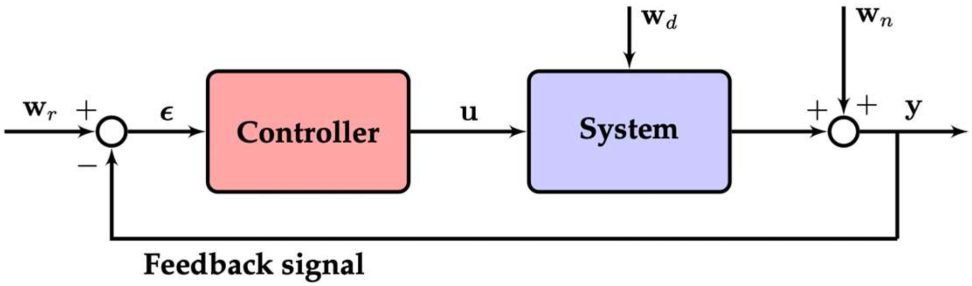
Figures are from Data-Driven Science and Engineering Book by J. Nathan Kutz and Steven L. Brunton

Why close the loop?

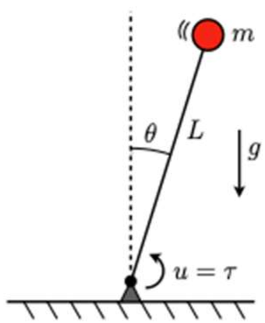
- Uncertainty
  - System model inaccuracies
- Instability
  - Stabilizing system dynamics
- Disturbances
  - Unmodelled errors on system
- Efficiency



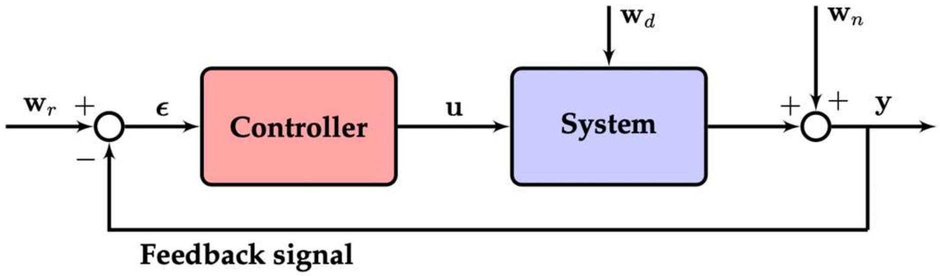
Open-loop Control Diagram



Closed-loop Control Diagram



Basic Feedback Control

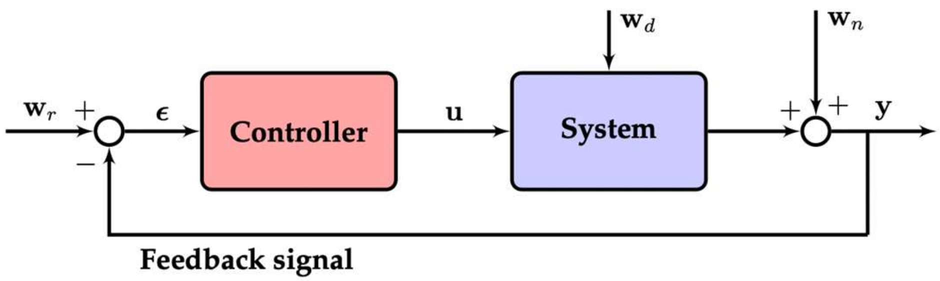


Closed-loop Control Diagram

- System dynamics and measurements
  - $\dot{x} = f(x, u, w_d)$
  - $y = g(x, u, w_n)$

- Goal is to design a controller $u = k(x, w_r)$ that will hold the system in a desired state
  - $\dot{x} = f(x, k(x, w_r), w_d)$

