Lecture 2: Classical Control Theory & Design

Textbook Sections 2.1-2.6

Dr. Jordan D. Larson

Introduction

- Model-based design: use dynamical systems theory to model real-world processes

Introduction

- Model-based design: use dynamical systems theory to model real-world processes
- Simplest model: single input, single output (SISO) and linear, time-invariant (LTI)
  - System to be designed: controller, K
  - System to be controlled: plant, G

Introduction

- Model-based design: use dynamical systems theory to model real-world processes
- Simplest model: single input, single output (SISO) and linear, time-invariant (LTI)
  - System to be designed: controller, K
  - System to be controlled: plant, G
- Dynamical systems: many varieties
  - Flight dynamics: MIMO, nonlinear, time-varying
  - Flight control design: MIMO, linear, time-invariant with robustness to model uncertainty
  - Classical control applies SISO LTI theory and design to multiple loops for aerospace vehicles

Introduction

- Model-based design: use dynamical systems theory to model real-world processes
- Simplest model: single input, single output (SISO) and linear, time-invariant (LTI)
  - System to be designed: controller, K
  - System to be controlled: plant, G
- Dynamical systems: many varieties
  - Flight dynamics: MIMO, nonlinear, time-varying
  - Flight control design: MIMO, linear, time-invariant with robustness to model uncertainty
  - Classical control applies SISO LTI theory and design to multiple loops for aerospace vehicles
- Open- and closed-loop control laws
  - A.k.a. feedforward and feedback, respectively
  - Feedback accounts for disturbances/uncertainties to plant model

Open-Loop Control

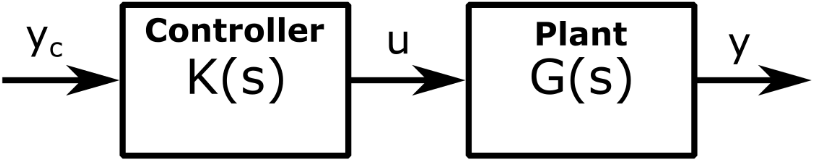
- Simplest types of control laws: **open-loop control** strategies, a.k.a. **feedforward control**

Open-Loop Control

- Simplest types of control laws: **open-loop control** strategies, a.k.a. **feedforward control**
- In general, logic flow:
  1. User specifies commanded output y_c
  2. Control law sets input u as function of commanded output
  3. Dynamically changes output $y(t)$

Open-Loop Control

- Simplest types of control laws: **open-loop control** strategies, a.k.a. **feedforward control**
- In general, logic flow:
  1. User specifies commanded output y_c
  2. Control law sets input u as function of commanded output
  3. Dynamically changes output $y(t)$
- Open-loop control of SISO LTI systems block diagram:



Open-Loop Control: Actuation System

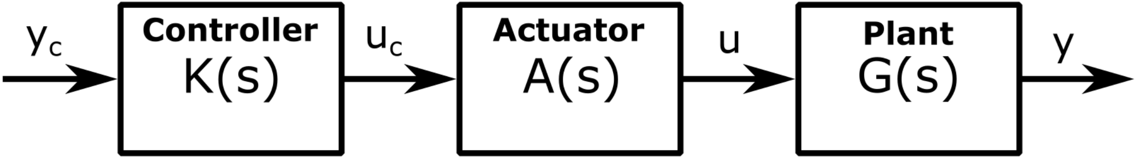
- For real systems, u will undergo some dynamics itself, i.e. plant input won't instantaneously reach electric controller's computed u

Open-Loop Control: Actuation System

- For real systems, u will undergo some dynamics itself, i.e. plant input won't instantaneously reach electric controller's computed u
- Truly actuated using some electro-mechanical interface: represent by actuation system, a.k.a. **actuator**

Open-Loop Control: Actuation System

- For real systems, u will undergo some dynamics itself, i.e. plant input won't instantaneously reach electric controller's computed u
- Truly actuated using some electro-mechanical interface: represent by actuation system, a.k.a. **actuator**
- Additional consideration results in following block diagram for SISO LTI systems



Open-Loop Control Actuation System (continued)

- Using dynamical systems theory, one can typically model actuation systems, $A(s)$, as first- or second-order LTI systems, e.g.

$$A(s) = \frac{\omega_a}{s + \omega_a} \tag{1}$$

Open-Loop Control Actuation System (continued)

- Using dynamical systems theory, one can typically model actuation systems, $A(s)$, as first- or second-order LTI systems, e.g.

$$A(s) = \frac{\omega_a}{s + \omega_a} \tag{1}$$

$$A(s) = \frac{\omega_a^2}{s^2 + 2\zeta_a \omega_a s + \omega_a^2} \tag{2}$$

  - ω_a: bandwidth of actuator
  - ζ_a: damping

Open-Loop Control Actuation System (continued)

- Using dynamical systems theory, one can typically model actuation systems, $A(s)$, as first- or second-order LTI systems, e.g.

$$A(s) = \frac{\omega_a}{s + \omega_a} \tag{1}$$

$$A(s) = \frac{\omega_a^2}{s^2 + 2\zeta_a \omega_a s + \omega_a^2} \tag{2}$$

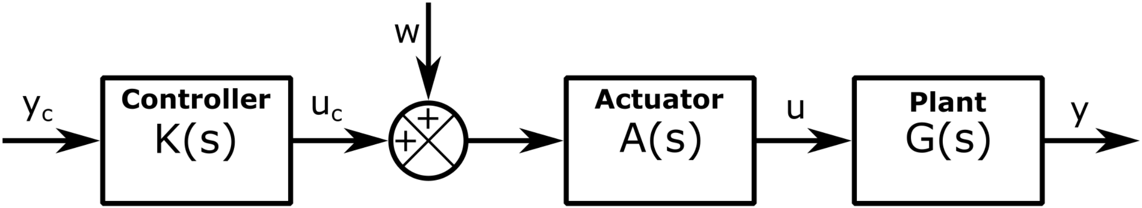
  - ω_a: bandwidth of actuator
  - ζ_a: damping
- Actuators also typically have hard limits on minimum and maximum output as well as hard rate limits

Open-Loop Control: Plant Disturbances

- For real systems will have additional disturbances to plant model
  - Modeled as additive disturbance signal, w, to control input, u, which results in a as

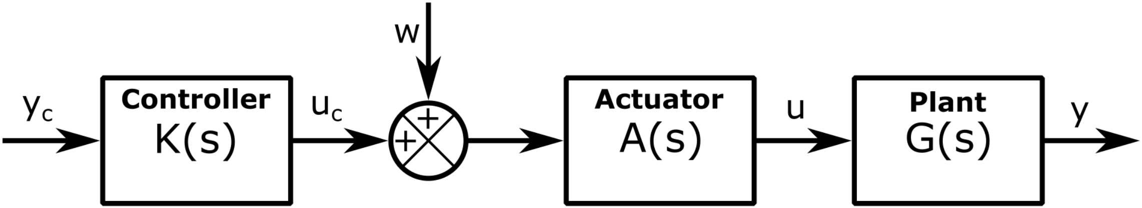
Open-Loop Control: Plant Disturbances

- For real systems will have additional disturbances to plant model
  - Modeled as additive disturbance signal, w, to control input, u, which results in a as
- Block diagram for SISO LTI systems:



Open-Loop Control: Plant Disturbances

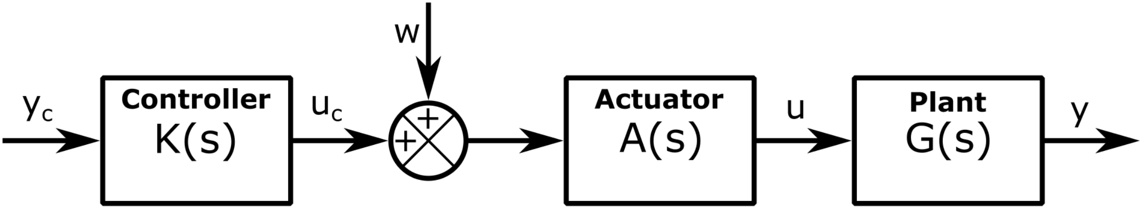
- For real systems will have additional disturbances to plant model
  - Modeled as additive disturbance signal, *w*, to control input, *u*, which results in a as
- Block diagram for SISO LTI systems:



- If no explicit knowledge of disturbances: may not guarantee control system satisfies design requirements using only open-loop control
  - Strategy does not gather any information about disturbances as they occur

Open-Loop Control: Plant Disturbances

- For real systems will have additional disturbances to plant model
  - Modeled as additive disturbance signal, *w*, to control input, *u*, which results in a as
- Block diagram for SISO LTI systems:



