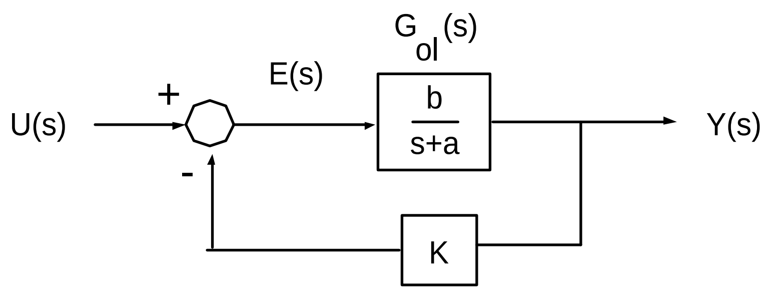
PID Control

Proportional Even Better Than Unity Feedback

Recall: Unity Feedback

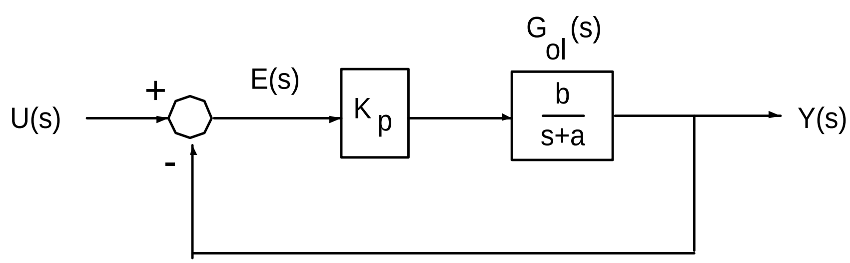


Given step input $u(t) = 1 \leftrightarrow U(s) = \frac{1}{s}$

Inverse Laplace: $y(t) = \frac{\quad}{+}(1 - e \qquad)$

Steady-state: $y_{ss} = \frac{b}{a + Kb}$

As feedback gain increases, output attenuates

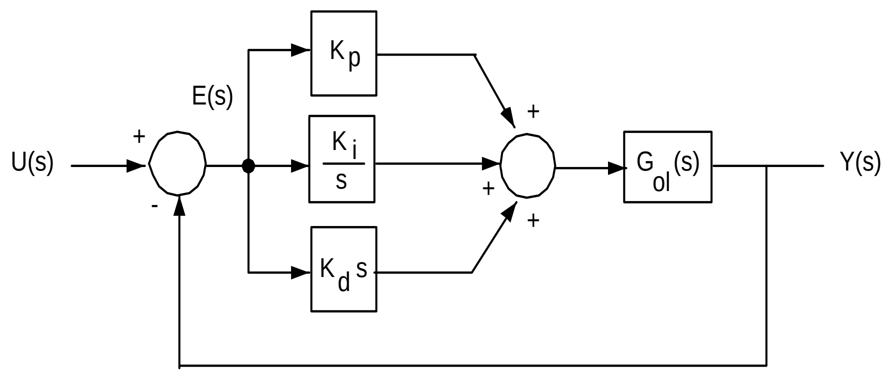


Inverse Laplace $y(t) = \frac{\quad}{+}(1 - e \qquad)$

Steady-state: $y_{ss} = \frac{K_p b}{a + K_p b}$

As gain increases, proportional output reaches 1 (as desired)

Review: Proportional-Integral-Derivative Control



$$Y(s) = G_{ol}(K_p + \frac{K_i}{s} + K_d s)(U - Y)$$

- I improves ____________________
  at the expense of _________
- D improves ______ at the expense of steady-state accuracy
- PD improves stability without degrading _________much
- PI improves steady-state accuracy without degrading ______ much

$$G_{cl}(s) = \frac{Y(s)}{U(s)} = \frac{G_{ol}(K_p s + K_i + K_d s^2)}{s + G_{ol}(K_p s + K_i + K_d s^2)}$$

