ECE 486: Control Systems

Lecture 10A: Dominant Pole Approximation
Dominant-Pole Approximation

The dominant poles of a higher-order system are the slowest poles (largest time constant).

We can often approximate a higher-order system by a:

1. First-order approximation if the dominant pole is real
2. Second-order approximation if the dominant pole(s) are a complex pair.

The approximation is accurate if the dominant pole(s) are significantly slower than the remaining poles.

(Dominant pole time constant is 5x larger than other poles)
Problem 1

For each system:
• Construct a first-order or second-order approximation from the dominant pole.
• Do you expect the dominant pole approximation to be accurate?

For system A: Roughly sketch the unit step response of the dominant pole approximation. Note the final time, settling time, and overshoot (if underdamped).

\[ G_A(s) = \frac{5000}{(s+2)(s+20)(s^2+20s+500)} \]

\[ G_B(s) = \frac{24}{(s+1)(s+2)^2(s+3)} \]
Solution 1A

- Construct a first-order or second-order approximation from the dominant pole.
- Do you expect the dominant pole approximation to be accurate?
- Roughly sketch the unit step response of the dominant pole approximation. Note the final time, settling time, and overshoot (if underdamped).

\[ G_A(s) = \frac{5000}{(s+2)(s+20)(s^2+20s+500)} \]
Solution 1B

• Construct a first-order or second-order approximation from the dominant pole.
• Do you expect the dominant pole approximation to be accurate?

\[ G_B(s) = \frac{24}{(s+1)(s+2)^2(s+3)} \]
Solution 1-Extra Space
This lecture describes impact of actuator saturation and rate limits. These limits:

- Cause slower speed of response and
- Can lead to overshoot and oscillations if the controller does not properly account for the limits.

Anti-windup is one method to reduce the effect of saturation.
- It will prevent overshoot and oscillations.
- However, it does not change the slower speed of response which is a physical limit of the actuators.
Problem 2

Consider the following plant and PI controller:

\[ \dot{y}(t) + 4y(t) = 2u(t) \quad u(t) = 5e(t) + 20 \int_0^t e(\tau)\,d\tau \]

A) What is the ODE that models the closed-loop from \( r \) to \( y \)?

B) The actuator saturates at \( u \in [-3, 3] \). Do you expect saturation to cause any issues if the reference commands are in the range \( r \in [-1, 1] \)? If yes, then how might you alleviate the issue?

C) Suppose instead that the references are in the range \( r \in [-10, 10] \). Can any controller (not just the PI controller above) achieve good reference tracking with this actuator? If not, then how would you re-select the actuator?
Consider the following plant and PI controller:

$$\dot{y}(t) + 4y(t) = 2u(t) \quad u(t) = 5e(t) + 20 \int_0^t e(\tau) d\tau$$

A) What is the ODE that models the closed-loop from $r$ to $y$?
Consider the following plant and PI controller:

\[ \dot{y}(t) + 4y(t) = 2u(t) \quad u(t) = 5e(t) + 20 \int_0^t e(\tau) \, d\tau \]

B) The actuator saturates at \( u \in [-3, +3] \). Do you expect saturation to cause any issues if the reference commands are in the range \( r \in [-1, +1] \)? If yes, then how might you alleviate the issue?
Consider the following plant and PI controller:

\[
\dot{y}(t) + 4y(t) = 2u(t) \quad u(t) = 5 e(t) + 20 \int_0^t e(\tau) \, d\tau
\]

C) Suppose instead that the references are in the range \( r \in [-10, +10] \). Can any controller (not just the PI controller above) achieve good reference tracking with this actuator? If not, then how would you re-select the actuator?
Solution 2-Extra Space
ECE 486: Control Systems

Lecture 10C: Control Law Implementation
Key Takeaways

It is common to implement controllers on a microprocessor. This lecture discusses some of the details associated with this implementation:

• Sample a measurement at specific (discrete) time intervals
• Update the control input \( u \) at each sample time.
• Hold the control input \( u \) constant until the next update.

The update equation is chosen to approximate the properties of the designed (ODE) controller. The update equation can be implemented on a microprocessor with a few lines of code.
Consider the following plant and PI controller:

\[ 2 \dot{y}(t) + 6y(t) = 8u(t) \]
\[ u(t) = 2.5 e(t) + 9 \int_0^t e(\tau) d\tau \]

A) What sampling time \( \Delta t \) would you recommend for a discrete-time implementation?

B) The value of \( u(t) \) at \( t = \Delta t \) is:

\[ u(\Delta t) = 2.5 e(\Delta t) + 9 \int_0^{\Delta t} e(\tau) d\tau \]

Approximate \( u_1 := u(\Delta t) \) in terms \( e_0 := e(0) \) and \( e_1 := e(\Delta t) \).

C) The computations in the discrete-time update are not instantaneous and require some time. How can this be modeled?
Solution 3A

\[ 2y(t) + 6y(t) = 8u(t) \quad u(t) = 2.5e(t) + 9 \int_0^t e(\tau) \, d\tau \]

A) What sampling time \( \Delta t \) would you recommend for a discrete-time implementation?
B) The value of $u(t)$ at $t=\Delta t$ is:

$$u(\Delta t) = 2.5 e(\Delta t) + 9 \int_0^{\Delta t} e(\tau) \, d\tau$$

Approximate $u_1 := u(\Delta t)$ in terms $e_0 := e(0)$ and $e_1 := e(\Delta t)$. 

$$2y(t) + 6y(t) = 8u(t) \quad \quad u(t) = 2.5 e(t) + 9 \int_0^t e(\tau) \, d\tau$$
Solution 3C

\[ 2\dot{y}(t) + 6y(t) = 8u(t) \quad u(t) = 2.5e(t) + 9 \int_0^t e(\tau) \, d\tau \]

C) The computations in the discrete-time update are not instantaneous and require some time. How can this be modeled?
Solution 3-Extra Space