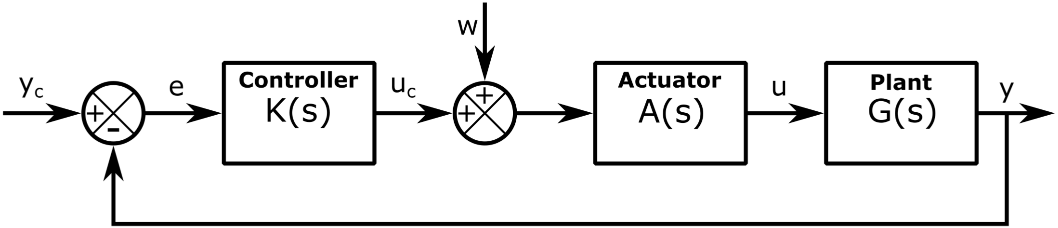
- If no explicit knowledge of disturbances: may not guarantee control system satisfies design requirements using only open-loop control
  - Strategy does not gather any information about disturbances as they occur
- To reject these disturbances and account for system uncertainties: use alternative control strategies

Closed-Loop Control

- **Closed-loop control**, a.k.a. **feedback control**:
  - Feed plant output signal back to control law

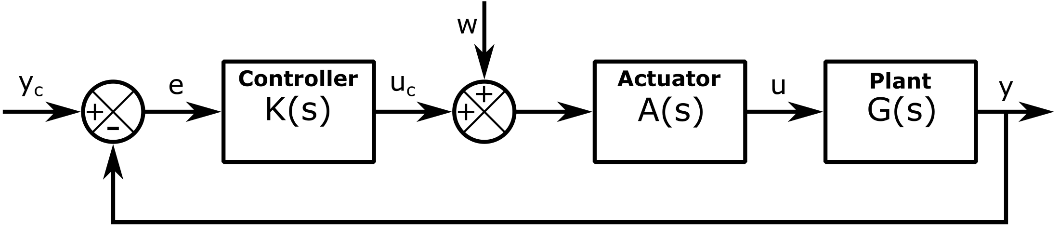
Closed-Loop Control

- **Closed-loop control**, a.k.a. **feedback control**:
  - Feed plant output signal back to control law
- Block diagram for SISO LTI systems:



Closed-Loop Control

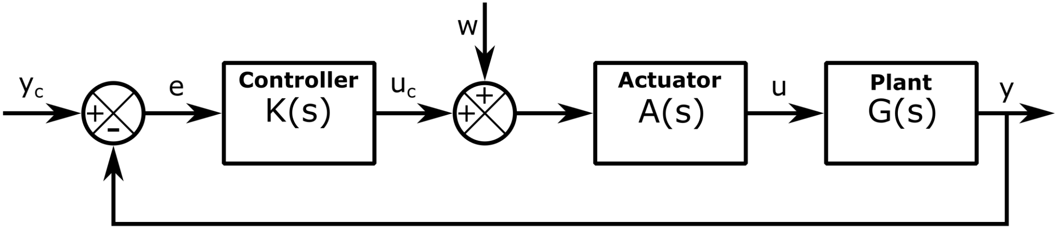
- **Closed-loop control**, a.k.a. **feedback control**:
  - Feed plant output signal back to control law
- Block diagram for SISO LTI systems:



- Use output of plant, $y(t)$, to compute tracking error, $e(t)$, directly affects control input to plant, $u(t)$, in turn, affects plant output and so on, as time continues

Closed-Loop Control

- **Closed-loop control**, a.k.a. **feedback control**:
  - Feed plant output signal back to control law
- Block diagram for SISO LTI systems:



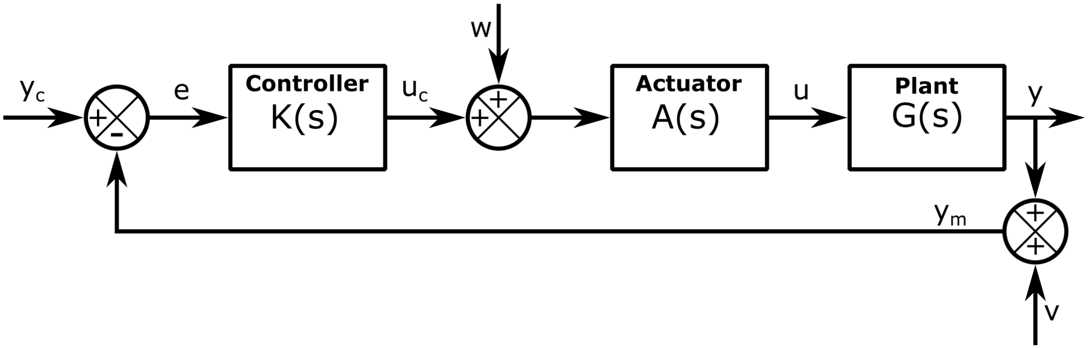
- Use output of plant, $y(t)$, to compute tracking error, $e(t)$, directly affects control input to plant, $u(t)$, in turn, affects plant output and so on, as time continues
- For real systems, output signal measured by sensor system, or **sensor**:
  - Measures output of system using physical phenomena converted into a measured signal
  - Similar to input signal being actuated in real systems

Closed-Loop Control: Sensor System

- No sensor provides perfect information: consider additive noise signal, v, to output, y

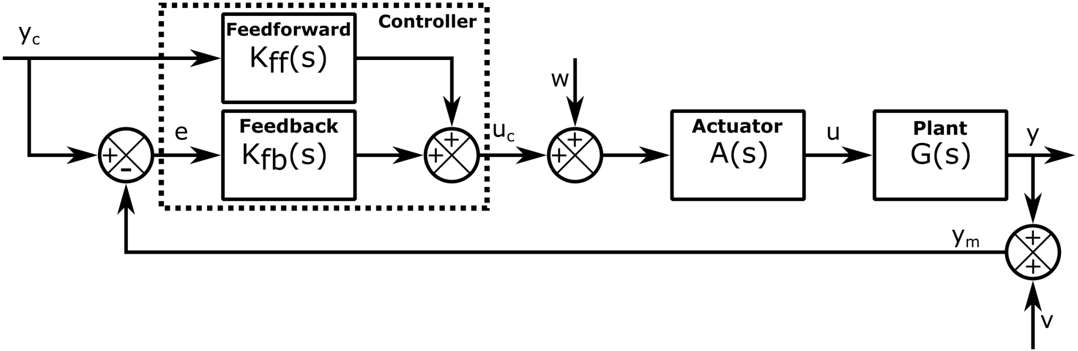
Closed-Loop Control: Sensor System

- No sensor provides perfect information: consider additive noise signal, v, to output, y
- Block diagram for SISO LTI systems:



Open- and Closed-Loop Control

- Feedforward and feedback control combined as block diagram:

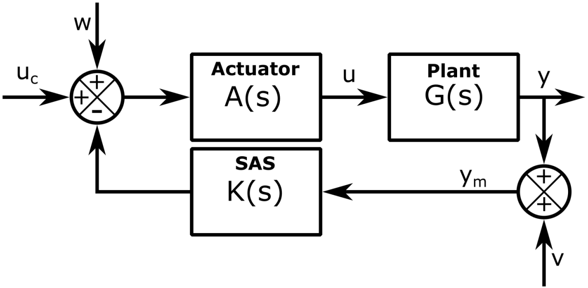


Stability Augmentation Systems

- Special type of feedback control system: **stability augmentation system (SAS)**
  - Plant controlled by external operator, e.g. pilot, but inherent response characteristics, e.g. modal stability or damping, not within design specifications
  - SAS used to “augment” plant dynamics to achieve certain dynamic responses, typically stability or excessive damping

Stability Augmentation Systems

- Special type of feedback control system: **stability augmentation system (SAS)**
  - Plant controlled by external operator, e.g. pilot, but inherent response characteristics, e.g. modal stability or damping, not within design specifications
  - SAS used to "augment" plant dynamics to achieve certain dynamic responses, typically stability or excessive damping
- SAS for SISO LTI systems as block diagram:

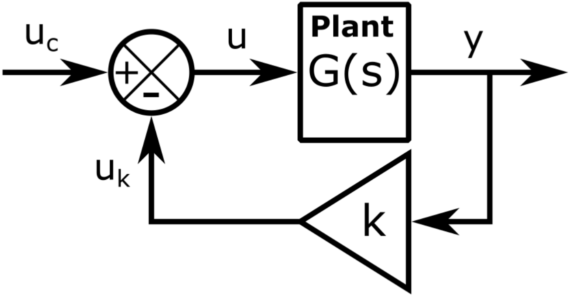


Proportional Stability Augmentation System

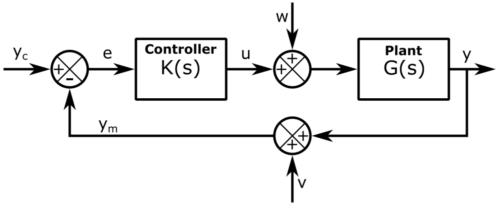
- Classical stability augmentation system design method: root locus method as **proportional SAS (P-SAS)**:

$$K_{P-SAS}(s) = \frac{u_k(s)}{y(s)} = k \tag{3}$$

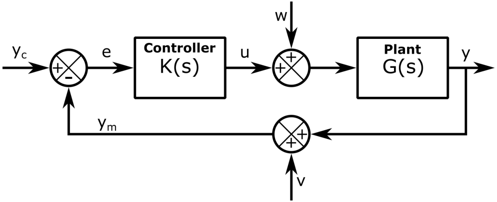
  - $k > 0$: negative feedback gain



Classical Feedback Control System Model

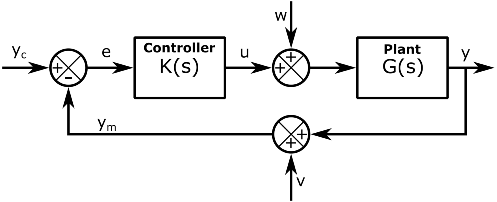


Classical Feedback Control System Model



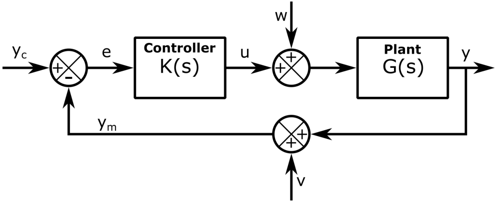
- $G(s)$: actuator-plant-sensor combination

Classical Feedback Control System Model



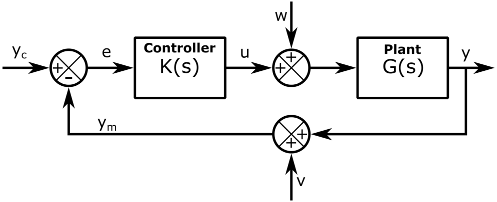
- $G(s)$: actuator-plant-sensor combination
- $K(s)$: controller

Classical Feedback Control System Model



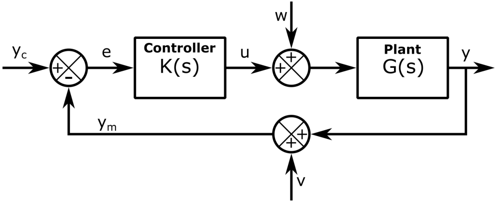
- $G(s)$: actuator-plant-sensor combination
- $K(s)$: controller
- y_c: desired output signal

Classical Feedback Control System Model



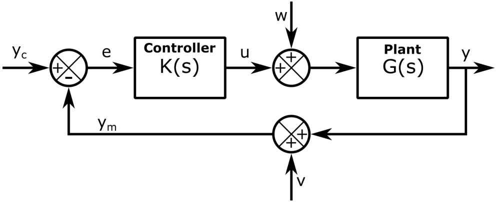
- $G(s)$: actuator-plant-sensor combination
- $K(s)$: controller
- y_c: desired output signal
- e: tracking error signal

Classical Feedback Control System Model



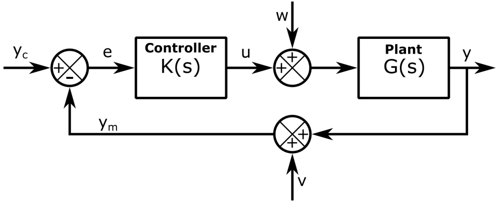
- $G(s)$: actuator-plant-sensor combination
- $K(s)$: controller
- y_c: desired output signal
- e: tracking error signal
- u: control input signal

Classical Feedback Control System Model



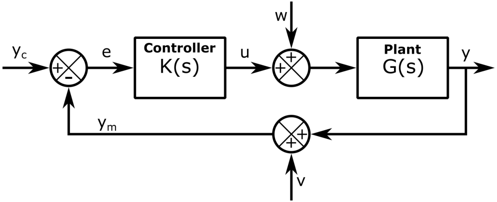
- $G(s)$: actuator-plant-sensor combination
- $K(s)$: controller
- y_c: desired output signal
- e: tracking error signal
- u: control input signal
- w: noise signal to dynamic system input, e.g. disturbance

Classical Feedback Control System Model



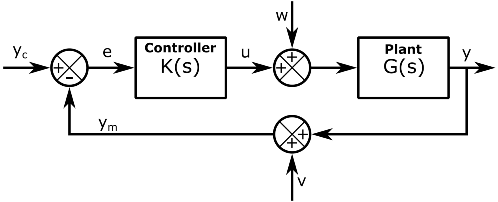
- $G(s)$: actuator-plant-sensor combination
- $K(s)$: controller
- y_c: desired output signal
- e: tracking error signal
- u: control input signal
- w: noise signal to dynamic system input, e.g. disturbance
- y: system output signal

Classical Feedback Control System Model



- $G(s)$: actuator-plant-sensor combination
- $K(s)$: controller
- y_c: desired output signal
- e: tracking error signal
- u: control input signal
- w: noise signal to dynamic system input, e.g. disturbance
- y: system output signal
- v: noise signal on dynamic system output, e.g. sensor noise

Classical Feedback Control System Model



- $G(s)$: actuator-plant-sensor combination
- $K(s)$: controller
- y_c: desired output signal
- e: tracking error signal
- u: control input signal
- w: noise signal to dynamic system input, e.g. disturbance
- y: system output signal
- v: noise signal on dynamic system output, e.g. sensor noise
- y_m: measured output signal

System Transfer Functions

- **Open-loop transfer function**:

$$L(s) = G(s)K(s) \tag{4}$$

  - Plays large role in analysis of SISO LTI control system
  - Note: $L(s)$ control system transfer function if open-loop control design

System Transfer Functions

- **Open-loop transfer function**:

$$L(s) = G(s)K(s) \tag{4}$$

  - Plays large role in analysis of SISO LTI control system
  - Note: $L(s)$ control system transfer function if open-loop control design
- 3 input signals (external/independent): y_c, w, v

System Transfer Functions

- **Open-loop transfer function**:

$$L(s) = G(s)K(s) \tag{4}$$

  - Plays large role in analysis of SISO LTI control system
  - Note: $L(s)$ control system transfer function if open-loop control design
- 3 input signals (external/independent): y_c, w, v
- 4 output signals (internal/dependent): e, u, y, y_m

System Transfer Functions

- **Open-loop transfer function**:

$$L(s) = G(s)K(s) \tag{4}$$

  - Plays large role in analysis of SISO LTI control system
  - Note: $L(s)$ control system transfer function if open-loop control design
- 3 input signals (external/independent): y_c, w, v
- 4 output signals (internal/dependent): e, u, y, y_m
- Input-output pairs have associated transfer function, 12 total:

$$\begin{bmatrix} y(s) \\ y_m(s) \\ e(s) \\ u(s) \end{bmatrix} = \begin{bmatrix} \frac{GK}{1+GK} & \frac{G}{1+GK} & -\frac{GK}{1+GK} \\ \frac{GK}{1+GK} & \frac{G}{1+GK} & \frac{1}{1+GK} \\ \frac{1}{1+GK} & -\frac{G}{1+GK} & -\frac{1}{1+GK} \\ \frac{K}{1+GK} & -\frac{GK}{1+GK} & -\frac{K}{1+GK} \end{bmatrix} \begin{bmatrix} y_c(s) \\ w(s) \\ v(s) \end{bmatrix} \tag{5}$$

Fundamental Transfer Functions

- Four fundamental transfer functions (ignoring sign)

$$\frac{1}{1+G(s)K(s)},\quad \frac{G(s)K(s)}{1+G(s)K(s))},\quad \frac{G(s)}{1+G(s)K(s)},\quad \frac{K(s)}{1+G(s)K(s)} \tag{6}$$

Fundamental Transfer Functions

- Four fundamental transfer functions (ignoring sign)

$$\frac{1}{1+G(s)K(s)}, \quad \frac{G(s)K(s)}{1+G(s)K(s))}, \quad \frac{G(s)}{1+G(s)K(s)}, \quad \frac{K(s)}{1+G(s)K(s)} \tag{6}$$

- Selecting $K(s)$ affects four fundamental transfer functions
  - Study of fundamental transfer functions: key to classical control systems theory

Fundamental Transfer Functions

- Four fundamental transfer functions (ignoring sign)

$$\frac{1}{1+G(s)K(s)}, \quad \frac{G(s)K(s)}{1+G(s)K(s))}, \quad \frac{G(s)}{1+G(s)K(s)}, \quad \frac{K(s)}{1+G(s)K(s)} \tag{6}$$

