Single Input Single Output system control

Elena VANNEAUX

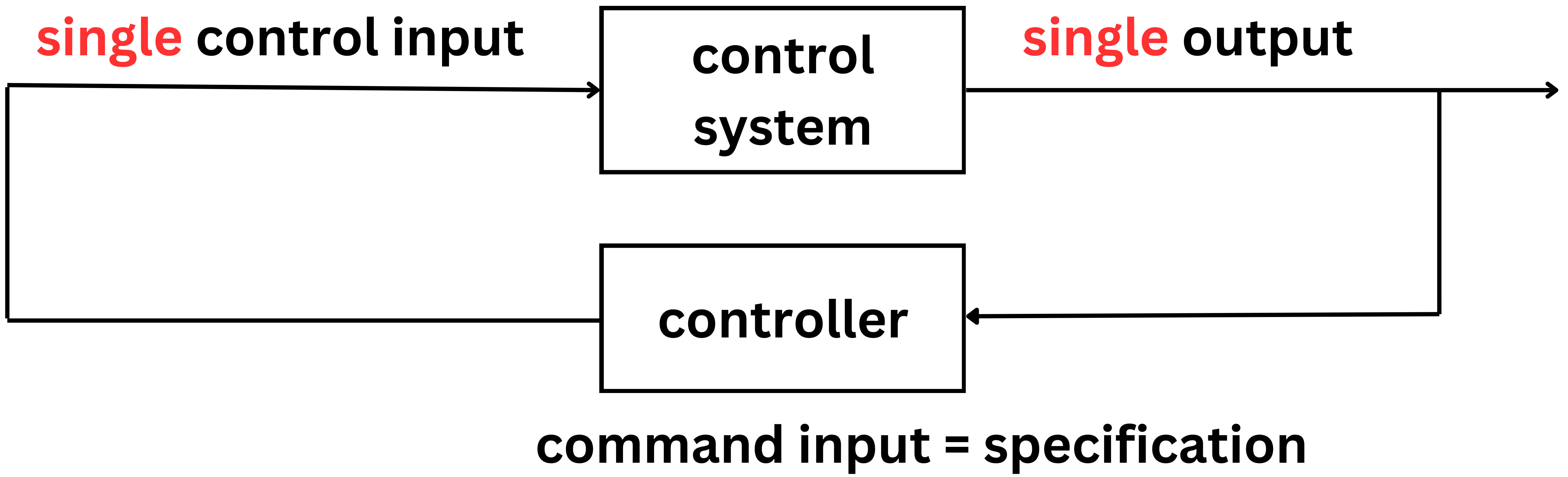
elena.vanneaux@ensta-paris.fr

Course grade breakdowns

Labs - 50%

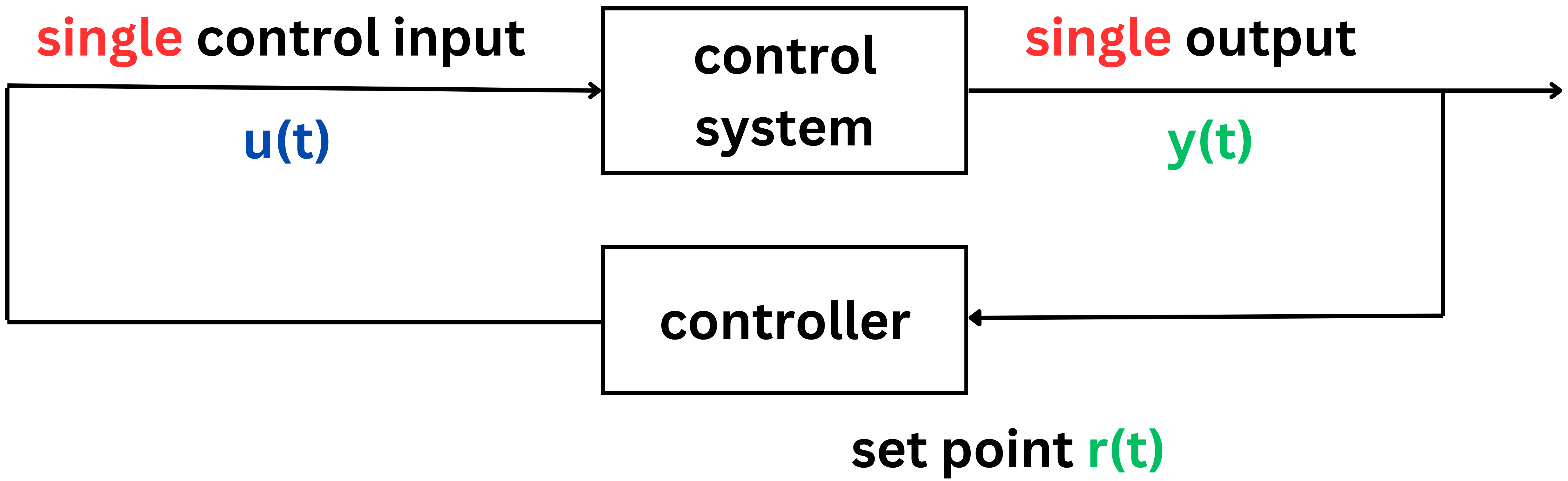
Final project - 50 %

SISO system feedback control



Feedback controller design = how to use the sensor data (output) to generate the correct actuator commands (control input) to ensure that the output of the system satisfies the specification

SISO system feedback control



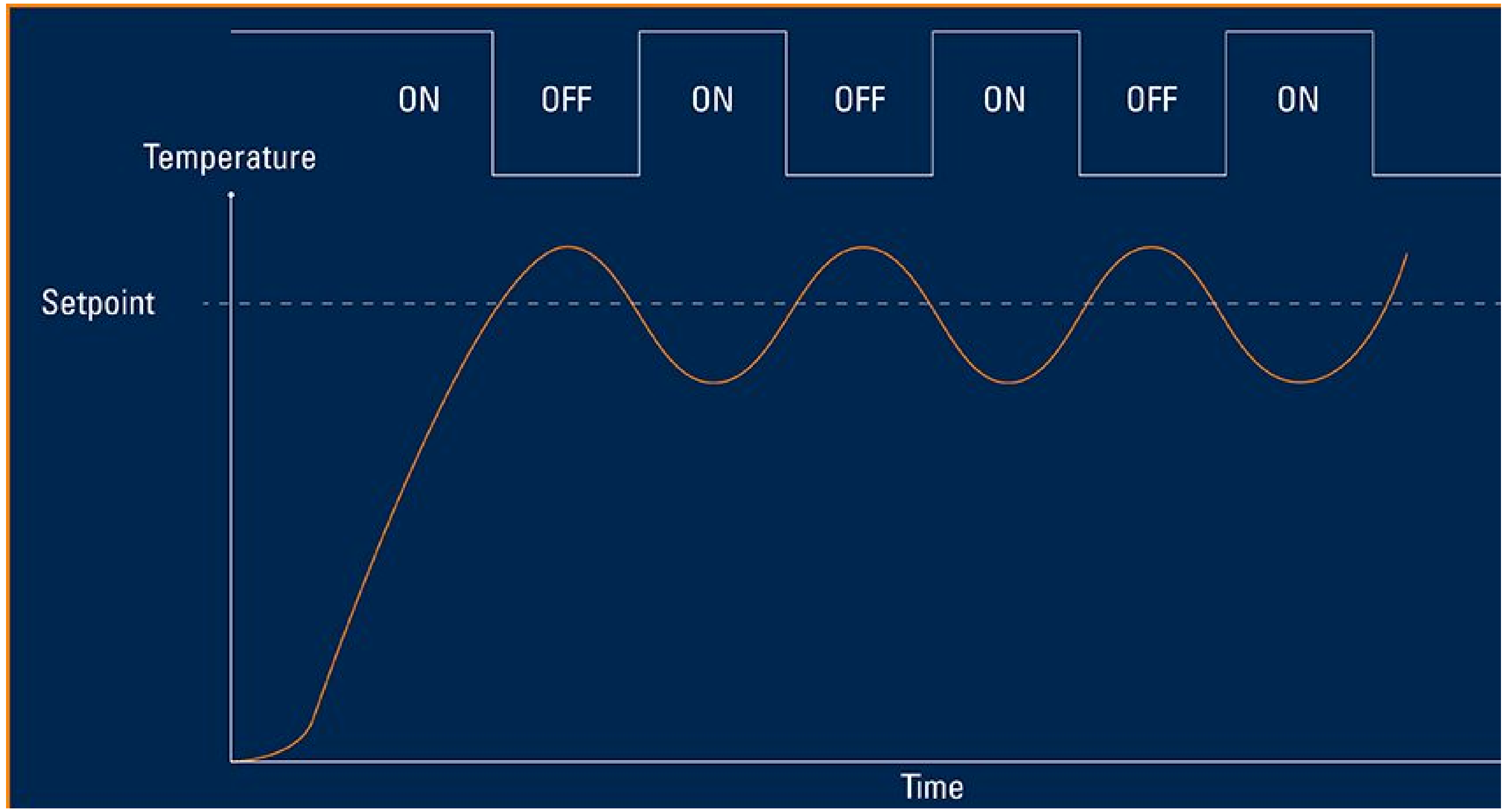
Specification: reference input (set point) r(t)

Feedback controller design: generate control input u(t) such that the output of the system y(t) coincides with r(t)