General PID CLTF

Case Study: Proportional only control

$$Y(s) = \frac{G_{ol}(K_p s)U(s)}{s + G_{ol}(K_p s)}$$

Step input $U(s) = \frac{1}{s}$

$$Y(s) = \frac{G_{ol} K_p}{s(1 + G_{ol} K_p)}$$

Final Value Theorem

$$\lim_{t \to \infty} \quad = \lim_{s \to 0}$$

$$y_{ss} = \lim_{s \to 0} sY(s) = \text{\_\_\_\_\_\_\_\_}$$

Response depends on OLTF poles and zeros

System Type: Keys to PID Design

$$y_{ss} = \lim_{s \to 0} sY(s) = \frac{G_{ol} K_p}{1 + G_{ol} K_p}$$

Proportional only control

$$G_{ol}(s) = \frac{(\tau_a s + 1) \cdots (\tau_m s + 1)}{s^i (\tau_1 s + 1) \cdots (\tau_n s + 1)} \qquad (1)$$

General form any TF

$i = 0$	Type 0 System
$i = 1$	Type 1 System
$i = 2$	Type 2 System

Case 1: Type 0 System (step response)

$$y_{ss} = \lim_{s \to 0} sY(s) = \lim_{s \to 0} \frac{\dfrac{(\tau_a s + 1) \cdots (\tau_m s + 1) K_p}{(\tau_1 s + 1) \cdots (\tau_n s + 1)}}{\dfrac{(\tau_1 s + 1) \cdots (\tau_n s + 1) + (\tau_a s + 1) \cdots (\tau_m s + 1) K_p}{(\tau_1 s + 1) \cdots (\tau_n s + 1)}}$$

$y_{ss} =$ ______

NB: There is a discrete error, but if $K_p >> 1$ then $y_{ss} \approx 1$

Case 2: Type 1 System (step response)

$$y_{ss} = \lim_{s \to 0} sY(s) = \lim_{s \to 0} \frac{\dfrac{(\tau_a s + 1) \cdots (\tau_m s + 1) K_p}{s(\tau_1 s + 1) \cdots (\tau_n s + 1)}}{\dfrac{s(\tau_1 s + 1) \cdots (\tau_n s + 1) + (\tau_a s + 1) \cdots (\tau_m s + 1) K_p}{s(\tau_1 s + 1) \cdots (\tau_n s + 1)}}$$

$y_{ss} =$ ______

No error! Type 1 and 2 systems have **no error** with step response

Goal: Increase System Type with Integrator

$$G_{cl}(s) = \frac{Y(s)}{U(s)} = \frac{G_{ol}(K_p s + K_i + K_d s^2)}{s + G_{ol}(K_p s + K_i + K_d s^2)}$$

General PID CLTF

Case 1: Pure Integrator $K_i = 1$ $\quad Y(s) = \frac{G_{ol}U(s)}{s + G_{ol}}$ $\quad$ Step input $U(s) = \frac{1}{s}$

Substituting OLTF general form from (1)

$$Y(s) = \frac{\frac{(\tau_a s+1)\cdots(\tau_m s+1)}{s^i(\tau_1 s+1)\cdots(\tau_n s+1)} \cdot \frac{1}{s}}{s + \frac{(\tau_a s+1)\cdots(\tau_m s+1)}{s^i(\tau_1 s+1)\cdots(\tau_n s+1)}}$$

$$Y(s) = \frac{(\tau_a s+1)\cdots(\tau_m s+1) \cdot \frac{1}{s}}{[s \cdot s^i(\tau_1 s+1)\cdots(\tau_n s+1)] + (\tau_a s+1)\cdots(\tau_m s+1)}$$

$$y_{ss} = \lim_{s \to 0} sY(s) = \frac{(\tau_a s+1)\cdots(\tau_m s+1)}{[s \cdot s^i(\tau_1 s+1)\cdots(\tau_n s+1)] + (\tau_a s+1)\cdots(\tau_m s+1)}$$

Thus, adding an integrator yields $y_{ss} =$ __ **irregardless** of system type

Goal: Decrease System Type with Derivative

$$G_{cl}(s) = \frac{Y(s)}{U(s)} = \frac{G_{ol}(K_p s + K_i + K_d s^2)}{s + G_{ol}(K_p s + K_i + K_d s^2)}$$

General PID CLTF

Case 2: Pure Derivative $K_d = 1$

$$Y(s) = \frac{G_{ol} s^2 U(s)}{s + G_{ol} s^2}$$ Step input $U(s) = \frac{1}{s}$ then $$Y(s) = \frac{G_{ol}}{1 + G_{ol} s}$$