- Selecting $K(s)$ affects four fundamental transfer functions
  - Study of fundamental transfer functions: key to classical control systems theory
- **Error transfer function**:

$$S(s) = \frac{e(s)}{y_c(s)} = \frac{1}{1+G(s)K(s)} = \frac{1}{1+L(s)} \tag{7}$$

Fundamental Transfer Functions

- Four fundamental transfer functions (ignoring sign)

$$\frac{1}{1+G(s)K(s)}, \quad \frac{G(s)K(s)}{1+G(s)K(s))}, \quad \frac{G(s)}{1+G(s)K(s)}, \quad \frac{K(s)}{1+G(s)K(s)} \tag{6}$$

- Selecting $K(s)$ affects four fundamental transfer functions
  - Study of fundamental transfer functions: key to classical control systems theory
- **Error transfer function**:

$$S(s) = \frac{e(s)}{y_c(s)} = \frac{1}{1+G(s)K(s)} = \frac{1}{1+L(s)} \tag{7}$$

- **Closed-loop transfer function**:

$$T(s) = \frac{y(s)}{y_c(s)} = \frac{G(s)K(s)}{1+G(s)K(s)} = \frac{L(s)}{1+L(s)} \tag{8}$$

Sensitivity Transfer Function

- $S(s)$ a.k.a. **sensitivity transfer function**

Sensitivity Transfer Function

- $S(s)$ a.k.a. **sensitivity transfer function**
- $T(s)$ a.k.a. **complementary sensitivity transfer function**:

$$S(s) + T(s) = \frac{1}{1 + G(s)K(s)} + \frac{G(s)K(s)}{1 + G(s)K(s)} \tag{9}$$

$$S(s) + T(s) = 1 \quad \forall\, s \in \mathbb{C} \tag{10}$$

Aside: P-SAS Root Locus

- P-SAS produces overall system transfer function:

$$y(s) = \left[\frac{G(s)}{1 + kG(s)}\right] u_c(s) \tag{11}$$

Aside: P-SAS Root Locus

- P-SAS produces overall system transfer function:

$$y(s) = \left[\frac{G(s)}{1 + kG(s)}\right] u_c(s) \tag{11}$$

- Closed-loop system poles:

$$d_G(s) + kn_G(s) = 0 \tag{12}$$

  - $n_G(s)$: numerator polynomial of $G(s)$
  - $d_G(s)$: denominator polynomial of $G(s)$

Aside: P-SAS Root Locus

- P-SAS produces overall system transfer function:

$$y(s) = \left[\frac{G(s)}{1 + kG(s)}\right] u_c(s) \tag{11}$$

- Closed-loop system poles:

$$d_G(s) + kn_G(s) = 0 \tag{12}$$

  - $n_G(s)$: numerator polynomial of $G(s)$
  - $d_G(s)$: denominator polynomial of $G(s)$
- Sweeping through values of k from 0 to ∞, alter system poles from $d_G(s)$ to $n_G(s)$ and/or $\pm\infty$
  - Depends on number of zeros relative to number of poles
  - Behavior: **root locus plot** of system poles and zeros as k varies from 0 to ∞

Designing Stable Feedback Control Systems

- Control system design, at minimum, requires stability

Designing Stable Feedback Control Systems

- Control system design, at minimum, requires stability
- For classical feedback control system model: if and only if *all* transfer functions stable

Designing Stable Feedback Control Systems

- Control system design, at minimum, requires stability
- For classical feedback control system model: if and only if *all* transfer functions stable
- Simplify by considering numerator-denominator of $G(s)$ and $K(s)$

$$G(s) = \frac{n_G(s)}{d_G(s)} \tag{13}$$

$$K(s) = \frac{n_K(s)}{d_K(s)} \tag{14}$$

Designing Stable Feedback Control Systems

- Control system design, at minimum, requires stability
- For classical feedback control system model: if and only if *all* transfer functions stable
- Simplify by considering numerator-denominator of $G(s)$ and $K(s)$

$$G(s) = \frac{n_G(s)}{d_G(s)} \tag{13}$$

$$K(s) = \frac{n_K(s)}{d_K(s)} \tag{14}$$

- $d_G d_K + n_G n_K$ denominator for each fundamental transfer function: **SISO feedback control system characteristic polynomial**

Stable Feedback Control System

- **SISO feedback control system characteristic equation:**

$$d_G(s)d_K(s) + n_G(s)n_K(s) = 0 \tag{15}$$

Stable Feedback Control System

- **SISO feedback control system characteristic equation**:

$$d_G(s)d_K(s) + n_G(s)n_K(s) = 0 \tag{15}$$

- Roots/poles in left half of complex plane $\rightarrow$ feedback system stable
  - True even if pole-zero cancellations

Stable Feedback Control System

- **SISO feedback control system characteristic equation**:

$$d_G(s)d_K(s) + n_G(s)n_K(s) = 0 \tag{15}$$

- Roots/poles in left half of complex plane $\rightarrow$ feedback system stable
  - True even if pole-zero cancellations
- Note: $1 + L = 1 + GK$ appears in denominator of each fundamental transfer function

$$1 + L(s) = 1 + G(s)K(s) = 1 + \frac{n_G n_K}{d_G d_k} = \frac{d_G(s)d_K(s) + n_G(s)n_K(s)}{d_G(s)d_K(s)} \tag{16}$$

  - Numerator of $1 + L(s)$: SISO LTI feedback characteristic polynomial

Stable Feedback Control System

- **SISO feedback control system characteristic equation**:

$$d_G(s)d_K(s) + n_G(s)n_K(s) = 0 \tag{15}$$

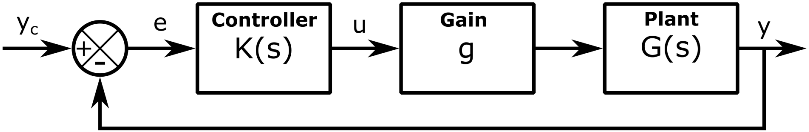
- Roots/poles in left half of complex plane $\rightarrow$ feedback system stable
  - True even if pole-zero cancellations
- Note: $1 + L = 1 + GK$ appears in denominator of each fundamental transfer function

$$1 + L(s) = 1 + G(s)K(s) = 1 + \frac{n_G n_K}{d_G d_k} = \frac{d_G(s)d_K(s) + n_G(s)n_K(s)}{d_G(s)d_K(s)} \tag{16}$$

  - Numerator of $1 + L(s)$: SISO LTI feedback characteristic polynomial
- As $1 + L(s)$ still affected by pole-zero cancellations: **stable feedback control system** if and only if
  1. No RHP pole-zero cancellations when forming $L(s)$
  2. $1 + L(s)$ no zeros in RHP

Critical Gain Model

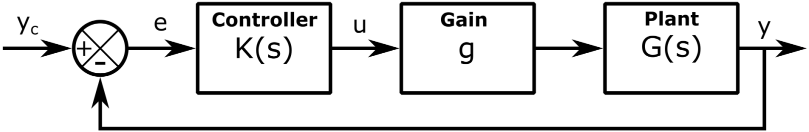
- Consider scaling perturbation to $G(s)$:



- $g > 0$: some scalar constant

Critical Gain Model

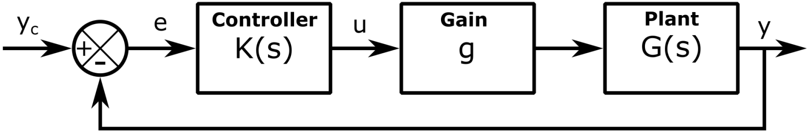
- Consider scaling perturbation to $G(s)$:



- $g > 0$: some scalar constant
- Define perturbed open-loop transfer function as $gL(s)$

Critical Gain Model

- Consider scaling perturbation to $G(s)$:



- $g > 0$: some scalar constant
- Define perturbed open-loop transfer function as $gL(s)$
- Gain margin determined by assessing for what critical values of g feedback control system goes unstable

Critical Gain Computation

- Recall feedback control system stable if
  1. No pole-zero cancellations
  2. Zeros of $1 + L(s) = 0$ only in LHP

Critical Gain Computation

- Recall feedback control system stable if
  1. No pole-zero cancellations
  2. Zeros of $1 + L(s) = 0$ only in LHP
- Critical gain, g_0, occurs when zeros of $1 + g_0 L(j\omega_0)$ on imaginary axis for some critical frequency ω_0, i.e.

$$1 + g_0 L(j\omega_0) = 0 \tag{17}$$

Critical Gain Computation

- Recall feedback control system stable if
  1. No pole-zero cancellations
  2. Zeros of $1 + L(s) = 0$ only in LHP
- Critical gain, g_0, occurs when zeros of $1 + g_0 L(j\omega_0)$ on imaginary axis for some critical frequency ω_0, i.e.

$$1 + g_0 L(j\omega_0) = 0 \tag{17}$$

- Rewriting:

$$L(j\omega_0) = -\frac{1}{g_0} \tag{18}$$

Critical Gain Computation (continued)

- Polar form (i.e. gain and phase):

$$|L(j\omega_0)|e^{j\angle L(j\omega_0)} = -\frac{1}{g_0} \tag{19}$$

Critical Gain Computation (continued)