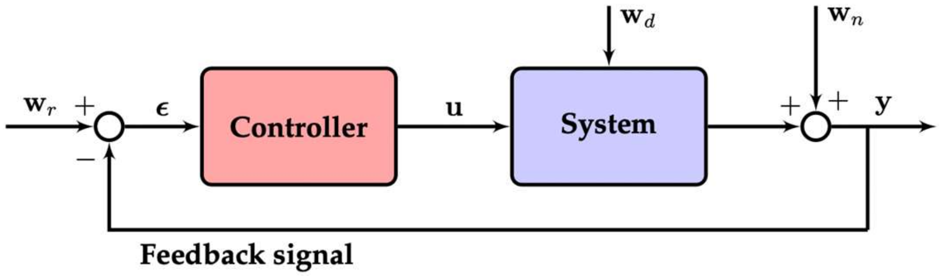
Basic Feedback Control



Closed-loop Control Diagram

- Motivating example: Cruise control
  - Let y be the car's speed and u be the gas fed into the engine
    - A simple model: $y = u$
    - An open-loop controller: $u = w_r$
    - Error is zero: $e = y - w_r$
  - BUT
    - If we have incorrect model i.e., in actuality $y = 2u$
    - What if go from driving flat to up hill? i.e., if $y = u + \sin(t)$
    - Open-loop would not work!
  - In contrast, the closed-loop control would reduce the error:
    - Closed-loop control: $u = K(y - w_r)$
    - For the actual model: $\mathrm{y} = 2\mathrm{u}$ $y = \frac{2K}{2K+1} w_r$ and K=50 then we have 1% steady-state error.

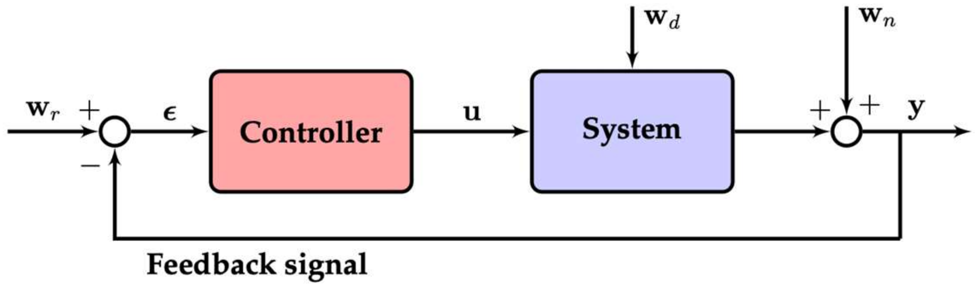
Basic Feedback Control



Closed-loop Control Diagram

- Linear time-invariant systems (LTI)
  - $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x}$
  - Where $\boldsymbol{x} \in R^n, \boldsymbol{A} \in R^{n\times n} constant$
- We know that $\boldsymbol{x}(t) = e^{\boldsymbol{A}t}\boldsymbol{x}(0)$ where $\boldsymbol{x}(0)$is the initial state.
- Stability:
  - An LTI system is **asymptotically stable** if and only if all the eigenvalues of A have **strictly negative real parts**.
- The simplest case: single-input single-output (SISO)
  - $x(t) = e^{at}x(0)$ if $a \geq 0$ the state blows as time goes to infinity.

Basic Feedback Control



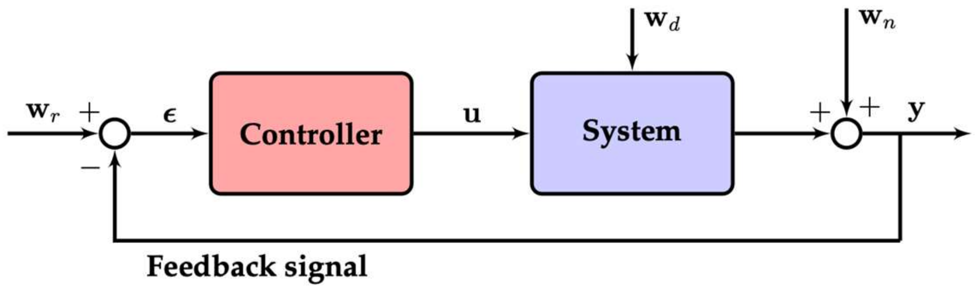
Closed-loop Control Diagram

- Stability:
  - Multi-input multi-output (MIMO)
    - $\boldsymbol{x}(t) = e^{\boldsymbol{A}t}\boldsymbol{x}(0)$
    - Eigen decomposition of A : $\boldsymbol{AT} = \boldsymbol{T\Lambda}$ where T is the eigenvector and Λ is the diagonal matrix of eigenvalues.
    - $\boldsymbol{e^{At} = e^{T\Lambda T^{-1}} = Te^{\Lambda t}T^{-1}}$
      - For full derivation -> Section 8.2 of Data Driven Science and Engineering Book