Substituting OLTF general form from (1)

$$Y(s) = \frac{\dfrac{(\tau_a s+1)\cdots(\tau_m s+1)}{s^i(\tau_1 s+1)\cdots(\tau_n s+1)}}{\dfrac{s^i(\tau_1 s+1)\cdots(\tau_n s+1) + (\tau_a s+1)\cdots(\tau_m s+1)s}{s^i(\tau_1 s+1)\cdots(\tau_n s+1)}}$$

$$y_{ss} = \lim_{s\to 0} sY(s) = \frac{s(\tau_a s+1)\cdots(\tau_m s+1)}{s^i(\tau_1 s+1)\cdots(\tau_n s+1) + (\tau_a s+1)\cdots(\tau_m s+1)s}$$

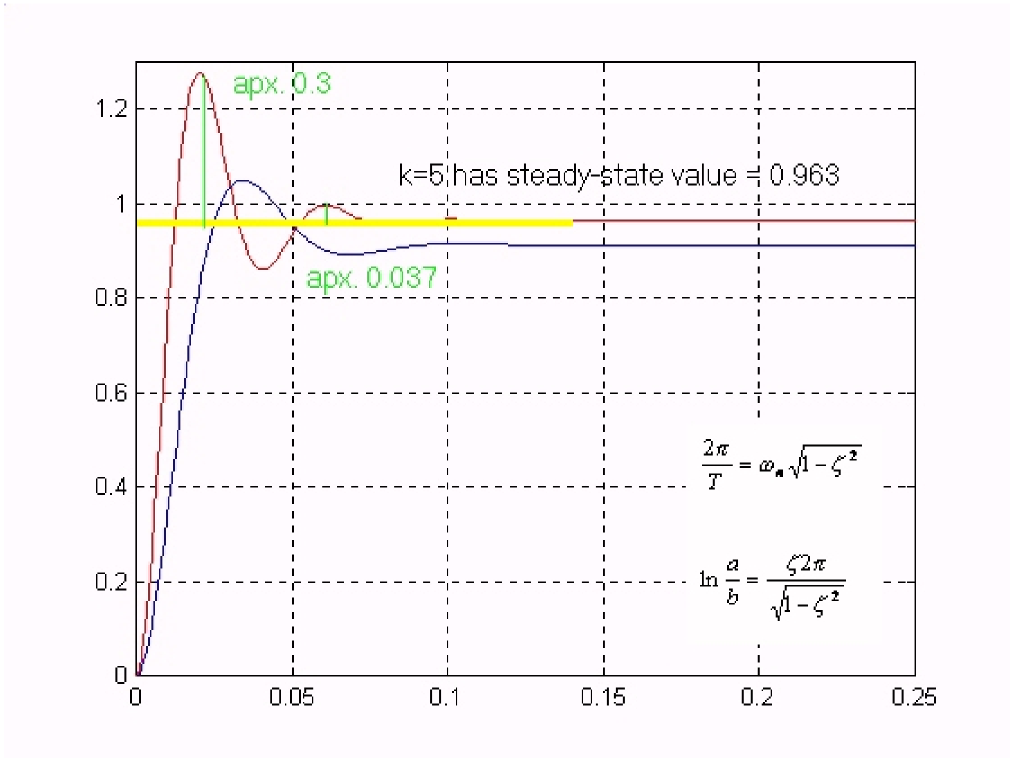
$$y_{ss} = \lim_{s\to 0} sY(s) = \frac{(\tau_a s+1)\cdots(\tau_m s+1)}{s^{i-1}(\tau_1 s+1)\cdots(\tau_n s+1) + (\tau_a s+1)\cdots(\tau_m s+1)}$$

Derivative will yield $y_{ss} =$ ____ if $i > 1$

Case Studies: Matlab Simulations

Recall motor given by: $G_{ol} = \dfrac{5200}{(s+10)(s+100)}$ Type 0 system

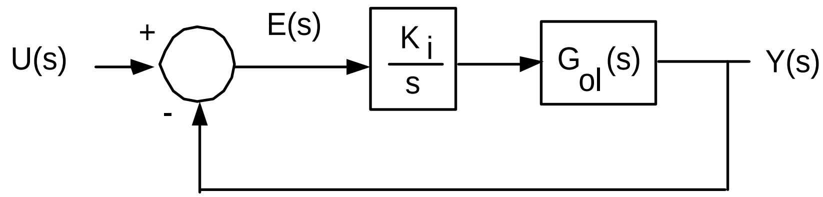
For $K_p = 2$ $G_{cl}(s) = \dfrac{10400}{s^2 + 110s + 11400}$ For $K_p = 5$ $G_{cl}(s) = \dfrac{26000}{s^2 + 110s + 27000}$