- Polar form (i.e. gain and phase):

$$|L(j\omega_0)|e^{j\angle L(j\omega_0)} = -\frac{1}{g_0} \tag{19}$$

- Magnitude/gain and phase:

$$g_0 = \frac{1}{|L(j\omega_0)|} \tag{20}$$

$$\angle L(j\omega_0) = \pm 180^\circ \tag{21}$$

Gain Margin Computation

- Identify gain margins by:

Gain Margin Computation

- Identify gain margins by:
  1. Identify all $\omega_{0,i}$ where $\angle L(j\omega_{0,i}) = \pm 180^\circ$

Gain Margin Computation

- Identify gain margins by:
  1. Identify all $\omega_{0,i}$ where $\angle L(j\omega_{0,i}) = \pm 180^\circ$
  2. Calculate all "candidate" $g_{0,i}$ using inverse of gain at $\omega_{0,i}$, i.e.

$$g_{0,i} = \frac{1}{|L(j\omega_{0,i})|} \tag{22}$$

Gain Margin Computation

- Identify gain margins by:
  1. Identify all $\omega_{0,i}$ where $\angle L(j\omega_{0,i}) = \pm 180^\circ$
  2. Calculate all "candidate" $g_{0,i}$ using inverse of gain at $\omega_{0,i}$, i.e.

$$g_{0,i} = \frac{1}{|L(j\omega_{0,i})|} \tag{22}$$

  3. **Upper gain margin**: $\bar{g} = \min_i g_{0,i} > 1$
  4. **Lower gain margin**: $\underline{g} = \max_i g_{0,i} < 1$

Gain Margin Computation

- Identify gain margins by:
  1. Identify all $\omega_{0,i}$ where $\angle L(j\omega_{0,i}) = \pm 180^\circ$
  2. Calculate all "candidate" $g_{0,i}$ using inverse of gain at $\omega_{0,i}$, i.e.

$$g_{0,i} = \frac{1}{|L(j\omega_{0,i})|} \tag{22}$$

  3. **Upper gain margin**: $\bar{g} = \min_i g_{0,i} > 1$
  4. **Lower gain margin**: $\underline{g} = \max_i g_{0,i} < 1$
- Feedback control system stable:

$$\underline{g} \leq g \leq \bar{g} \tag{23}$$

- Note: not all systems have upper *and* lower gain margins

Gain Margin Computation

- Identify gain margins by:
  1. Identify all $\omega_{0,i}$ where $\angle L(j\omega_{0,i}) = \pm 180°$
  2. Calculate all "candidate" $g_{0,i}$ using inverse of gain at $\omega_{0,i}$, i.e.

$$g_{0,i} = \frac{1}{|L(j\omega_{0,i})|} \tag{22}$$

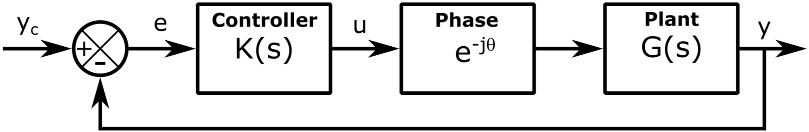
  3. **Upper gain margin**: $\bar{g} = \min_i g_{0,i} > 1$
  4. **Lower gain margin**: $\underline{g} = \max_i g_{0,i} < 1$
- Feedback control system stable:

$$\underline{g} \leq g \leq \bar{g} \tag{23}$$

- Note: not all systems have upper *and* lower gain margins
- FDC: feedback control system typically sufficiently robust if $\underline{g} \leq 0.5$ and $\bar{g} \geq 2$ for gain margin
  - I.e. $\underline{g} \leq -6$ dB and $\bar{g} \geq 6$ dB

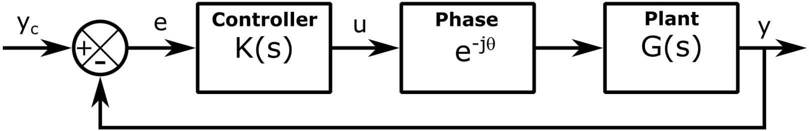
Critical Phase Model

- Consider phase perturbation to $G(s)$



Critical Phase Model

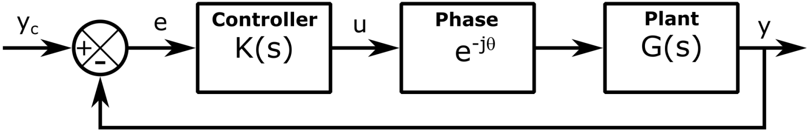
- Consider phase perturbation to $G(s)$



- Define perturbed open-loop transfer function as $e^{-j\theta}L(s)$

Critical Phase Model

- Consider phase perturbation to $G(s)$



- Define perturbed open-loop transfer function as $e^{-j\theta}L(s)$
- Phase margin determined by assessing for what critical values of θ feedback control system goes unstable

Critical Phase Computation

- Recall feedback control system stable if
  1. No pole-zero cancellations
  2. Zeros of $1 + L(s) = 0$ only in LHP

Critical Phase Computation

- Recall feedback control system stable if
  1. No pole-zero cancellations
  2. Zeros of $1 + L(s) = 0$ only in LHP
- Critical phase, θ_0, occurs when zeros of $1 + e^{-j\theta} L(j\omega_0)$ on imaginary axis for some critical frequency ω_0, i.e.

$$1 + e^{-j\theta_0} L(j\omega_0) = 0 \tag{24}$$

Critical Phase Computation

- Recall feedback control system stable if
  1. No pole-zero cancellations
  2. Zeros of $1 + L(s) = 0$ only in LHP
- Critical phase, θ_0, occurs when zeros of $1 + e^{-j\theta} L(j\omega_0)$ on imaginary axis for some critical frequency ω_0, i.e.

$$1 + e^{-j\theta_0} L(j\omega_0) = 0 \tag{24}$$

- Rewriting:

$$L(j\omega_0) = -e^{j\theta_0} \tag{25}$$

Critical Phase Computation (continued)

- Polar form (i.e. gain and phase):

$$|L(j\omega_0)|e^{j\angle L(j\omega_0)} = -e^{j\theta_0} \tag{26}$$

Critical Phase Computation (continued)

- Polar form (i.e. gain and phase):

$$|L(j\omega_0)|e^{j\angle L(j\omega_0)} = -e^{j\theta_0} \tag{26}$$

- Magnitude/gain and phase:

$$|L(j\omega_0)| = 1 \tag{27}$$

$$\theta_0 = \pm 180^\circ + \angle L(j\omega_0) \tag{28}$$

Critical Phase Computation (continued)

- Polar form (i.e. gain and phase):

$$|L(j\omega_0)|e^{j\angle L(j\omega_0)} = -e^{j\theta_0} \tag{26}$$

- Magnitude/gain and phase:

$$|L(j\omega_0)| = 1 \tag{27}$$

$$\theta_0 = \pm 180^\circ + \angle L(j\omega_0) \tag{28}$$

- θ_0 typically represented as positive number

Critical Phase Computation (continued)

- Polar form (i.e. gain and phase):

$$|L(j\omega_0)|e^{j\angle L(j\omega_0)} = -e^{j\theta_0} \tag{26}$$

- Magnitude/gain and phase:

$$|L(j\omega_0)| = 1 \tag{27}$$

$$\theta_0 = \pm 180^\circ + \angle L(j\omega_0) \tag{28}$$

- θ_0 typically represented as positive number
- Phase margin referenced to either ± 180 and symmetric for positive and negative θ

Phase Margin Computation

- Identify phase margin by:

Phase Margin Computation

- Identify phase margin by:
  1. Identify $\omega_{0,i}$ where $|L(j\omega_{0,i})| = 1$

Phase Margin Computation

- Identify phase margin by:
  1. Identify $\omega_{0,i}$ where $|L(j\omega_{0,i})| = 1$
  2. Calculate "candidate" $\theta_{0,i}$ using distance from 180°, i.e.

$$\theta_{0,i} = \pm 180^\circ + \angle L(j\omega_0) \tag{29}$$

Phase Margin Computation

- Identify phase margin by:
  1. Identify $\omega_{0,i}$ where $|L(j\omega_{0,i})| = 1$
  2. Calculate "candidate" $\theta_{0,i}$ using distance from 180°, i.e.

$$\theta_{0,i} = \pm 180^\circ + \angle L(j\omega_0) \tag{29}$$

  3. Phase margin, $\bar{\theta} = \min_i \; |\theta_{0,i}|$

Phase Margin Computation

- Identify phase margin by:
  1. Identify $\omega_{0,i}$ where $|L(j\omega_{0,i})| = 1$
  2. Calculate "candidate" $\theta_{0,i}$ using distance from 180°, i.e.

$$\theta_{0,i} = \pm 180^\circ + \angle L(j\omega_0) \tag{29}$$

  3. Phase margin, $\bar{\theta} = \min_i |\theta_{0,i}|$
- Feedback control system stable:

$$-\bar{\theta} \leq \theta \leq \bar{\theta} \tag{30}$$

Phase Margin Computation

- Identify phase margin by:
  1. Identify $\omega_{0,i}$ where $|L(j\omega_{0,i})| = 1$
  2. Calculate “candidate” $\theta_{0,i}$ using distance from 180°, i.e.