$$e^{\boldsymbol{\Lambda}t} = \begin{bmatrix} e^{\lambda_1 t} & 0 & \cdots & 0 \\ 0 & e^{\lambda_2 t} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & e^{\lambda_n t} \end{bmatrix}$$

    - Eigenvalues may be complex values:
      - $\lambda = a + ib$
      - If ALL of the eigen values have real negative real part i.e., $Re(\lambda) = a < 0$ then the system is stable.

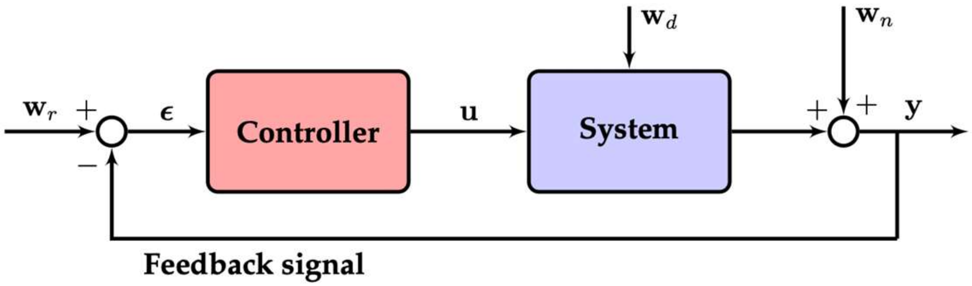
Basic Feedback Control



Closed-loop Control Diagram

- Eigenvalues and Linear Phase Portraits
  - Visualization in Wolfram Player

Basic Feedback Control



Closed-loop Control Diagram

- Controlling LTI systems
  - $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}u \rightarrow \dot{\boldsymbol{x}} = (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K})x$
  - $\boldsymbol{y} = \boldsymbol{C}\boldsymbol{x}$
  - The problem is reduced to an autonomous system of differential equations.
  - Select a K such that the system is stabilized:
    - Assuming C is identity
    - No control input $\boldsymbol{x}(t) = e^{\boldsymbol{A}t}\boldsymbol{x}(0)$
    - With control input $\boldsymbol{x}(t) = e^{(\boldsymbol{A}-\boldsymbol{B}\boldsymbol{K})t}\boldsymbol{x}(0) = \mathrm{e}^{\tilde{A}t}x(0)$
    - Analyze $\tilde{A}$ and select the eigenvalues that would stabilize the system

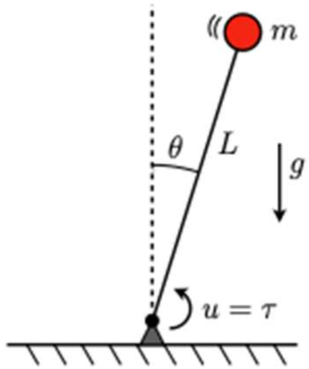
Linearization

- Majority of the systems are nonlinear
  - $\dot{x} = f(x, u) \quad y = g(x, u)$
- To apply linear control theory, we linearize the system at fixed points:
  - Find fixed points $(\hat{x}, \hat{u})$ where $f(\hat{x}, \hat{u}) = 0$
  - Expand the input-output functions in a Tyler series for small displacements $\Delta x = x - \tilde{x} \quad \Delta u = u - \tilde{u}$