On-Off Controller

Simple thermostat

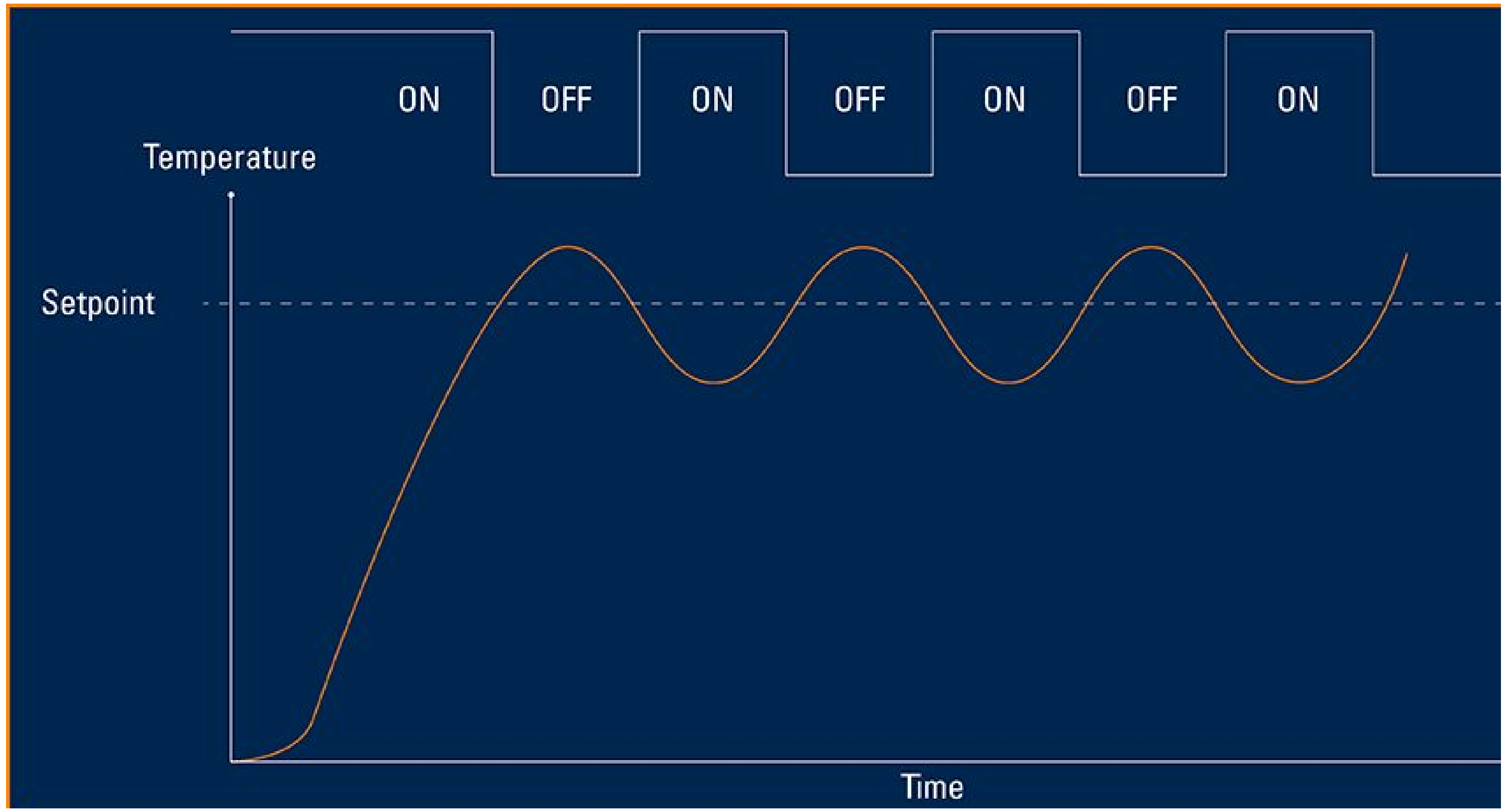
turning on heaters when it is too cold and turning them off when it is too hot.



On-Off Controller

Simple thermostat

turning on heaters when it is too cold and turning them off when it is too hot.



Push back, against the direction of the error

e(t) = y(t) - r(t)

with constant action u

$$e < 0 \Rightarrow u := on \Rightarrow \dot{x} = f(x, on) > 0$$

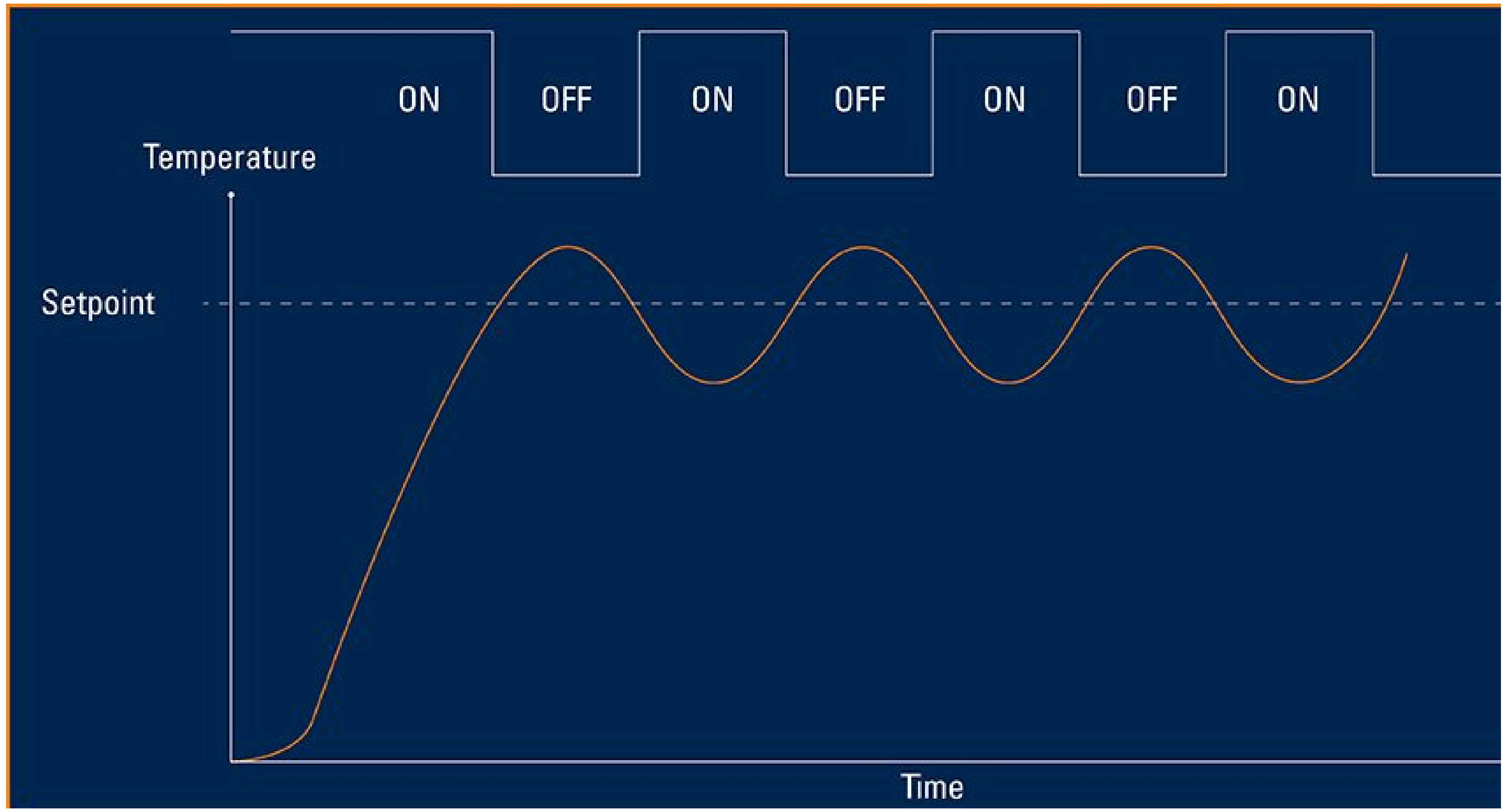
$$e > 0 \Rightarrow u := off \Rightarrow \dot{x} = f(x, off) < 0$$

Don’t switch more often than T minutes to avoid flattering

On-Off Controller

Simple thermostat

turning on heaters when it is too cold and turning them off when it is too hot.



Push back, against the direction of the error

e(t) = r(t) - y(t)