Matlab confirmed that NON-ZERO steady-state error (see mtr_p.m)

Case Studies: Matlab Simulations – Pure Integrator

Problem: Given



$$G_{ol} = \frac{5200}{(s+10)(s+100)}$$

$$Y(s) = G_{ol}(s)\frac{K_i}{s}(U - Y)$$

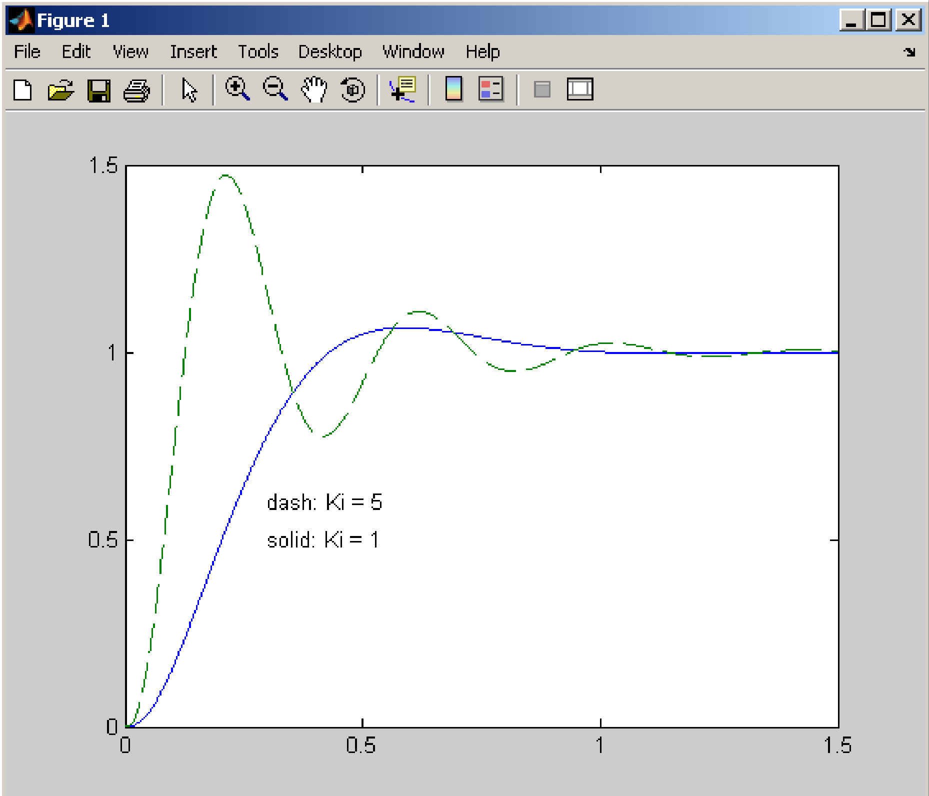
1-1: Show that the CLTF is

$$G_{cl}(s) = \frac{Y(s)}{U(s)} = \frac{5200K_i}{s(s+10)(s+100)+5200K_i}$$

1-2: Show that for a unit step response that $y_{ss} = 1$

1-3: In Matlab, simulate and display the unit step response with $K_i = 1$ and $K_i = 5$

Pure Integrator Control

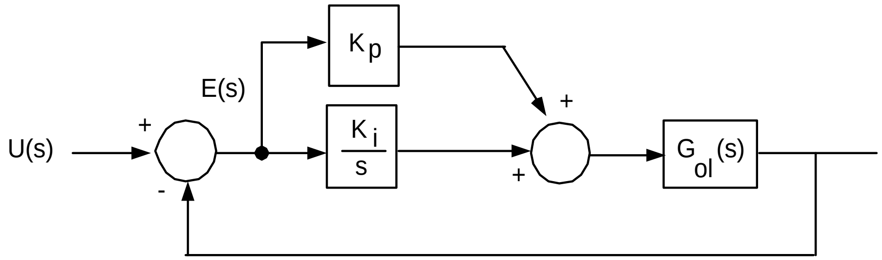


Code snippet

```
numki1 = [0   0   0 5200];     % numerator for Ki = 1
denki1 = [1 110 1000 5200]; % denominator

numki5 = [0   0   0 26000];     % numerator for Ki = 1
denki5 = [1 110 1000 26000]; % denominator

tFinal = input('Time range e.g. enter 0.5: ');
freq = input('Sampling frequency in Hz e.g. enter 1000: ');
tStep = 1/freq;
```