$$\mathbf{f}(\bar{\mathbf{x}} + \Delta\mathbf{x}, \bar{\mathbf{u}} + \Delta\mathbf{u}) = \mathbf{f}(\bar{\mathbf{x}}, \bar{\mathbf{u}}) + \underbrace{\left.\frac{\mathbf{df}}{\mathbf{dx}}\right|_{(\bar{\mathbf{x}},\bar{\mathbf{u}})}}_{\mathbf{A}} \cdot\Delta\mathbf{x} + \underbrace{\left.\frac{\mathbf{df}}{\mathbf{du}}\right|_{(\bar{\mathbf{x}},\bar{\mathbf{u}})}}_{\mathbf{B}} \cdot\Delta\mathbf{u} + \cdots . \qquad \mathbf{g}(\bar{\mathbf{x}} + \Delta\mathbf{x}, \bar{\mathbf{u}} + \Delta\mathbf{u}) = \mathbf{g}(\bar{\mathbf{x}}, \bar{\mathbf{u}}) + \underbrace{\left.\frac{\mathbf{dg}}{\mathbf{dx}}\right|_{(\bar{\mathbf{x}},\bar{\mathbf{u}})}}_{\mathbf{C}} \cdot\Delta\mathbf{x} + \underbrace{\left.\frac{\mathbf{dg}}{\mathbf{du}}\right|_{(\bar{\mathbf{x}},\bar{\mathbf{u}})}}_{\mathbf{D}} \cdot\Delta\mathbf{u} + \cdots .$$

  - Drop the higher order terms as they are negligible:
    - $\dot{x} = Ax + Bu \;\; y = Cx + Du$

Example: Inverted pendulum



- System dynamics: $\ddot{\theta} = -\frac{g}{L}\sin(\theta) + u.$
- State: angular position and velocity. Control input: torque.

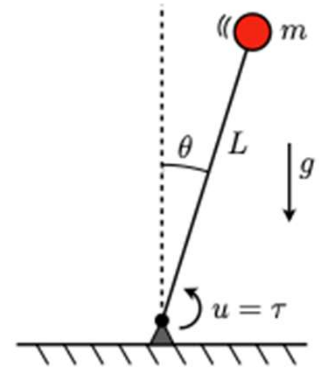
$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} \quad \Longrightarrow \quad \frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -\frac{g}{L}\sin(x_1) + u \end{bmatrix}$$

- Take the derivative: $\frac{\mathbf{df}}{\mathbf{dx}} = \begin{bmatrix} 0 & 1 \\ -\frac{g}{L}\cos(x_1) & 0 \end{bmatrix}, \quad \frac{\mathbf{df}}{\mathbf{du}} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$

- Linearize at fixed points: up $(x_1 = \pi, x_2 = 0)$ down $(x_1 = 0, x_2 = 0)$
- $\underbrace{\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{g}{L} & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix}u}_{\text{Pendulum up,} \quad \lambda=\pm\sqrt{g/L}}$ $\quad \underbrace{\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{g}{L} & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix}u}_{\text{Pendulum down,} \quad \lambda=\pm i\sqrt{g/L}}$

Is the system stable?
(Assume $\sqrt{g}/L$=1)

Example: Inverted pendulum

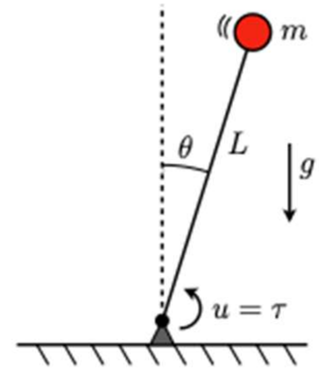


- Is the system stable?

$$\underbrace{\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{g}{L} & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u}_{\text{Pendulum up,} \quad \lambda = \pm\sqrt{g/L}} \qquad \underbrace{\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{g}{L} & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u}_{\text{Pendulum down,} \quad \lambda = \pm i\sqrt{g/L}}$$

  - Yes, if pendulum down
  - No, if pendulum up because positive eigen value
    - $\lambda = \pm 1$
- Let's apply closed-loop control $\dot{\boldsymbol{x}} = (\boldsymbol{A} - \boldsymbol{BK})x$
- Can you find a K that will stabilize the pendulum when it is up?
  - Assume $\sqrt{g}/L$=1