with constant action u

$$e < 0 \Rightarrow u := on \Rightarrow \dot{x} = f(x, on) > 0$$

$$e > 0 \Rightarrow u := off \Rightarrow \dot{x} = f(x, off) < 0$$

Don't switch more often than T minutes to avoid flattering

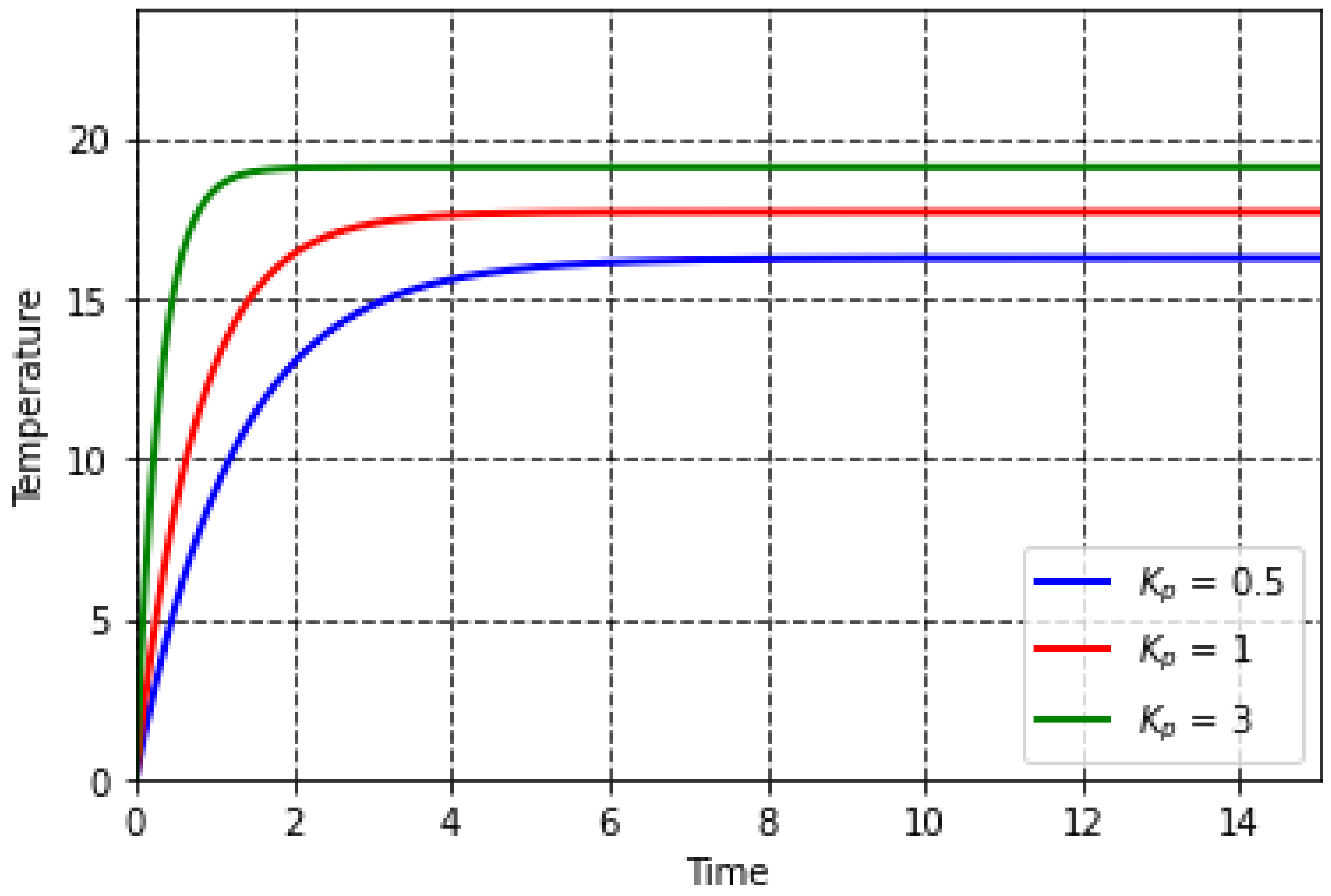
Good for reaching the setpoint

Not very good for staying near it

P - Proportional Controller

Simple thermostat

$$\dot{T} = a(T_{ambient} - T) + u(t)$$



Proportional control

$$u(t) = K_p(r(t) - y(t))$$

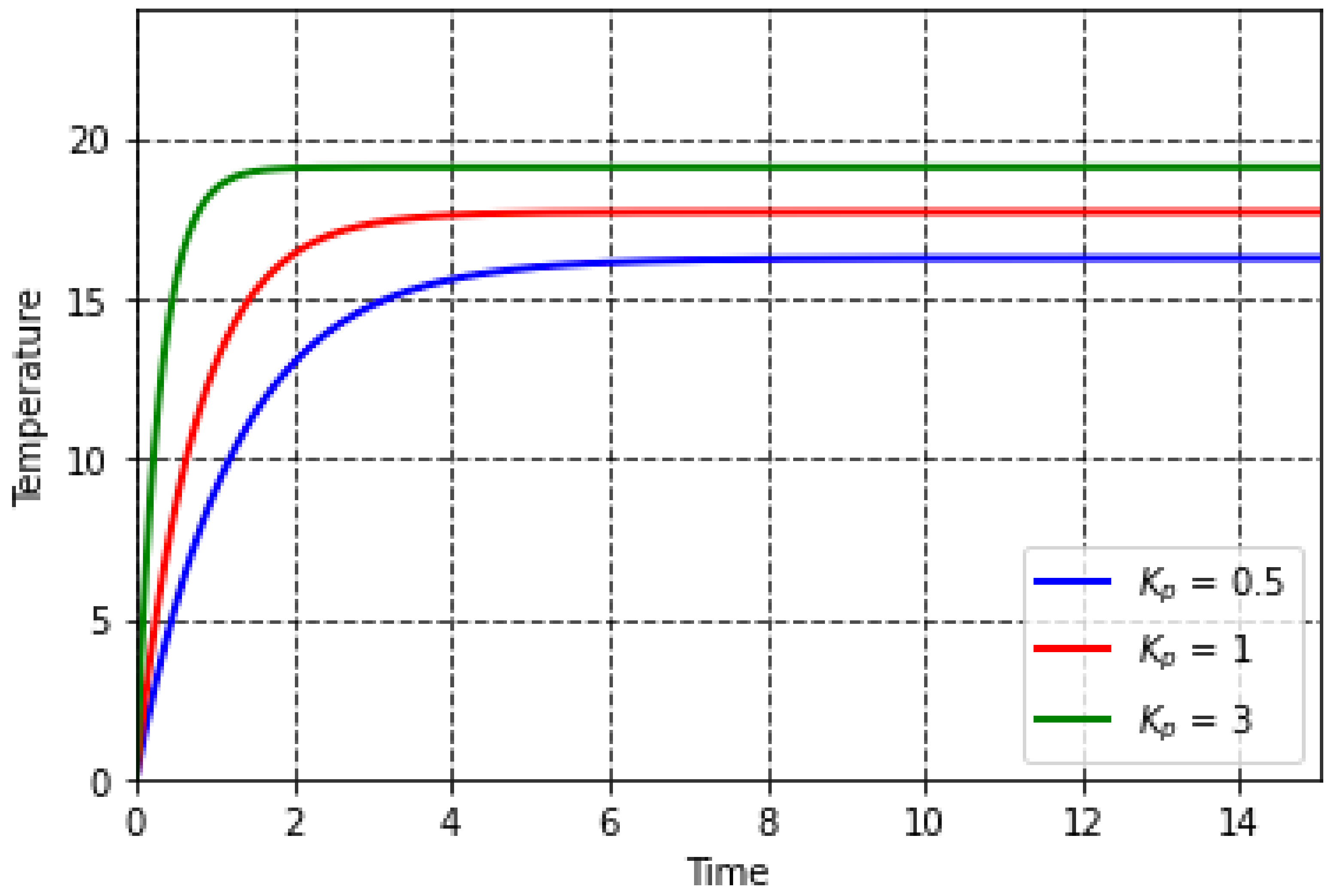
The control input tries to move the system in a direction that is opposite to the error, and is proportional to the error in magnitude.

P - Proportional Controller

Simple thermostat

$$\dot{T} = a(T_{ambient} - T) + u(t) \Leftrightarrow T = -aT + u(t) + aT_{ambient}$$

disturbance



Proportional control

$$u(t) = K_p(r(t) - y(t))$$

The control input tries to move the system in a direction that is opposite to the error, and is proportional to the error in magnitude.

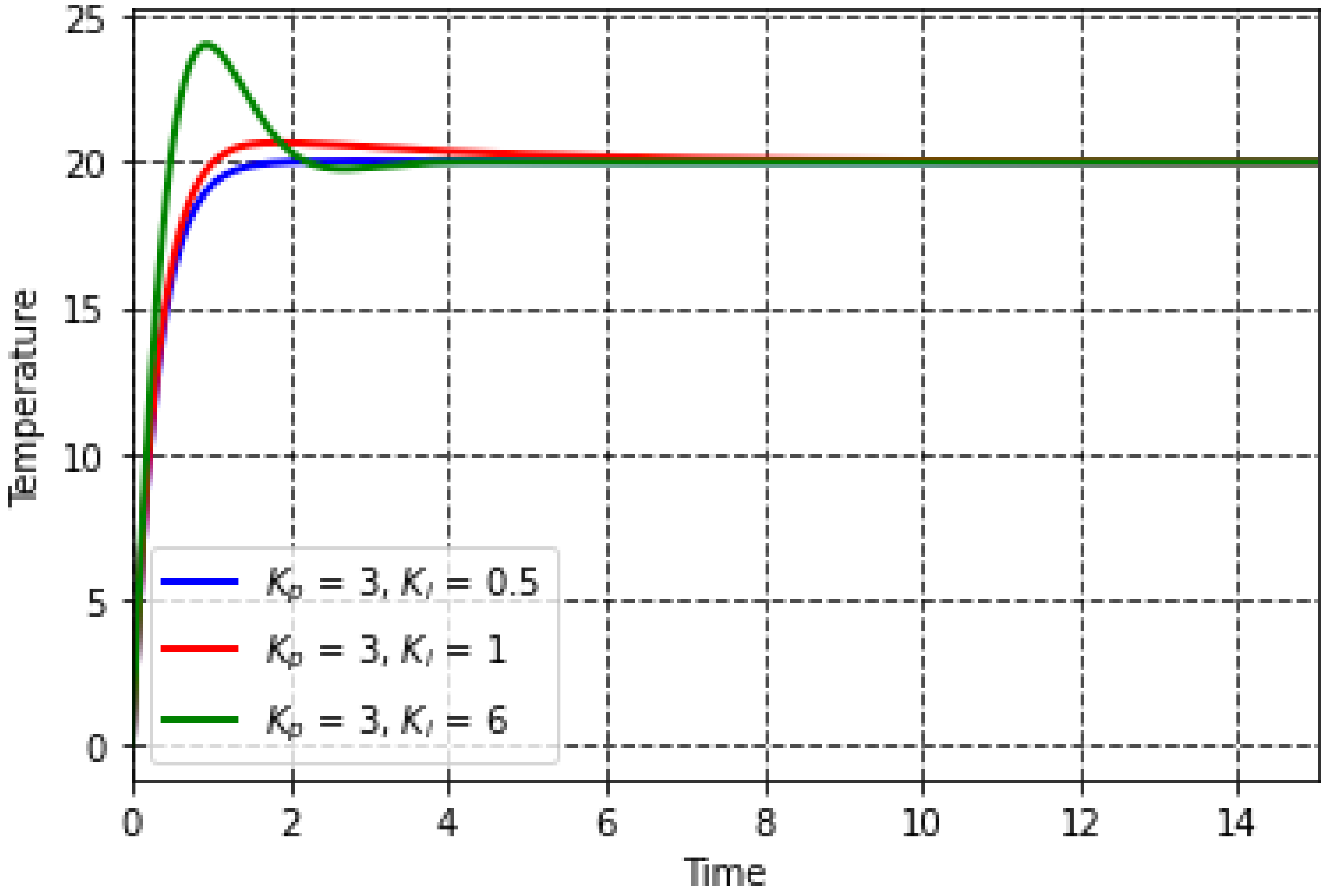
Non zero steady state error

I - Integral Controller

Simple thermostat

$$\dot{T} = a(T_{ambient} - T) + u(t) \Leftrightarrow T = -aT + u(t) + aT_{ambient}$$

disturbance



Integral Term

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau$$

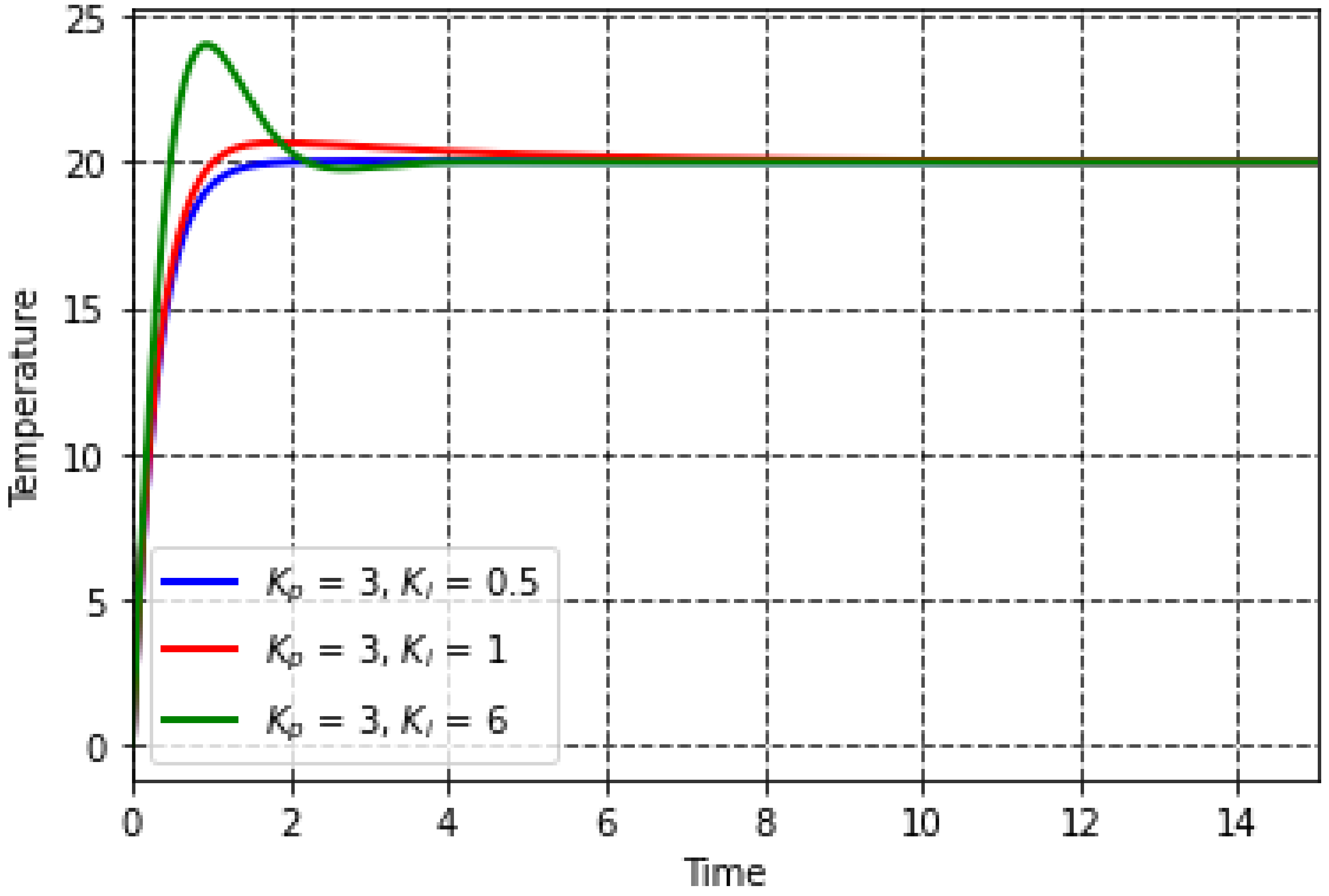
Term I accounts for past values of the error and integrates them over time to produce the I term.