Case Studies: Matlab Simulations – PI control

Problem: Given



$$G_{ol} = \frac{5200}{(s+10)(s+100)}$$

2-1: Show that the CLTF is

$$G_{cl}(s) = \frac{Y(s)}{U(s)} = \frac{5200K_p s + 5200K_i}{s^3 + 110s^2 + (1000 + 5200K_p)s + 5200K_i}$$

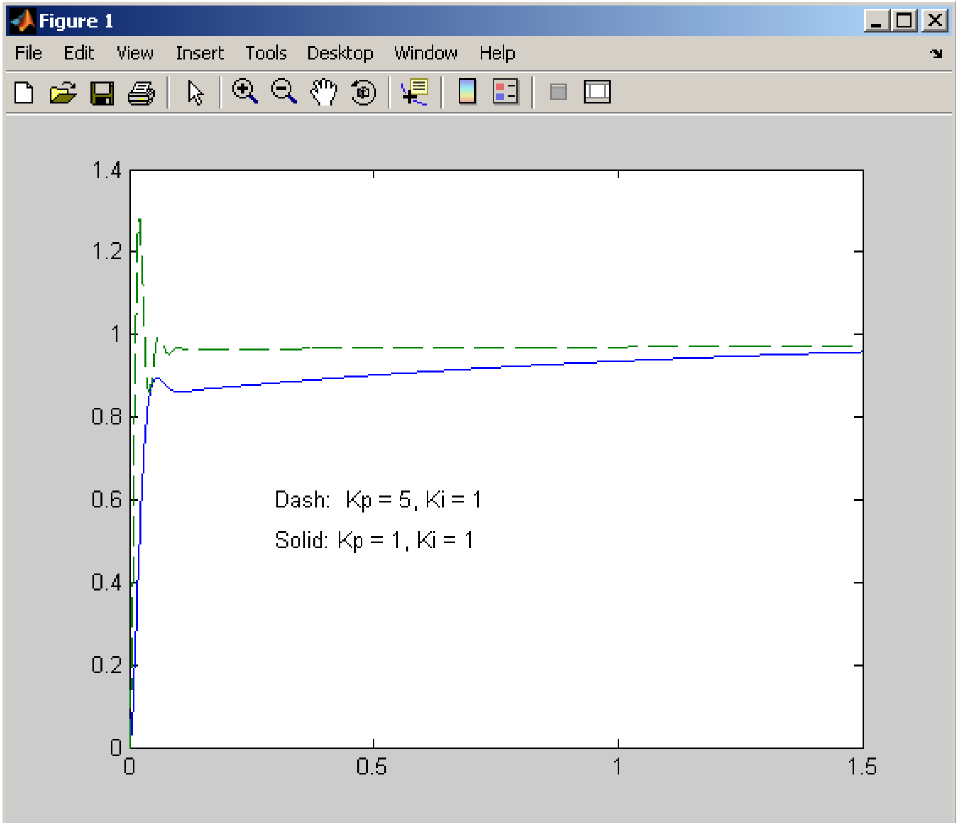
2-2: In Matlab, simulate and display the unit step response with