Example: Inverted pendulum



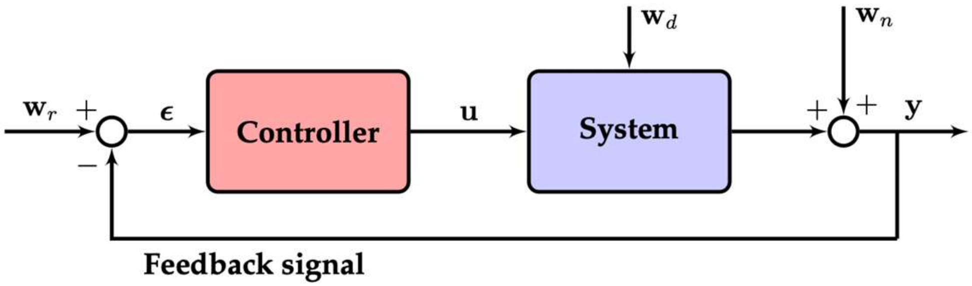
- Is the system stable?

$$\underbrace{\frac{d}{dt}\begin{bmatrix}x_1\\x_2\end{bmatrix} = \begin{bmatrix}0&1\\\frac{g}{L}&0\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix}+\begin{bmatrix}0\\1\end{bmatrix}u}_{\text{Pendulum up}, \quad \lambda=\pm\sqrt{g/L}} \qquad \underbrace{\frac{d}{dt}\begin{bmatrix}x_1\\x_2\end{bmatrix} = \begin{bmatrix}0&1\\-\frac{g}{L}&0\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix}+\begin{bmatrix}0\\1\end{bmatrix}u}_{\text{Pendulum down}, \quad \lambda=\pm i\sqrt{g/L}}$$

  - Yes, if pendulum down
  - No, if pendulum up because positive eigen value
    - $\lambda = \pm 1$
- Let's apply closed-loop control $\dot{\boldsymbol{x}} = (\boldsymbol{A} - \boldsymbol{BK})x$
- Select $K = [4, 4]$ and analyze eigenvalues ->
  - The new eigenvalues are -1, -7
  - The system is stabilized for the up position for small displacements of control input

```
In [10]: A = np.array([[0, 1], [1, 0]])

In [11]: B = np.array([0, 1])

In [12]: K = np.array([4, 4])

In [13]: A_ = A - np.dot(B, K)

In [14]: np.linalg.eig(A_)
Out[14]:
(array([-1., -7.]),
 array([[ 0.70710678,  0.70710678],
        [-0.70710678,  0.70710678]]))
```