I - Integral Controller

Simple thermostat

$$\dot{T} = a(T_{ambient} - T) + u(t) \Leftrightarrow T = -aT + u(t) + aT_{ambient}$$

disturbance



Integral Term

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau$$

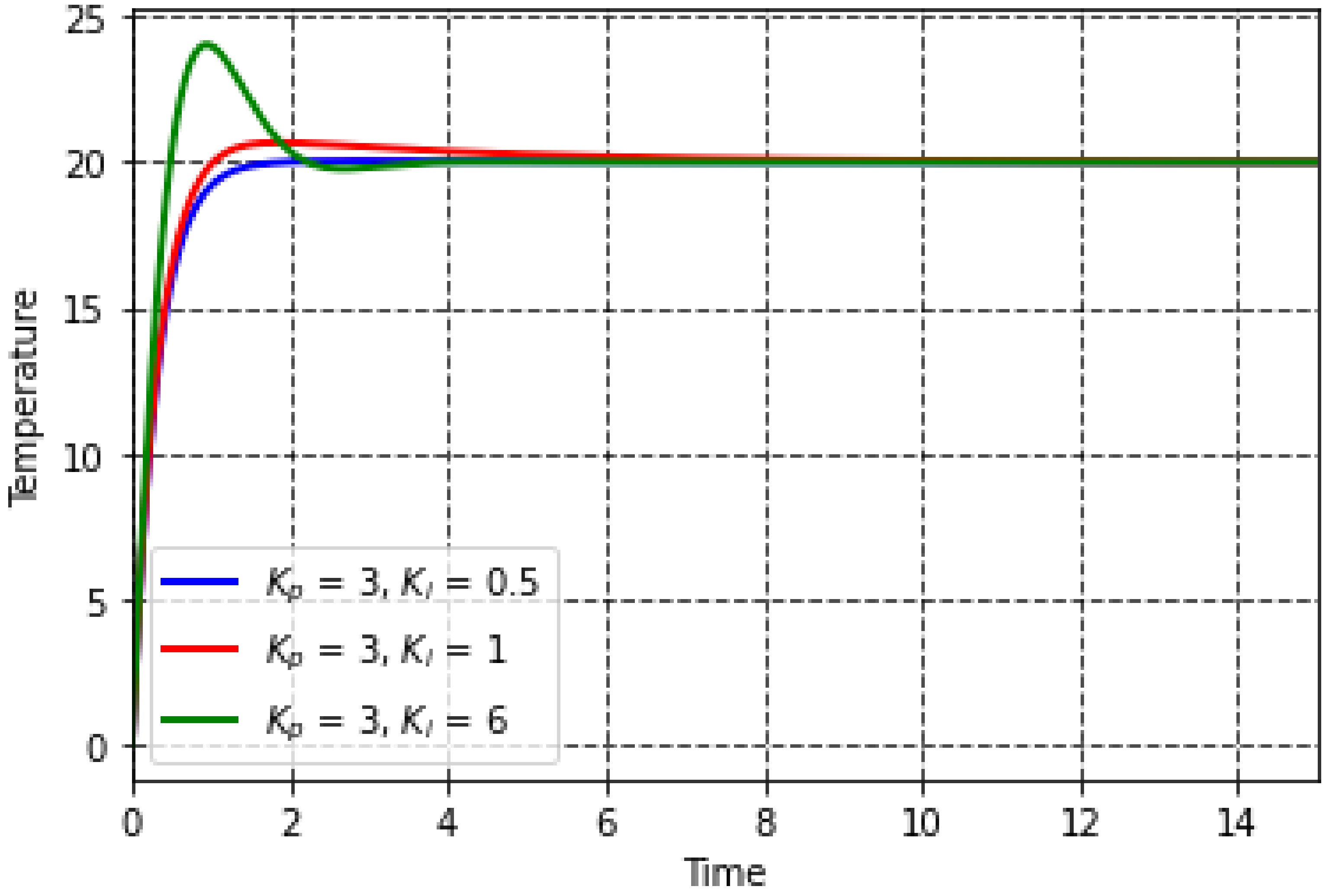
if there is a residual error after the application of P control, the integral term seeks to eliminate the residual error by adding a control effect due to the historic cumulative value of the error.

I - Integral Controller

Simple thermostat

$$\dot{T} = a(T_{ambient} - T) + u(t) \Leftrightarrow T = -aT + u(t) + aT_{ambient}$$

disturbance

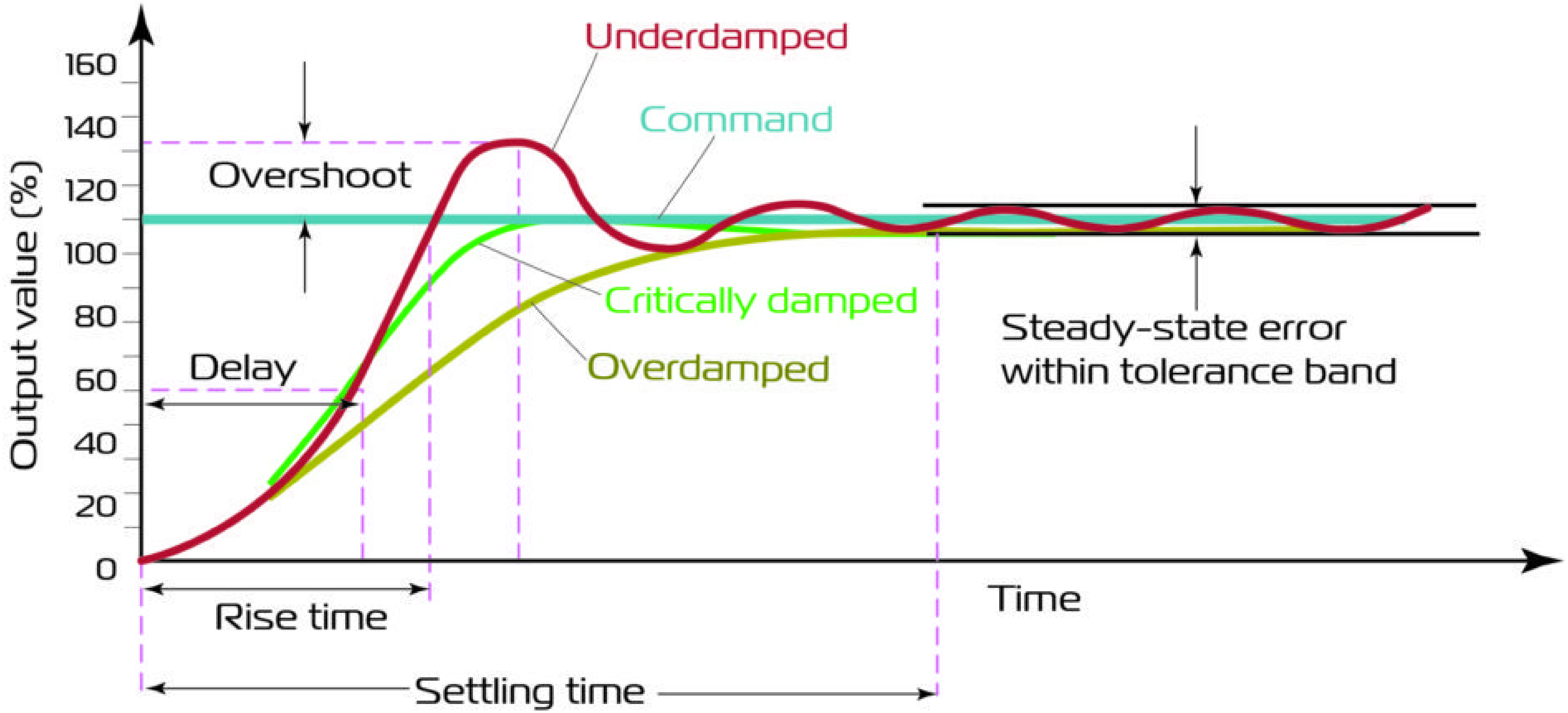


Integral Term

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau$$

When the error is eliminated, the integral term will cease to grow. This will result in the proportional effect diminishing as the error decreases, but this is compensated for by the growing integral effect.

Glossary


Underdamped
Command
Overshoot
Critically damped
Overdamped
Delay
Steady-state error
within tolerance band
Output value (%)
160
140
120
100
80
60
40
20
0
Time
Rise time
Settling time

Glossary

Steady state error is the deviation of the output of control system from desired response during steady state

Settling time is the time required for the response curve to reach and stay within a range of certain percentage (usually 5% or 2%) of the final value.

Overshoot is the occurrence of a signal or function exceeding its target. **Undershoot** is the same phenomenon in the opposite direction.

Delay time is the time required for the response to reach half of its final value from the zero instant.

Rise time is the time required for the response to rise from 0% to 100% of its final value. This is applicable for the under-damped systems.
For the over-damped systems, consider the duration from 10% to 90% of the final value.