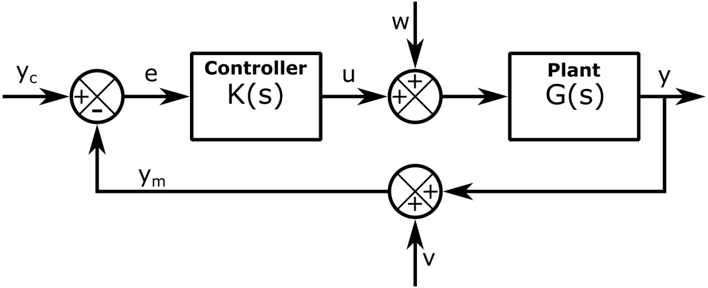
$$\theta_{0,i} = \pm 180^\circ + \angle L(j\omega_0) \tag{29}$$

  3. Phase margin, $\bar{\theta} = \min_i |\theta_{0,i}|$
- Feedback control system stable:

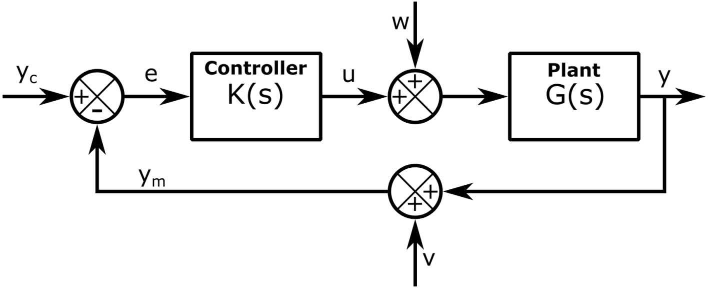
$$-\bar{\theta} \leq \theta \leq \bar{\theta} \tag{30}$$

- FDC: feedback control system typically sufficiently robust if $\bar{\theta} \geq 45^\circ$ for phase margin

Feedback Control System

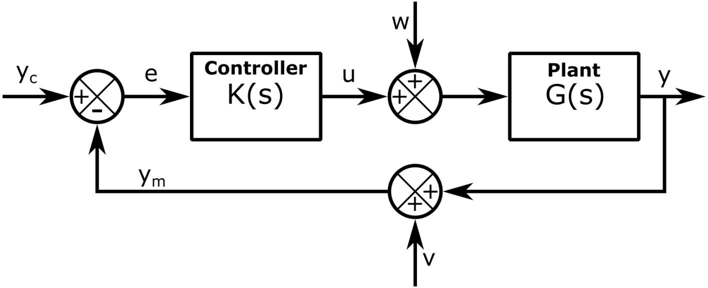


Feedback Control System



- Synthesize $K(s)$ to achieve design requirements:
  1. Stable with good stability margins

Feedback Control System



- Synthesize $K(s)$ to achieve design requirements:
  1. Stable with good stability margins
  2. Good tracking, i.e. transfer function from $y_c \to e$ small
  3. Disturbance rejection, i.e. transfer function from $w \to y$ small
  4. Sensor noise filtering, i.e. transfer function from $v \to e$ small
  5. Control effort realistic, i.e. $|u|$ not too large
- Performance requirements: 2-5

Loop-Shaping to Satisfy Requirements

- **Loop-shaping** control design for synthesizing $K(s)$ for SISO systems:
  - "Shape" SISO open-loop transfer function, $L(s) = G(s)K(s)$, to have certain frequency response characteristics

Loop-Shaping to Satisfy Requirements

- **Loop-shaping** control design for synthesizing $K(s)$ for SISO systems:
  - "Shape" SISO open-loop transfer function, $L(s) = G(s)K(s)$, to have certain frequency response characteristics
- Primary aspect of loop-shaping: $K(s)$ has additive effect on magnitude subplot of Bode plot when considered in decibels:

$$20\log_{10}|L(j\omega)| = 20\log_{10}|G(j\omega)K(j\omega)| = 20\log_{10}|G(j\omega)| + 20\log_{10}|K(j\omega)| \tag{31}$$

Loop-Shaping to Satisfy Requirements

- **Loop-shaping** control design for synthesizing $K(s)$ for SISO systems:
  - "Shape" SISO open-loop transfer function, $L(s) = G(s)K(s)$, to have certain frequency response characteristics
- Primary aspect of loop-shaping: $K(s)$ has additive effect on magnitude subplot of Bode plot when considered in decibels:

$$20\log_{10}|L(j\omega)| = 20\log_{10}|G(j\omega)K(j\omega)| = 20\log_{10}|G(j\omega)| + 20\log_{10}|K(j\omega)| \quad (31)$$

  - To form suitable $L(s)$, use multiple **control stages** for different regions of Bode plot using this additive property for shaping $L(s)$
  - Multiplying each stage together, i.e. using each stage *in series*: provides full controller transfer function $K(s)$

$L(s)$ for Stability Requirement

- If open-loop transfer function, $L(s)$, satisfies following:
  1. No poles or zeros in RHP
  2. $|L(0)| > 0$
  3. Single gain crossover frequency ω_c
  4. $|L(j\omega)|_{\text{dB}}/\text{decade} \geq -30$ dB/decade for $\frac{\omega_c}{\sqrt{10}} < \omega < \sqrt{10}\omega_c$
  5. $|L|_{\text{dB}} \geq 6$ dB for $\omega \leq \frac{\omega_c}{\sqrt{10}}$
  6. $|L|_{\text{dB}} \leq -6$ dB for $\omega \geq \sqrt{10}\omega_c$

  Then, confidently claim feedback control system achieves stability requirements

Performance Requirements Summary

- Recall $S + T = 1$: one cannot simultaneously satisfy all design requirements

Performance Requirements Summary

- Recall $S + T = 1$: one cannot simultaneously satisfy all design requirements
- *Key idea* in frequency domain control design: output commands designed to be low frequency signals relative to any high frequency noise signals on actual output

Performance Requirements Summary

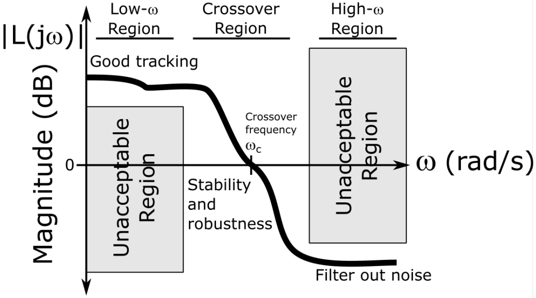
- Recall $S + T = 1$: one cannot simultaneously satisfy all design requirements
- *Key idea* in frequency domain control design: output commands designed to be low frequency signals relative to any high frequency noise signals on actual output
- Requires feedback control system satisfies:
  1. $|S(j\omega)| \ll 1$ at low ω
  2. $|T(j\omega)| \ll 1$ at high ω
  3. $|K(j\omega)S(j\omega)| \ll 1 \;\; \forall\, \omega$

Performance Requirements Summary

- Recall $S + T = 1$: one cannot simultaneously satisfy all design requirements
- *Key idea* in frequency domain control design: output commands designed to be low frequency signals relative to any high frequency noise signals on actual output
- Requires feedback control system satisfies:
  1. $|S(j\omega)| \ll 1$ at low ω
  2. $|T(j\omega)| \ll 1$ at high ω
  3. $|K(j\omega)S(j\omega)| \ll 1 \ \ \forall \ \omega$
- Translating 1 and 2 to open-loop transfer function, $L(j\omega)$, provides equivalent loop-shaping performance requirements:
  1. $|L(j\omega)| \gg 1$ at low ω
  2. $|L(j\omega)| \ll 1$ at high ω
  3. $\left|\frac{K(j\omega)}{1+L(j\omega)}\right| \ll 1 \ \ \forall \ \omega$

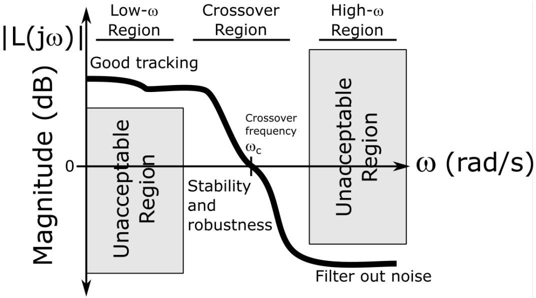
Loop-Shaping Visualization

- Analyzed visually using magnitude subplot of Bode plot of $L(j\omega)$



Loop-Shaping Visualization

- Analyzed visually using magnitude subplot of Bode plot of $L(j\omega)$



- For control effort requirement: analyze Bode plot of $K(j\omega)S(j\omega)$ in parallel with loop-shaping of $L(s)$
  - Typically design $K(j\omega)$ not too large where $G(j\omega)$ small
  - Similar to high frequency requirement for sensor noise filtering