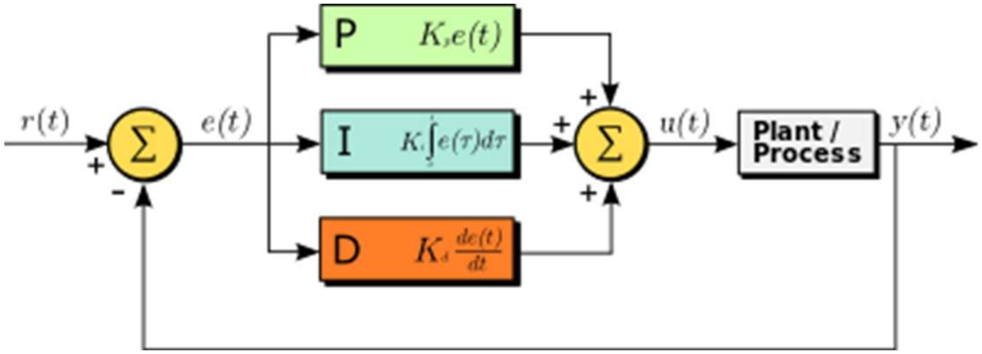
Basic Feedback Control



Closed-loop Control Diagram

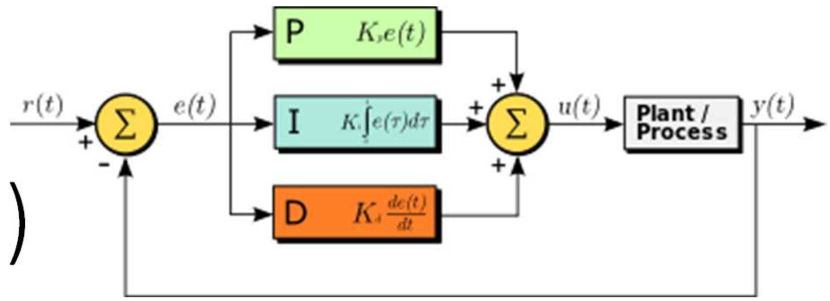
- Is selecting K naively a good strategy?
  - System might be stabilized but it might not be *optimal*
    - Overly stable eigenvalues might use large control inputs
    - The controller can overreact to disturbances and noise
- Solution: Optimal Control
  - Formulate the selection of K as an optimization problem
    - Find a balance between the controller stability and and aggressiveness of control
  - E.g., Linear-Quadratic Controller (LQR)

Proportional-Integral-Derivative Controller (PID)

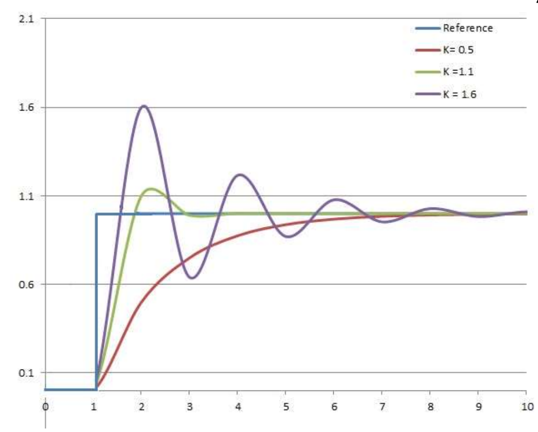


- Goal is to minimize the error term: $e(t) = r(t) - y(t)$
  - $\lim_{t \to \infty} e(t) = 0$
- PID control law: $u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}$

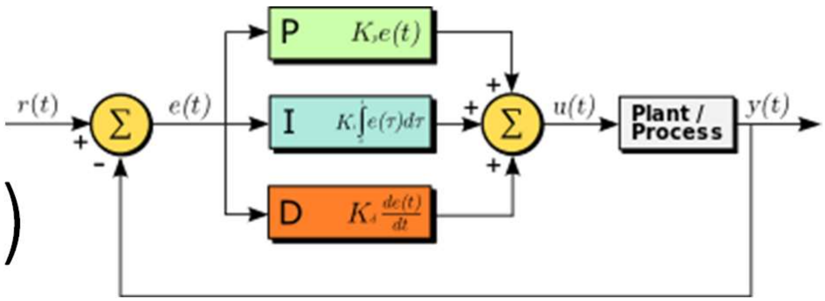
Proportional-Integral-Derivative Controller (PID)



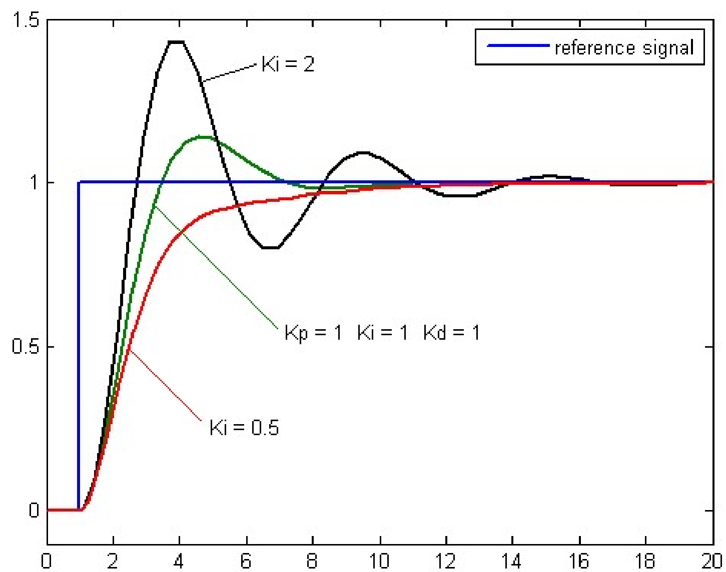
- PID control law: $u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt}$
  - P term: proportional gain $u(t) = K_p e(t)$
    - The control input is proportional to the error.
  - Steady-state error: As the error gets close to 0, small P would not reach the desired state and large P would overshoot.
  - The control input would be 0 at the system at the desired state. Thus, the system would oscillate