Critical damping: the condition in which the damping of an oscillator causes it to return as quickly as possible to its equilibrium position without oscillating back and forth about this position

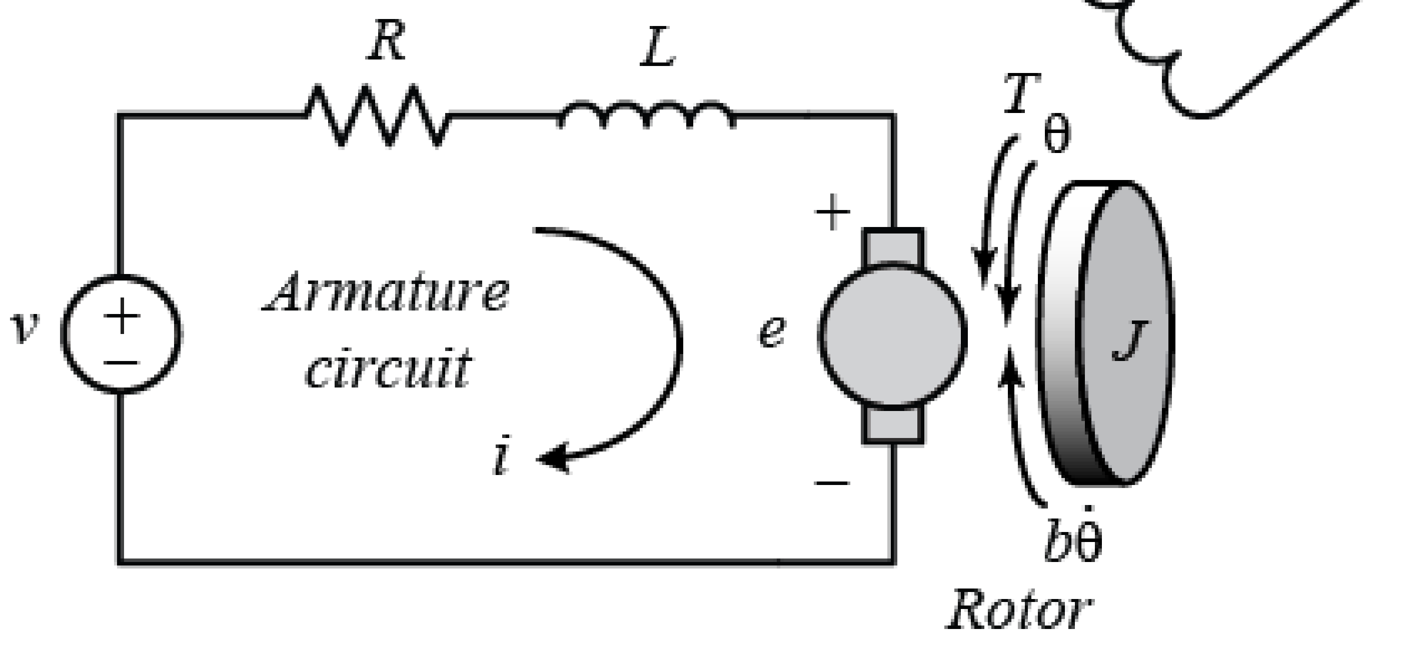
Over damping: the condition in which damping of an oscillator causes it to return to equilibrium without oscillating; oscillator moves more slowly toward equilibrium than in the critically damped system

Under damping: the condition in which damping of an oscillator causes it to return to equilibrium with the amplitude gradually decreasing to zero; system returns to equilibrium faster but overshoots and crosses the equilibrium position one or more times

DC motor control design

$$\dot{x} = Ax + Bu$$

$$y = Cx$$



A common actuator in control systems is the DC motor. It directly provides rotary motion and, coupled with wheels or drums and cables, can provide translational motion.

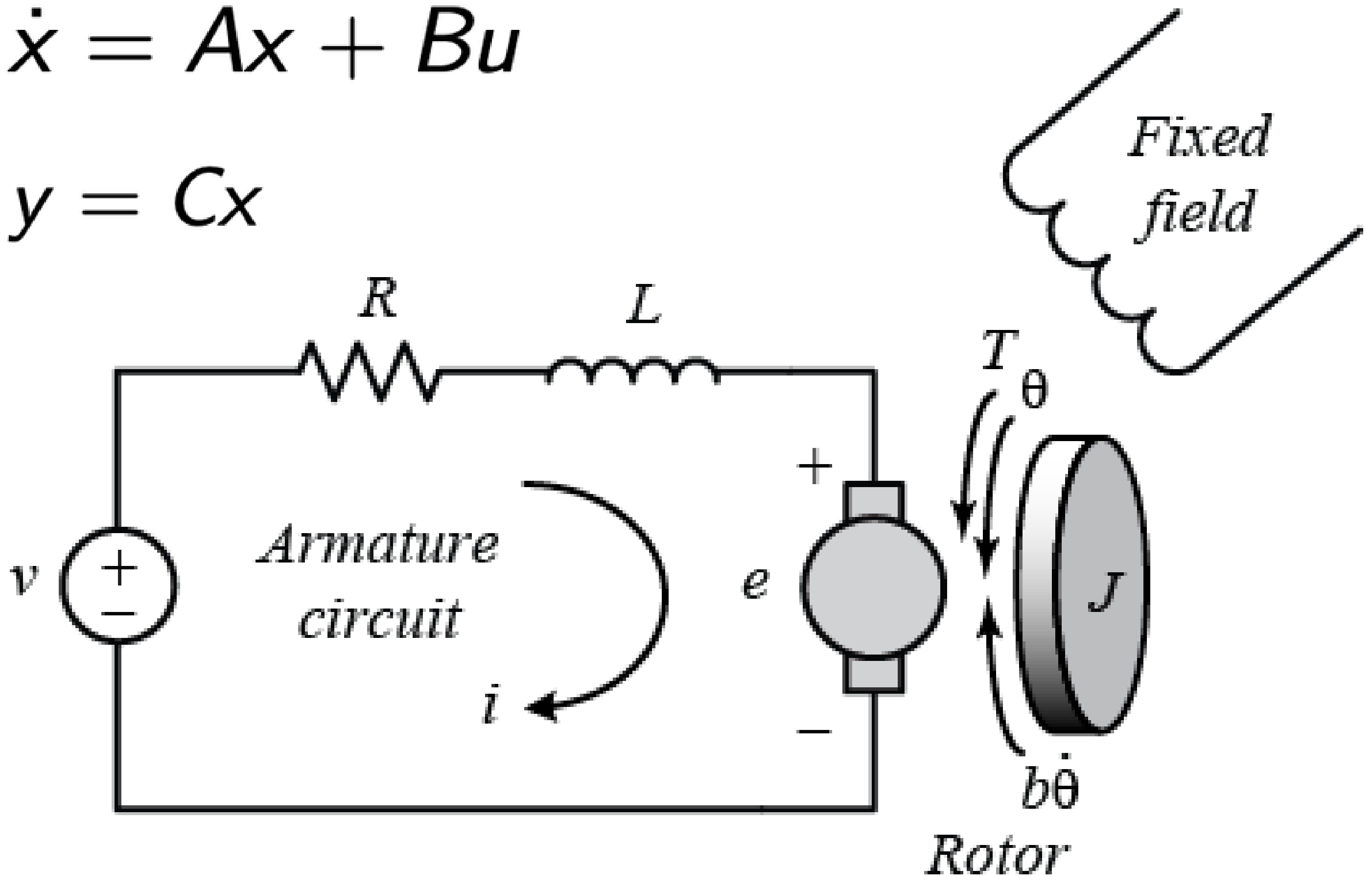
$$A = \begin{pmatrix} -\frac{b}{J} & \frac{K}{J} \\ -\frac{K}{L} & -\frac{R}{L} \end{pmatrix}, \ B = \begin{pmatrix} 0 \\ \frac{1}{L} \end{pmatrix}$$

$$C = (1, 0), x = \begin{pmatrix} \dot{\theta} \\ i \end{pmatrix}, r(t) = 1 \ rad/sec$$

DC motor control design

$$\dot{x} = Ax + Bu$$

$$y = Cx$$



A common actuator in control systems is the DC motor. It directly provides rotary motion and, coupled with wheels or drums and cables, can provide translational motion.

$$A = \begin{pmatrix} -\frac{b}{J} & \frac{K}{J} \\ -\frac{K}{L} & -\frac{R}{L} \end{pmatrix}, \quad B = \begin{pmatrix} 0 \\ \frac{1}{L} \end{pmatrix}$$

$$C = (1, 0), \; x = \begin{pmatrix} \dot{\theta} \\ i \end{pmatrix}, \; r(t) = 1 \; rad/sec$$

The design criteria are the following:

- Settling time less than 2 seconds
- Overshoot less than 5%
- Steady-state error less than 2%

The design criteria are the following:

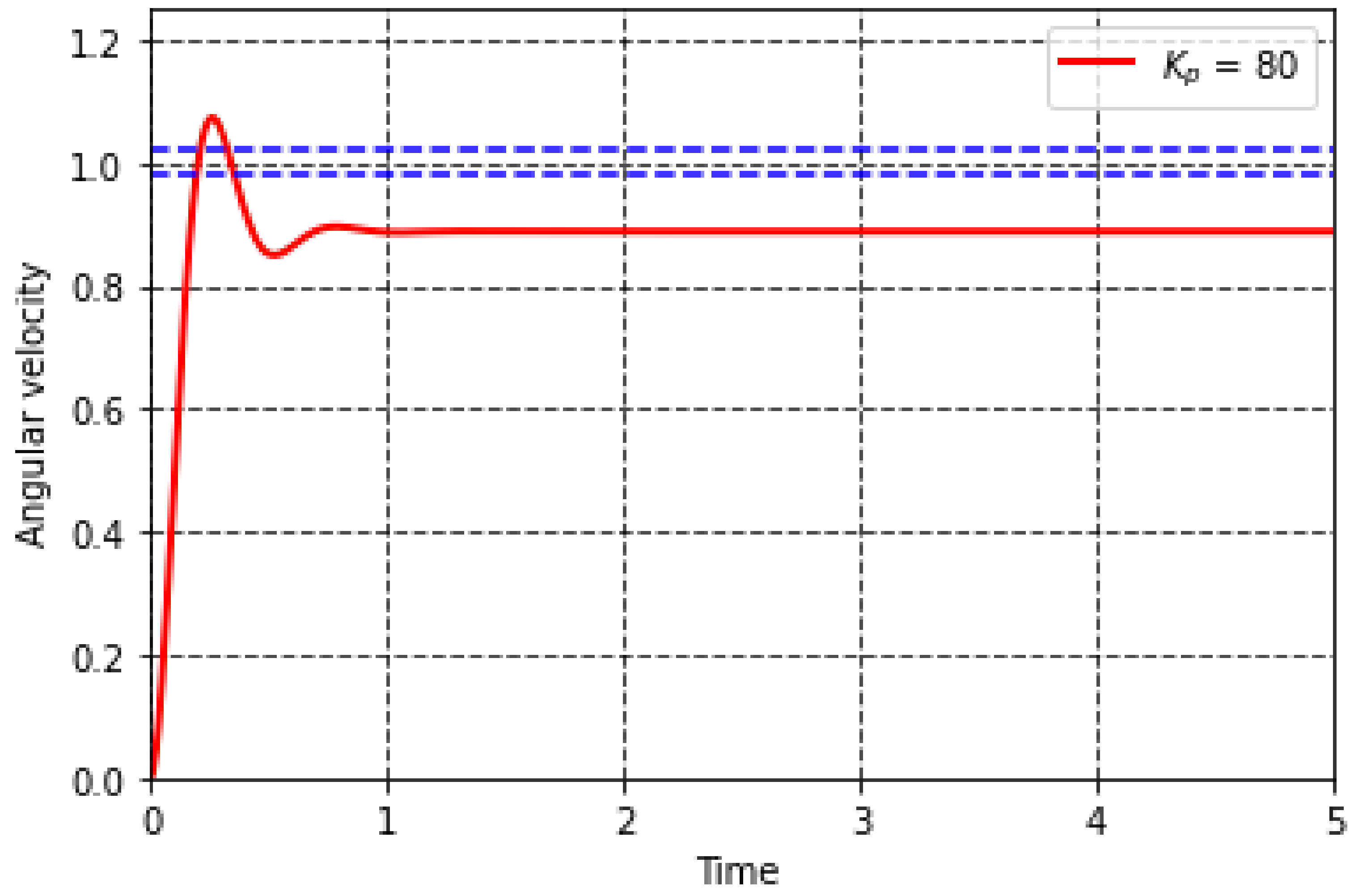
- Settling time less than 2 seconds
- Overshoot less than 5%
- Steady-state error less than 2%