Loop-Shaping Control Stages

- Loop-shaping uses multiple N **control stages** to shape different frequency regions of $L(s)$

Loop-Shaping Control Stages

- Loop-shaping uses multiple N **control stages** to shape different frequency regions of $L(s)$
- Design controllers in *series form*:

$$K(s) = K_1(s) \cdots K_N(s) \tag{32}$$

  - Additive in Bode plot: primary analysis tool
  - Each stage in series $\rightarrow$ multiplying each together provides full controller, $K(s)$

Loop-Shaping Control Stages

- Loop-shaping uses multiple N **control stages** to shape different frequency regions of $L(s)$
- Design controllers in *series form*:

$$K(s) = K_1(s) \cdots K_N(s) \tag{32}$$

  - Additive in Bode plot: primary analysis tool
  - Each stage in series $\rightarrow$ multiplying each together provides full controller, $K(s)$
- Four typical control stages:
  1. Proportional
  2. Integral
  3. Lead
  4. Low-Pass Filter

Loop-Shaping Control Stages

- Loop-shaping uses multiple N **control stages** to shape different frequency regions of $L(s)$
- Design controllers in *series form*:

$$K(s) = K_1(s) \cdots K_N(s) \tag{32}$$

  - Additive in Bode plot: primary analysis tool
  - Each stage in series $\rightarrow$ multiplying each together provides full controller, $K(s)$
- Four typical control stages:
  1. Proportional
  2. Integral
  3. Lead
  4. Low-Pass Filter
- Typically least number of stages, i.e., less complexity, preferred by control designers

General SISO Loop-Shaping Design

- Consider following iterative procedure for **loop-shaping design** of $K(s)$ until all design requirements satisfied

General SISO Loop-Shaping Design

- Consider following iterative procedure for **loop-shaping design** of $K(s)$ until all design requirements satisfied
  1. Use proportional gain control stage to set loop bandwidth, ω_c
     - May need to obtain requirements on ω_c from t_s

General SISO Loop-Shaping Design

- Consider following iterative procedure for **loop-shaping design** of $K(s)$ until all design requirements satisfied
  1. Use proportional gain control stage to set loop bandwidth, ω_c
     - May need to obtain requirements on ω_c from t_s
  2. Use lead control stage(s) to reduce $|L(j\omega)|$ slope about ω_c to > -30 dB/10ω for stability and robustness

General SISO Loop-Shaping Design

- Consider following iterative procedure for **loop-shaping design** of $K(s)$ until all design requirements satisfied
  1. Use proportional gain control stage to set loop bandwidth, ω_c
     - May need to obtain requirements on ω_c from t_s
  2. Use lead control stage(s) to reduce $|L(j\omega)|$ slope about ω_c to > -30 dB/10ω for stability and robustness
  3. Use integral control stage(s) to increase $|L(j\omega)|$ at low ω, i.e. good tracking
     - May need to obtain requirements on $L(j\omega)$ from $S(j\omega)$ or e_{ss}

General SISO Loop-Shaping Design

- Consider following iterative procedure for **loop-shaping design** of $K(s)$ until all design requirements satisfied
  1. Use proportional gain control stage to set loop bandwidth, ω_c
     - May need to obtain requirements on ω_c from t_s
  2. Use lead control stage(s) to reduce $|L(j\omega)|$ slope about ω_c to > -30 dB/10ω for stability and robustness
  3. Use integral control stage(s) to increase $|L(j\omega)|$ at low ω, i.e. good tracking
     - May need to obtain requirements on $L(j\omega)$ from $S(j\omega)$ or e_{ss}
  4. Use low-pass filter stage(s) to decrease $|L(j\omega)|$ at high ω, i.e. good noise filtering
     - May need to obtain requirements on $L(j\omega)$ from $T(j\omega)$

General SISO Loop-Shaping Design

- Consider following iterative procedure for **loop-shaping design** of $K(s)$ until all design requirements satisfied
  1. Use proportional gain control stage to set loop bandwidth, ω_c
     - May need to obtain requirements on ω_c from t_s
  2. Use lead control stage(s) to reduce $|L(j\omega)|$ slope about ω_c to > -30 dB/10ω for stability and robustness
  3. Use integral control stage(s) to increase $|L(j\omega)|$ at low ω, i.e. good tracking
     - May need to obtain requirements on $L(j\omega)$ from $S(j\omega)$ or e_{ss}
  4. Use low-pass filter stage(s) to decrease $|L(j\omega)|$ at high ω, i.e. good noise filtering
     - May need to obtain requirements on $L(j\omega)$ from $T(j\omega)$
  5. Potentially iterate:
     - Integral and filter stages may slightly affect ω_c

PID Parallel Form

- Most common classical control design method: **proportional-integral-derivative (PID)** control law defined in **parallel form**:

$$K_{PID}(s) = \frac{u(s)}{e(s)} = K_p + \frac{K_i}{s} + K_d s \tag{33}$$

  - K_p: **proportional gain**
  - K_i: **integral gain**
  - K_d: **derivative gain**

PID Parallel Form

- Most common classical control design method: **proportional-integral-derivative (PID)** control law defined in **parallel form**:

$$K_{PID}(s) = \frac{u(s)}{e(s)} = K_p + \frac{K_i}{s} + K_d s \tag{33}$$

  - K_p: **proportional gain**
  - K_i: **integral gain**
  - K_d: **derivative gain**
- If $K_d = 0$: proportional-integral (PI) controller

PID Parallel Form

- Most common classical control design method: **proportional-integral-derivative (PID)** control law defined in **parallel form**:

$$K_{PID}(s) = \frac{u(s)}{e(s)} = K_p + \frac{K_i}{s} + K_d s \tag{33}$$

  - K_p: **proportional gain**
  - K_i: **integral gain**
  - K_d: **derivative gain**
- If $K_d = 0$: proportional-integral (PI) controller
- If $K_i = 0$: proportional-derivative (PD) controller

PID Parallel Form

- Most common classical control design method: **proportional-integral-derivative (PID)** control law defined in **parallel form**:

$$K_{PID}(s) = \frac{u(s)}{e(s)} = K_p + \frac{K_i}{s} + K_d s \tag{33}$$

  - K_p: **proportional gain**
  - K_i: **integral gain**
  - K_d: **derivative gain**
- If $K_d = 0$: proportional-integral (PI) controller
- If $K_i = 0$: proportional-derivative (PD) controller
- If $K_d = K_i = 0$: simple proportional (P) controller

Alternative Forms of PID Controller

- For PID control design via loop-shaping: **series form**, a.k.a. **interacting form**, of PID controller

$$K_{PID}(s) = k\left(\frac{s+\omega_i}{s}\right)\left(\frac{1}{\omega_d}s+1\right) \tag{34}$$

  - k: overall gain
  - ω_i: integral frequency
  - ω_d: derivative frequency

Filtered Derivative PID Control

- Noting derivative boost stage: derivative term increases asymptotic slope of open-loop transfer function, $L(s)$
  - Amplifies any high frequency sensor noise in y_m in classical feedback control system

Filtered Derivative PID Control

- Noting derivative boost stage: derivative term increases asymptotic slope of open-loop transfer function, $L(s)$
  - Amplifies any high frequency sensor noise in y_m in classical feedback control system
- To overcome unfavorable outcome, often derivative term in PID control alternatively implemented using **filtered derivative**

Filtered Derivative PID Control

- Noting derivative boost stage: derivative term increases asymptotic slope of open-loop transfer function, $L(s)$
  - Amplifies any high frequency sensor noise in y_m in classical feedback control system
- To overcome unfavorable outcome, often derivative term in PID control alternatively implemented using **filtered derivative**
- PID **filtered parallel form**:

$$K_{PID'}(s) = K_p + \frac{K_i}{s} + K_d \frac{\omega_\infty}{1 + \omega_\infty \frac{1}{s}} = K_p + \frac{K_i}{s} + K_d \frac{s}{\frac{1}{\omega_\infty} s + 1} \tag{35}$$

Filtered Derivative PID Control

- Noting derivative boost stage: derivative term increases asymptotic slope of open-loop transfer function, $L(s)$
  - Amplifies any high frequency sensor noise in y_m in classical feedback control system
- To overcome unfavorable outcome, often derivative term in PID control alternatively implemented using **filtered derivative**
- PID **filtered parallel form**:

$$K_{PID'}(s) = K_p + \frac{K_i}{s} + K_d \frac{\omega_\infty}{1 + \omega_\infty \frac{1}{s}} = K_p + \frac{K_i}{s} + K_d \frac{s}{\frac{1}{\omega_\infty} s + 1} \tag{35}$$