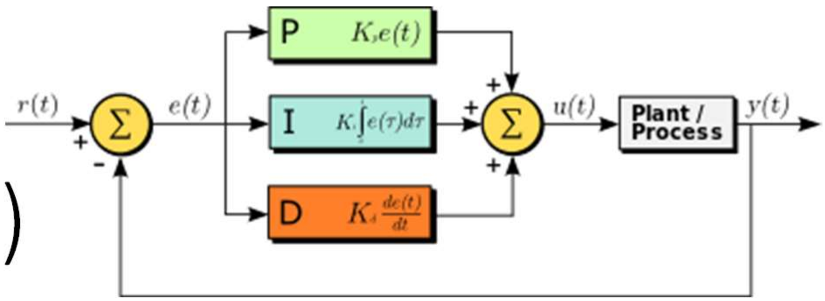
Proportional-Integral-Derivative Controller (PID)



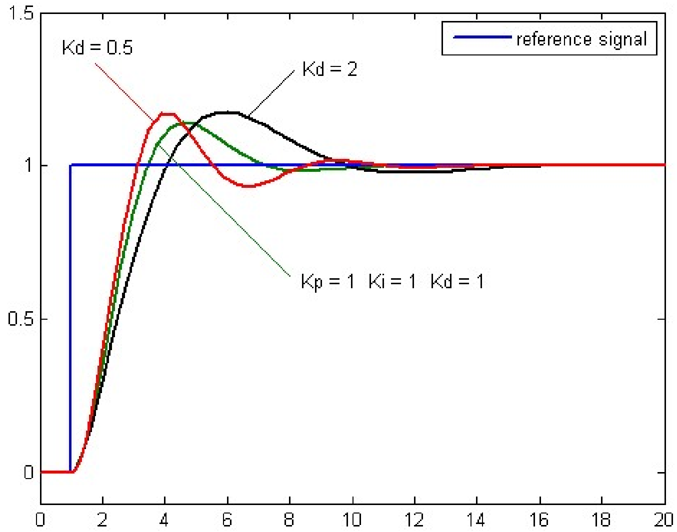
- PID control law: $u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt}$
  - I term: integrator gain $u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau$
    - Uses the history of error accumulation
    - Can overcome the steady-state error



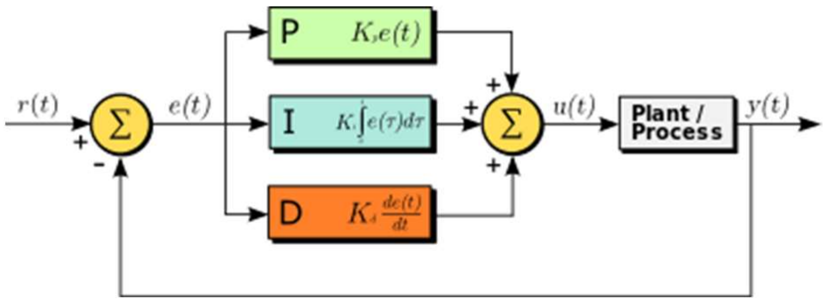
Proportional-Integral-Derivative Controller (PID)

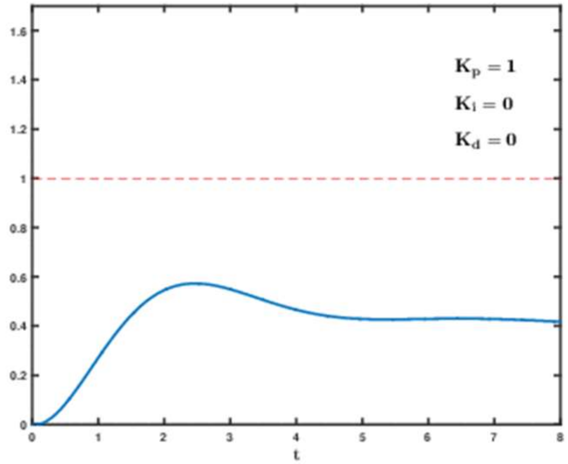


- PID control law: $u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt}$
  - D term: derivative gain $u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt}$
    - Measures the rate of error change
    - Looks at he future