Proportional term

$$u(t) = K_p(r(t) - y(t))$$

The design criteria are the following:

- **Settling time less than 2 seconds**
- **Overshoot less than 5%**
- **Steady-state error less than 2%**



Proportional term

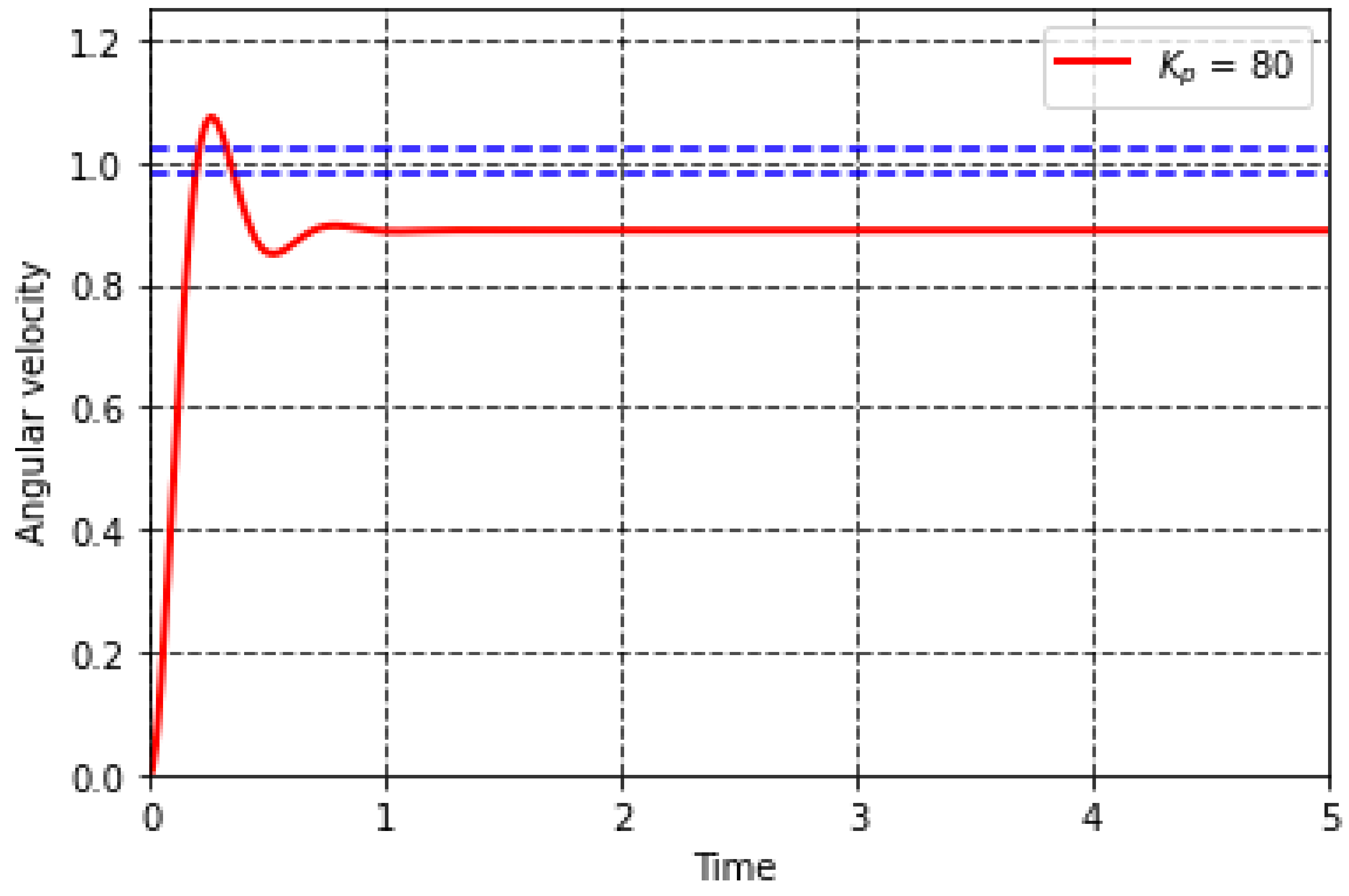
$$u(t) = K_p(r(t) - y(t))$$

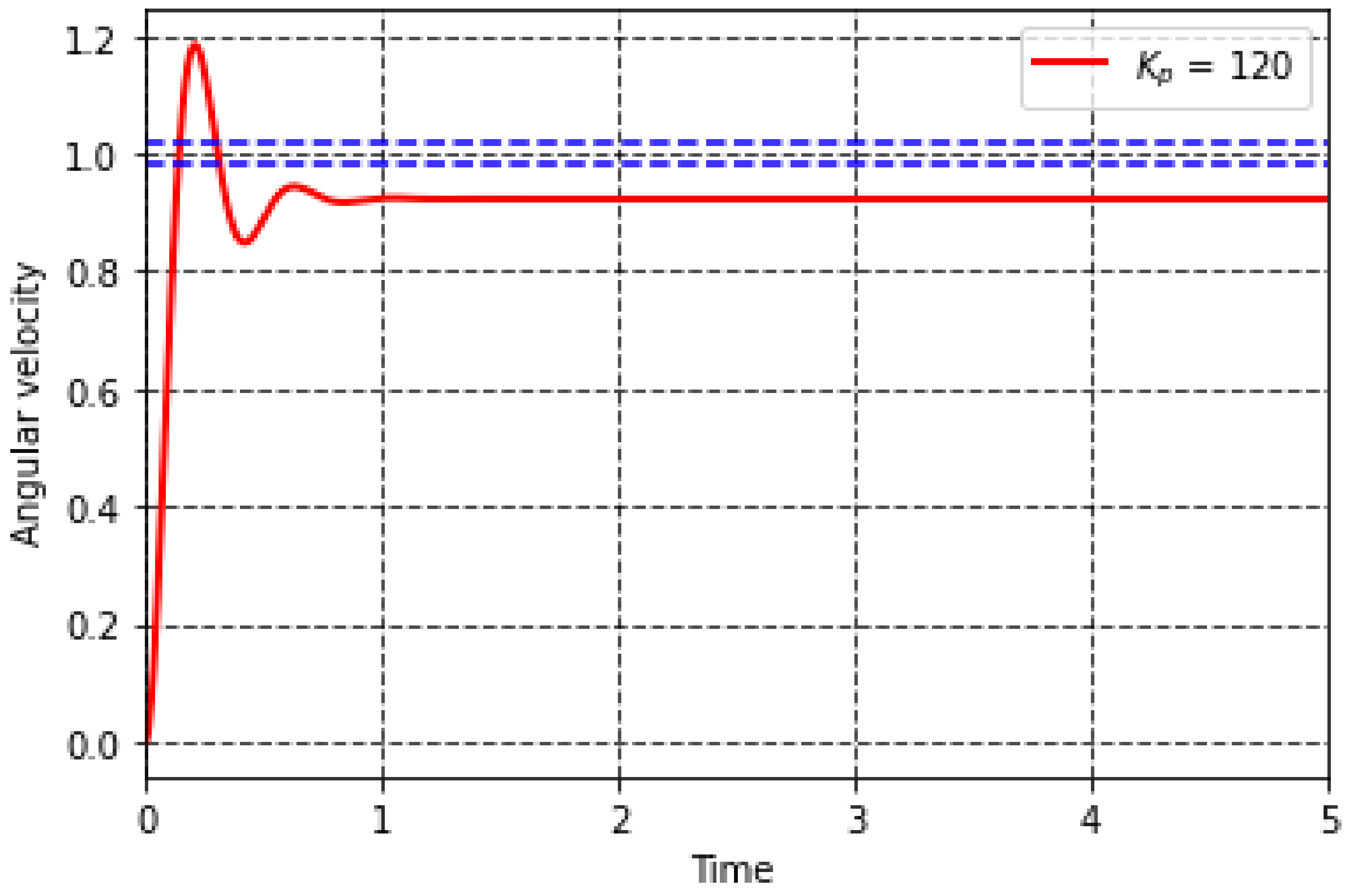
The design criteria are the following:

- **Settling time less than 2 seconds**
- **Overshoot less than 5%**
- **Steady-state error less than 2%**

Proportional term

$$u(t) = K_p(r(t) - y(t))$$





The design criteria are the following:

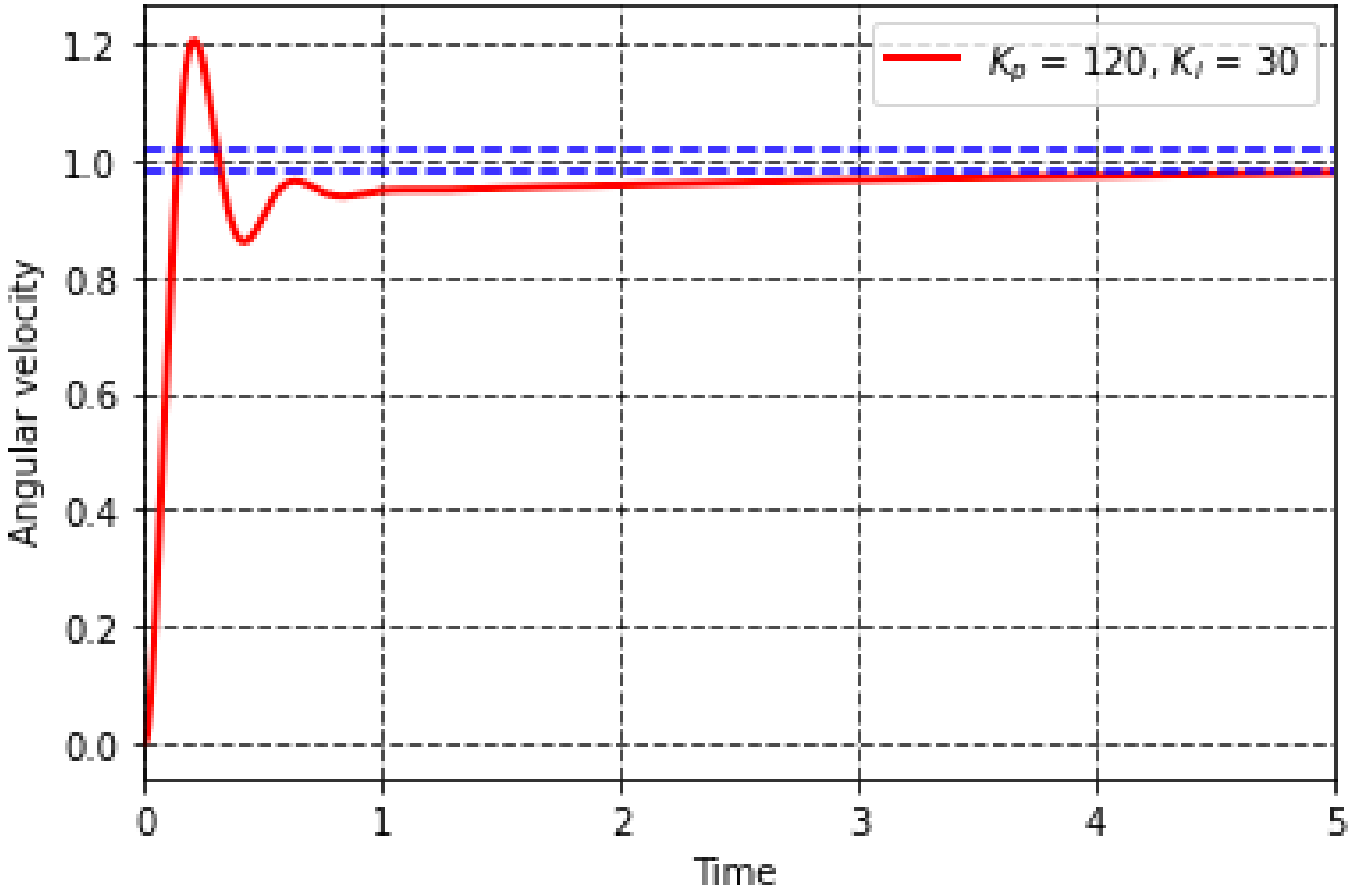
- **Settling time less than 2 seconds**
- **Overshoot less than 5%**
- **Steady-state error less than 2%**