- PID **filtered series form**:

$$K_{PID'}(s) = k \left(\frac{s + \omega_i}{s} \right) \left(\frac{\beta^2 s + \omega_\infty}{s + \omega_\infty} \right) \tag{36}$$

PID Control via Loop-shaping Control Stages

- Equivalently, by substitutions $\omega_\infty = \beta\omega_c$:

$$K_{PID'}(s) = k\beta\left(\frac{s+\omega_i}{s}\right)\left(\frac{\beta s+\omega_c}{s+\beta\omega_c}\right) \tag{37}$$

PID Control via Loop-shaping Control Stages

- Equivalently, by substitutions $\omega_\infty = \beta\omega_c$:

$$K_{PID'}(s) = k\beta\left(\frac{s+\omega_i}{s}\right)\left(\frac{\beta s+\omega_c}{s+\beta\omega_c}\right) \tag{37}$$

- Demonstrates filtered PID control has particular loop-shaping controller employing three control stages:
  - Proportional gain of $k\beta$
  - Integral boost at ω_i
  - Lead stage at β and $\omega_c = \frac{\omega_\infty}{\beta}$

PID Control via Loop-shaping Control Stages

- Equivalently, by substitutions $\omega_\infty = \beta\omega_c$:

$$K_{PID'}(s) = k\beta\left(\frac{s+\omega_i}{s}\right)\left(\frac{\beta s+\omega_c}{s+\beta\omega_c}\right) \tag{37}$$

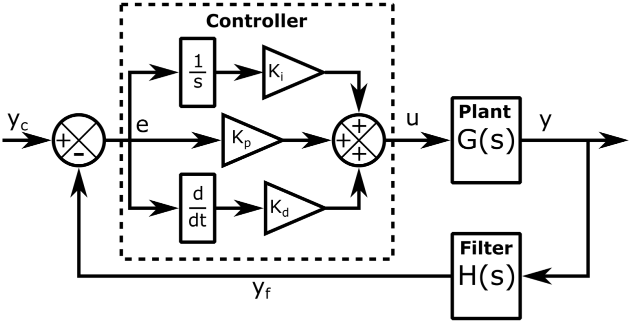
- Demonstrates filtered PID control has particular loop-shaping controller employing three control stages:
  - Proportional gain of $k\beta$
  - Integral boost at ω_i
  - Lead stage at β and $\omega_c = \frac{\omega_\infty}{\beta}$
- Thus, stages allow loop-shaping design procedure to iterate on values to affect low frequency and crossover regions of the $L(j\omega)$
  - Also implement as classical parallel PID control gains

PID Control with Filter Subsystem

- Note: PID controller does not inherently use any low-pass filter stages
  - Many need to use low-pass filter stage in addition to PID controller

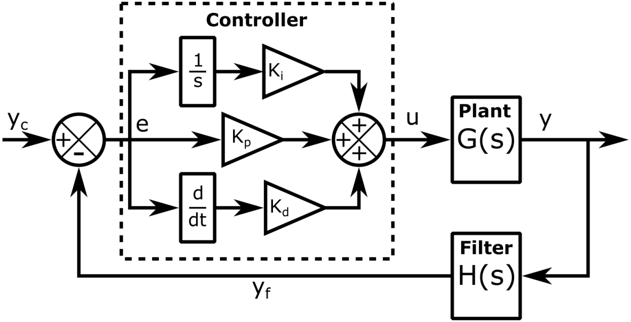
PID Control with Filter Subsystem

- Note: PID controller does not inherently use any low-pass filter stages
  - Many need to use low-pass filter stage in addition to PID controller
  - Often part of filter subsystem for y_m



PID Control with Filter Subsystem

- Note: PID controller does not inherently use any low-pass filter stages
  - Many need to use low-pass filter stage in addition to PID controller
  - Often part of filter subsystem for y_m



- Digital system typically limits update rate, i.e. frequency, of y_c upstream of control system

PI Control with Rate Feedback

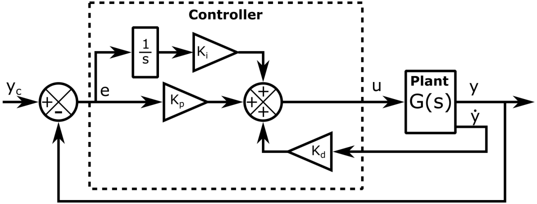
- Another alternative: measure output derivative directly instead of computing derivative of tracking error signal

PI Control with Rate Feedback

- Another alternative: measure output derivative directly instead of computing derivative of tracking error signal
- Derivative term in PID control alternatively implemented **rate feedback control**: derivative of output, i.e. “rate,” fed back

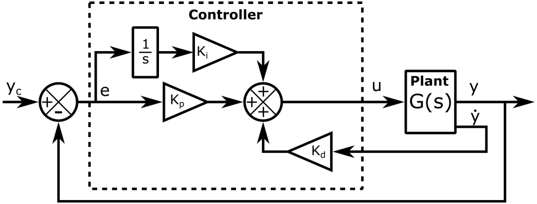
PI Control with Rate Feedback

- Another alternative: measure output derivative directly instead of computing derivative of tracking error signal
- Derivative term in PID control alternatively implemented **rate feedback control**: derivative of output, i.e. “rate,” fed back
- **PI control with rate feedback** block diagram:



PI Control with Rate Feedback

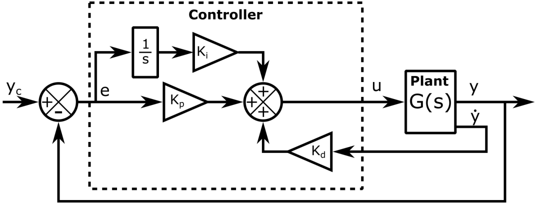
- Another alternative: measure output derivative directly instead of computing derivative of tracking error signal
- Derivative term in PID control alternatively implemented **rate feedback control**: derivative of output, i.e. "rate," fed back
- **PI control with rate feedback** block diagram:



- Considered SIMO feedback control system: two separate outputs fed back to controller: error and output rate

PI Control with Rate Feedback

- Another alternative: measure output derivative directly instead of computing derivative of tracking error signal
- Derivative term in PID control alternatively implemented **rate feedback control**: derivative of output, i.e. "rate," fed back
- **PI control with rate feedback** block diagram:



- Considered SIMO feedback control system: two separate outputs fed back to controller: error and output rate
- Also considered **stability augmentation system (SAS)** using rate, second closed-loop PI controller using output error

Summary

- SISO LTI "Classical" Feedback Control System
  - Negative feedback to cascaded controller, K, and plant, G
  - Characterized by four fundmental transfer functions
  - Open-loop transfer function: $L = GK$

Summary

- SISO LTI "Classical" Feedback Control System
  - Negative feedback to cascaded controller, K, and plant, G
  - Characterized by four fundmental transfer functions
  - Open-loop transfer function: $L = GK$
- SISO LTI System: Stability and Robustness
  - Numerator of $1 + L(s)$: SISO LTI feedback characteristic polynomial
  - Classical margins: gain or phase when $1 + L(s)$ zeros $\rightarrow$ RHP

Summary

- SISO LTI "Classical" Feedback Control System
  - Negative feedback to cascaded controller, K, and plant, G
  - Characterized by four fundmental transfer functions
  - Open-loop transfer function: $L = GK$
- SISO LTI System: Stability and Robustness
  - Numerator of $1 + L(s)$: SISO LTI feedback characteristic polynomial
  - Classical margins: gain or phase when $1 + L(s)$ zeros $\rightarrow$ RHP
- Loop-Shaping Control Design
  - Output commands: low frequency signals & output noise signals: high frequency
  - Shape low frequency tracking, stability margins, high frequency suppression
  - Stages: Proportional Gain, Lead, Integral/Low Frequency Boost, Rolloff

Summary

- SISO LTI "Classical" Feedback Control System
  - Negative feedback to cascaded controller, K, and plant, G
  - Characterized by four fundmental transfer functions
  - Open-loop transfer function: $L = GK$
- SISO LTI System: Stability and Robustness
  - Numerator of $1 + L(s)$: SISO LTI feedback characteristic polynomial
  - Classical margins: gain or phase when $1 + L(s)$ zeros $\to$ RHP
- Loop-Shaping Control Design
  - Output commands: low frequency signals & output noise signals: high frequency
  - Shape low frequency tracking, stability margins, high frequency suppression
  - Stages: Proportional Gain, Lead, Integral/Low Frequency Boost, Rolloff
- PID Control Design
  - Uses 3-stages in parallel or series
  - Tuning using time- or frequency-domain techniques