Effects of PID Gains





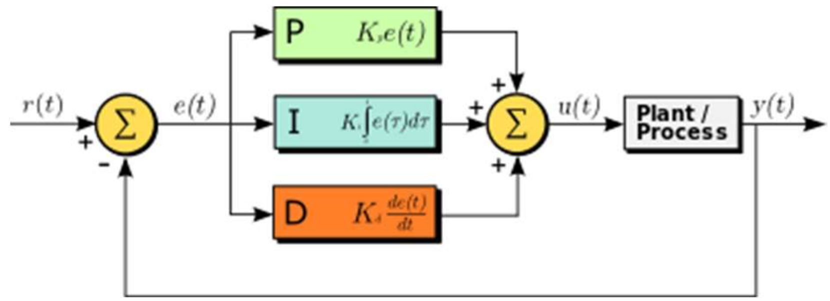
Effects of different PID gains

Effects of *increasing* a parameter independently[22][23]

Parameter	Rise time	Overshoot	Settling time	Steady-state error	Stability
K_p	Decrease	Increase	Small change	Decrease	Degrade
K_i	Decrease	Increase	Increase	Eliminate	Degrade
K_d	Minor change	Decrease	Decrease	No effect in theory	Improve if K_d small

Figures are from Wikipedia: https://en.wikipedia.org/wiki/PID_controller

Tuning PID Gains

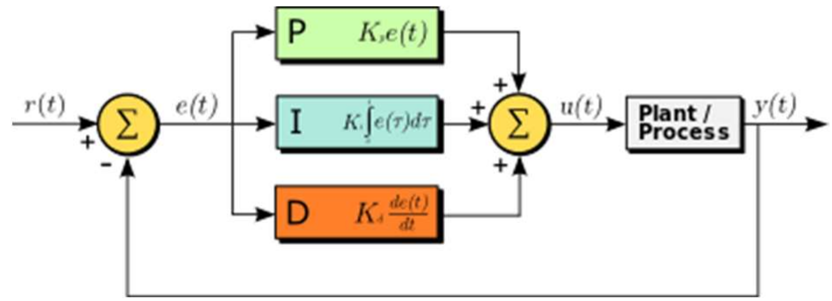


Effects of *increasing* a parameter independently[22][23]

Parameter	Rise time	Overshoot	Settling time	Steady-state error	Stability
K_p	Decrease	Increase	Small change	Decrease	Degrade
K_i	Decrease	Increase	Increase	Eliminate	Degrade
K_d	Minor change	Decrease	Decrease	No effect in theory	Improve if K_d small

- Manual tuning:
  - Set K_i and K_d to 0
  - Increase K_p until the system oscillates
  - Set K_p to half the oscillated value
  - Increase K_i until the steady-state error disappears (if exists)
    - Large K_i can contribute to overshooting
  - If needed, increase K_d until the the system is acceptably quick to reach to desired state

Tuning PID Gains



Effects of *increasing* a parameter independently[22][23]

Parameter	Rise time	Overshoot	Settling time	Steady-state error	Stability
K_p	Decrease	Increase	Small change	Decrease	Degrade
K_i	Decrease	Increase	Increase	Eliminate	Degrade
K_d	Minor change	Decrease	Decrease	No effect in theory	Improve if K_d small

- Ziegler-Nichols method:
    - Set K_i and K_d to 0
    - Increase K_p until the system oscillates K_u
    - Measure the oscillation rate T_u
    - Use the following values to set the gains using T_u and K_u

Ziegler–Nichols method

Control Type	K_p	K_i	K_d
P	$0.50K_u$	–	–
PI	$0.45K_u$	$0.54K_u/T_u$	–
PID	$0.60K_u$	$1.2K_u/T_u$	$3K_uT_u/40$

Tuning PID gains