Integral term

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau$$

The design criteria are the following:

- **Settling time less than 2 seconds**
- **Overshoot less than 5%**
- **Steady-state error less than 2%**



Integral term

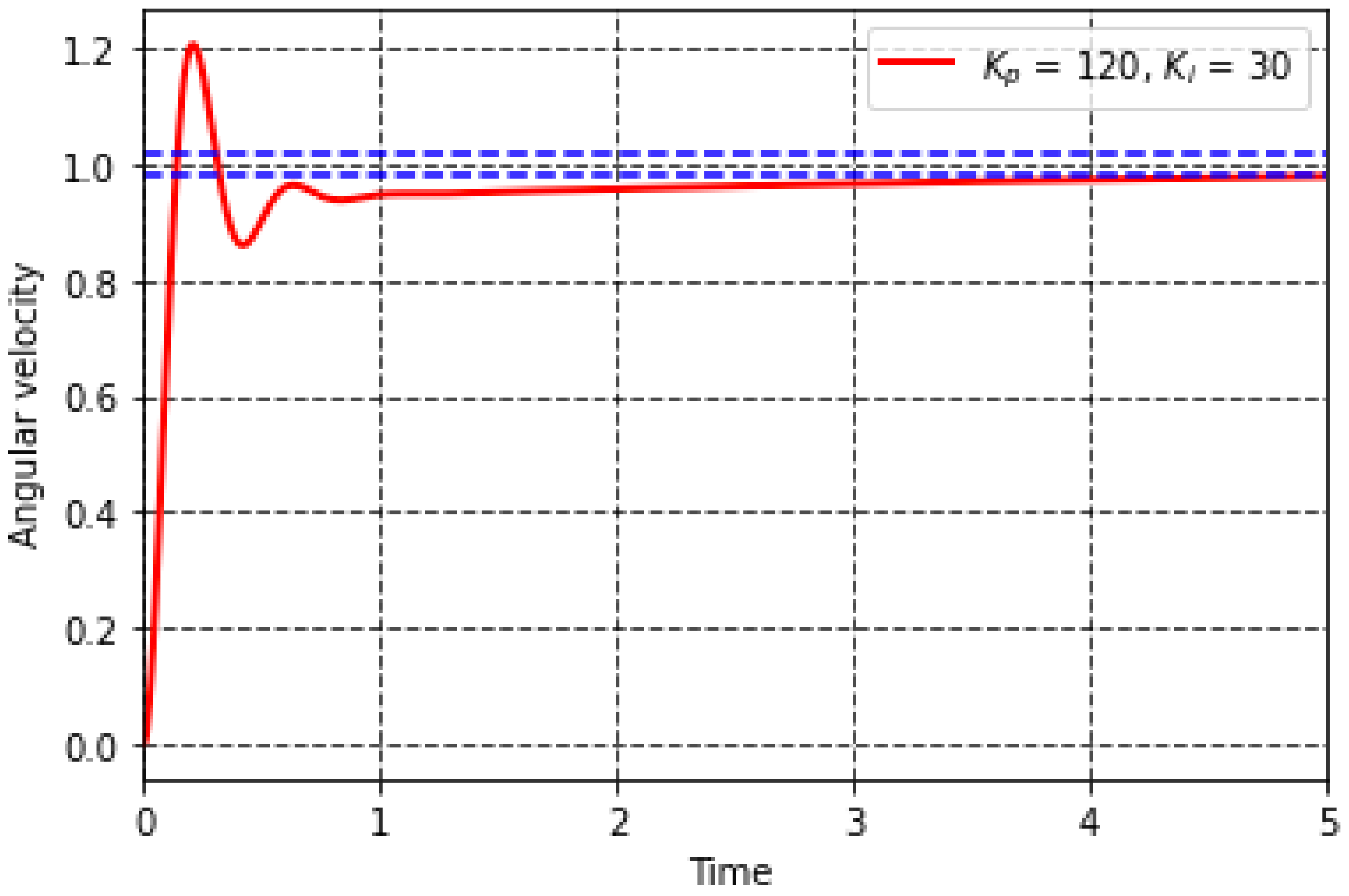
$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau$$

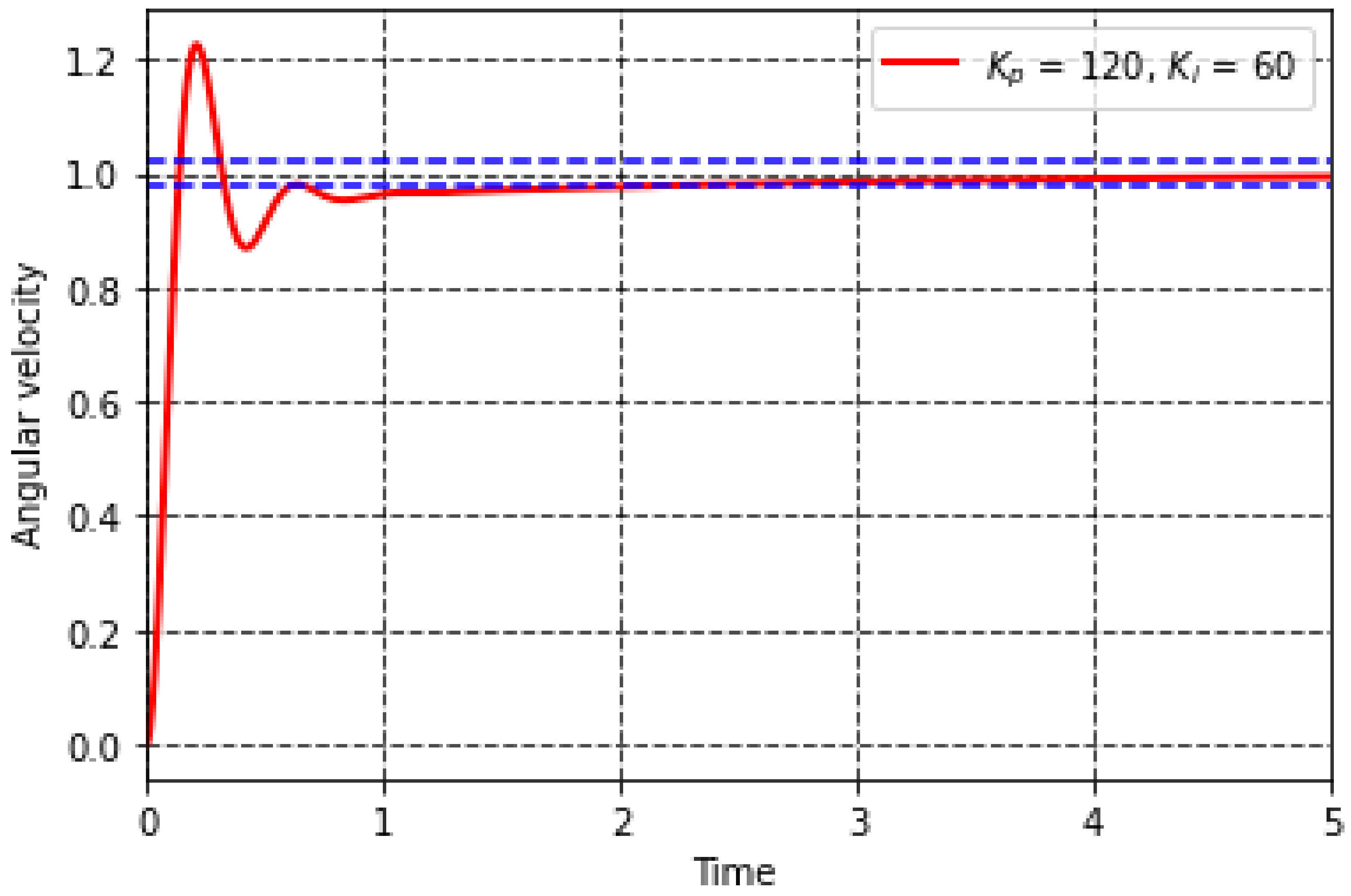
The design criteria are the following:

- **Settling time less than 2 seconds**
- **Overshoot less than 5%**
- **Steady-state error less than 2%**

Integral term

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau$$





The design criteria are the following:

- **Settling time less than 2 seconds**
- **Overshoot less than 5%**
- **Steady-state error less than 2%**