A. $K_i = 1 \quad K_p = 1$

B. $K_i = 1 \quad K_p = 5$

C. $K_i = 5 \quad K_p = 1$

D. $K_i = 5 \quad K_p = 5$

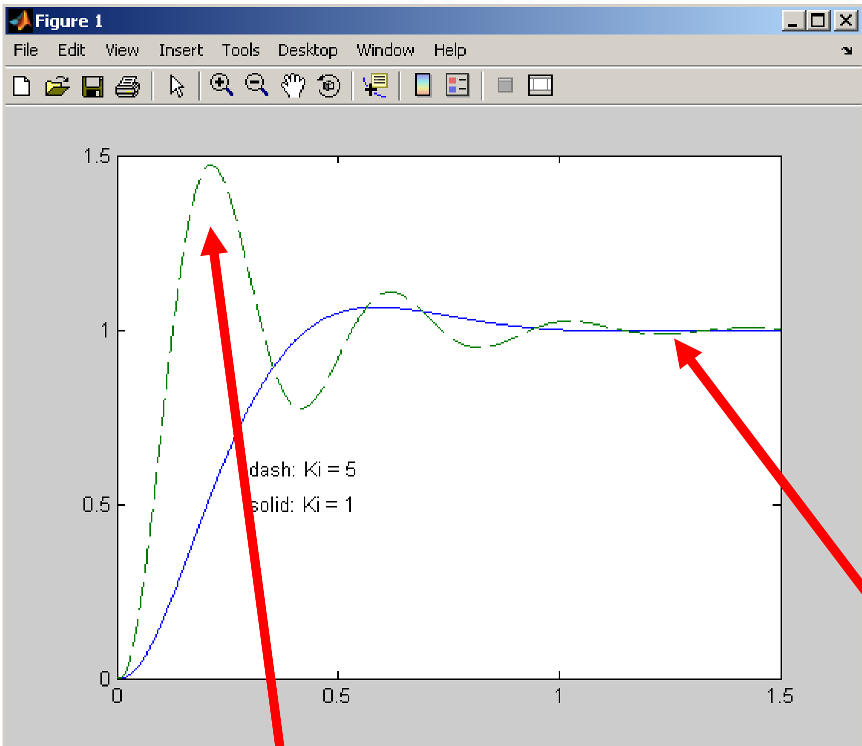


PI Control

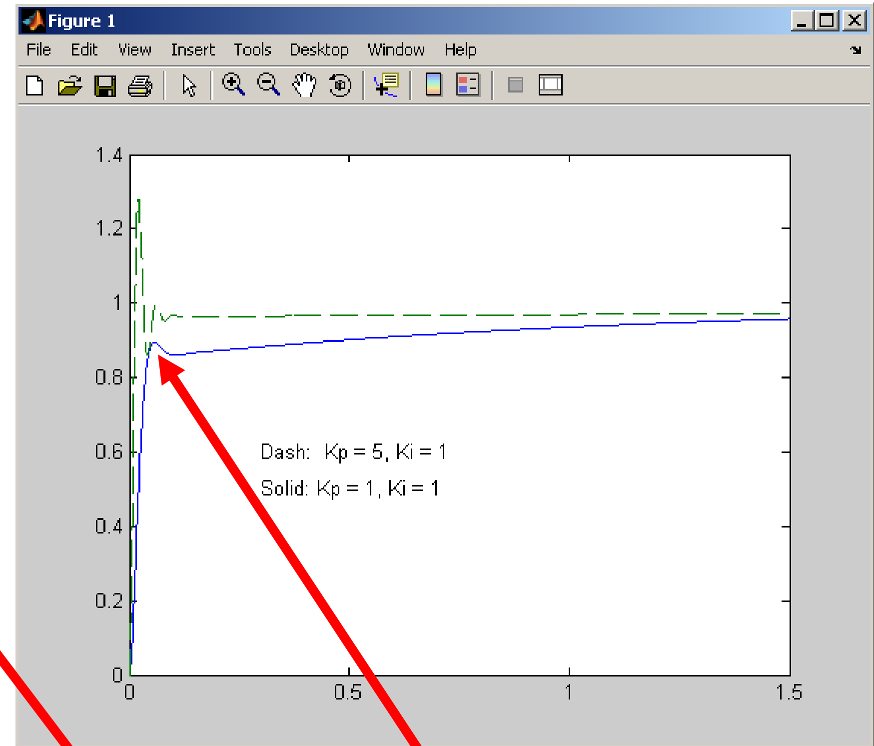
```
numkp5ki1 = [0 0   26000 5200]; % numerator for Kp = 5, Ki = 1
denkp5ki1 = [1 110 27000 5200]; % denominator

numkp1ki1 = [0 0   5200 5200]; % numerator for Kp = 1 , Ki = 1
denkp1ki1 = [1 110 6200 5200]; % denominator

tFinal = input('Time range e.g. enter 0.5: ');
freq = input('Sampling frequency in Hz e.g. enter 1000: ');
tStep = 1/freq;
```

Code snippet

Conclusions



Pure Integrator Control



PI Control

- I improves steady-state accuracy at the expense of stability
- PI improves steady-state accuracy without degrading stability much

Just like was stated in the beginning (see Slide3)