Derivative term

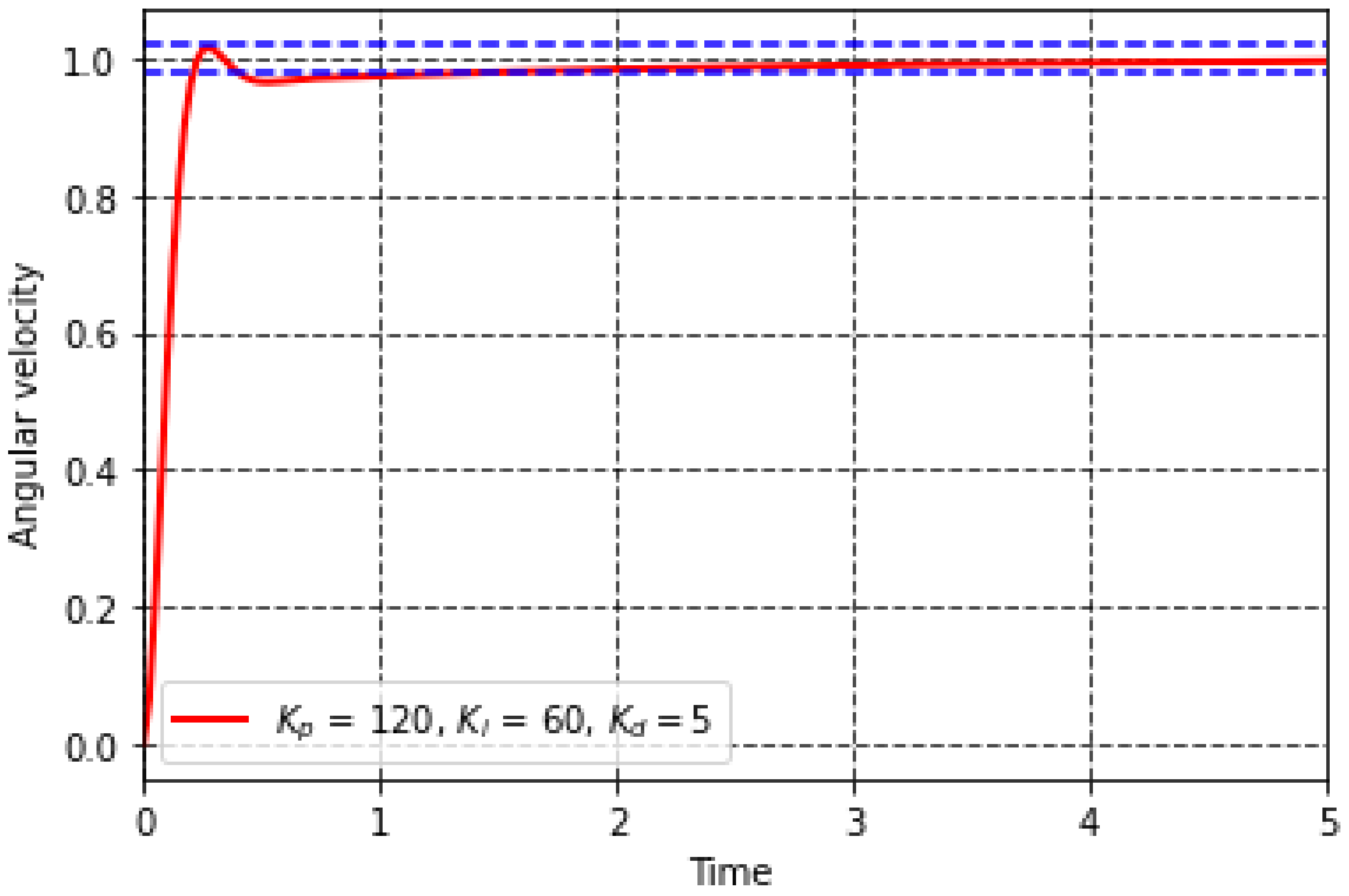
$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau + K_d \dot{e}(t)$$

The design criteria are the following:

- **Settling time less than 2 seconds**
- **Overshoot less than 5%**
- **Steady-state error less than 2%**

Derivative term

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau + K_d \dot{e}(t)$$

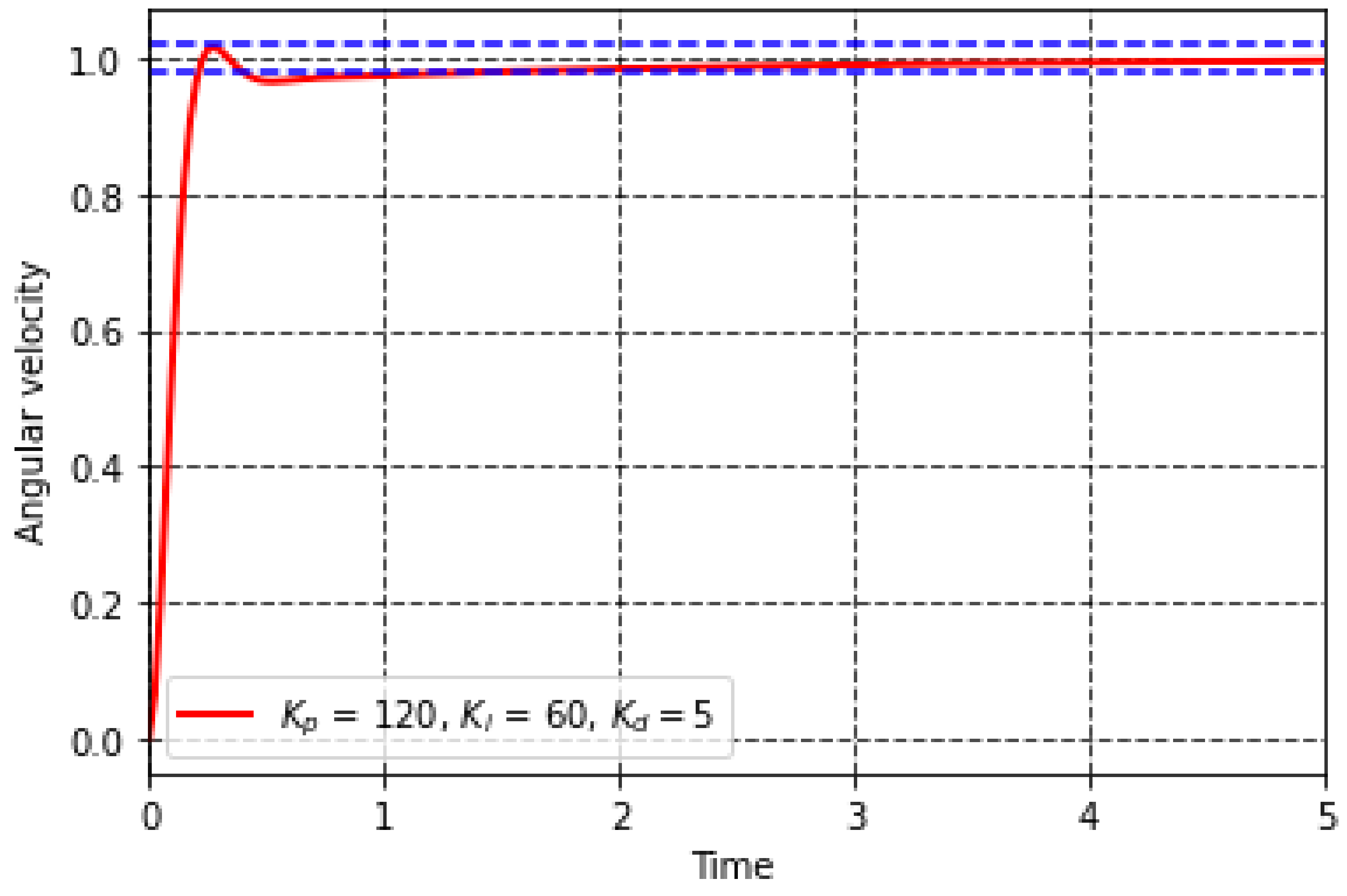


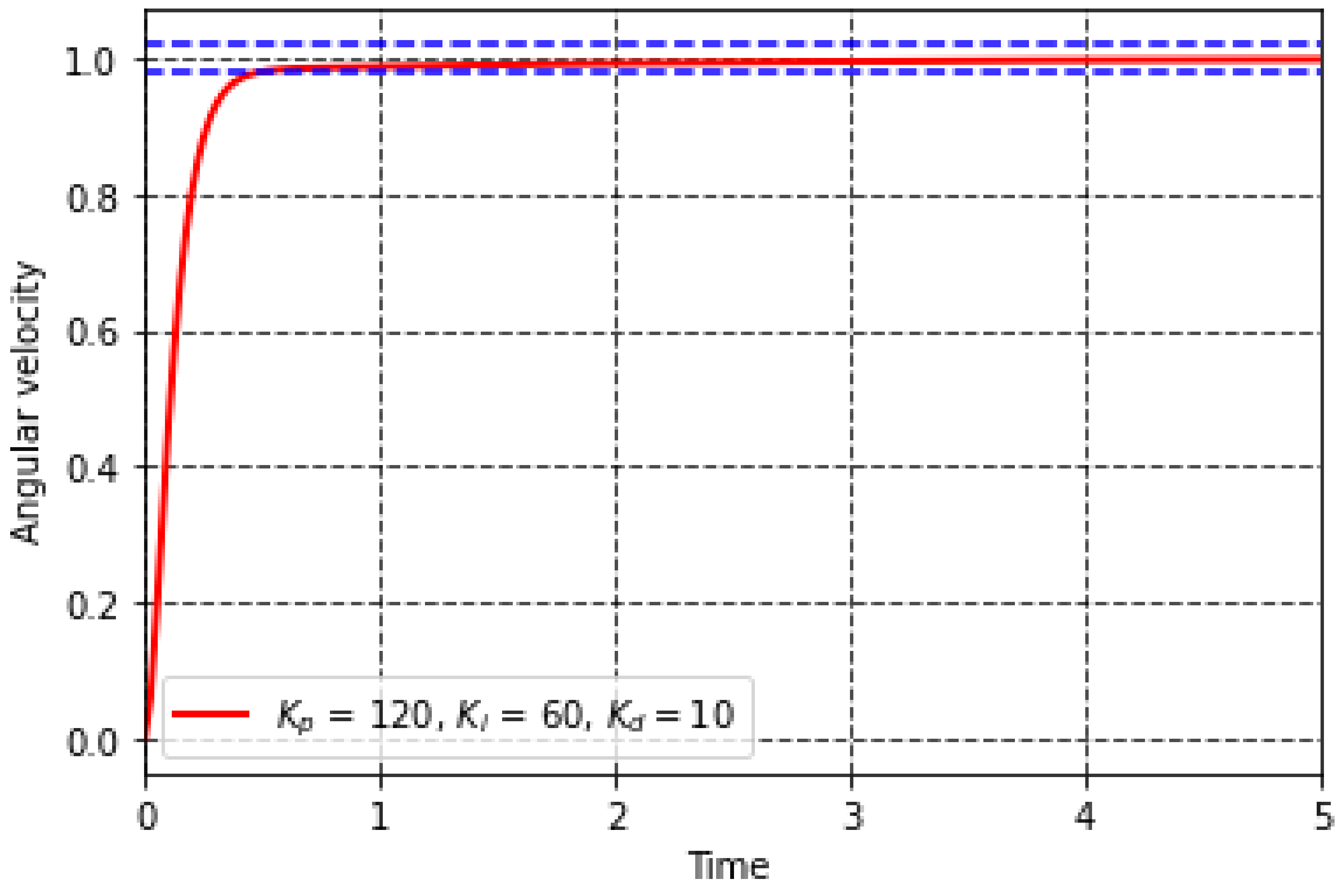
The design criteria are the following:

- **Settling time less than 2 seconds**
- **Overshoot less than 5%**
- **Steady-state error less than 2%**

Derivative term

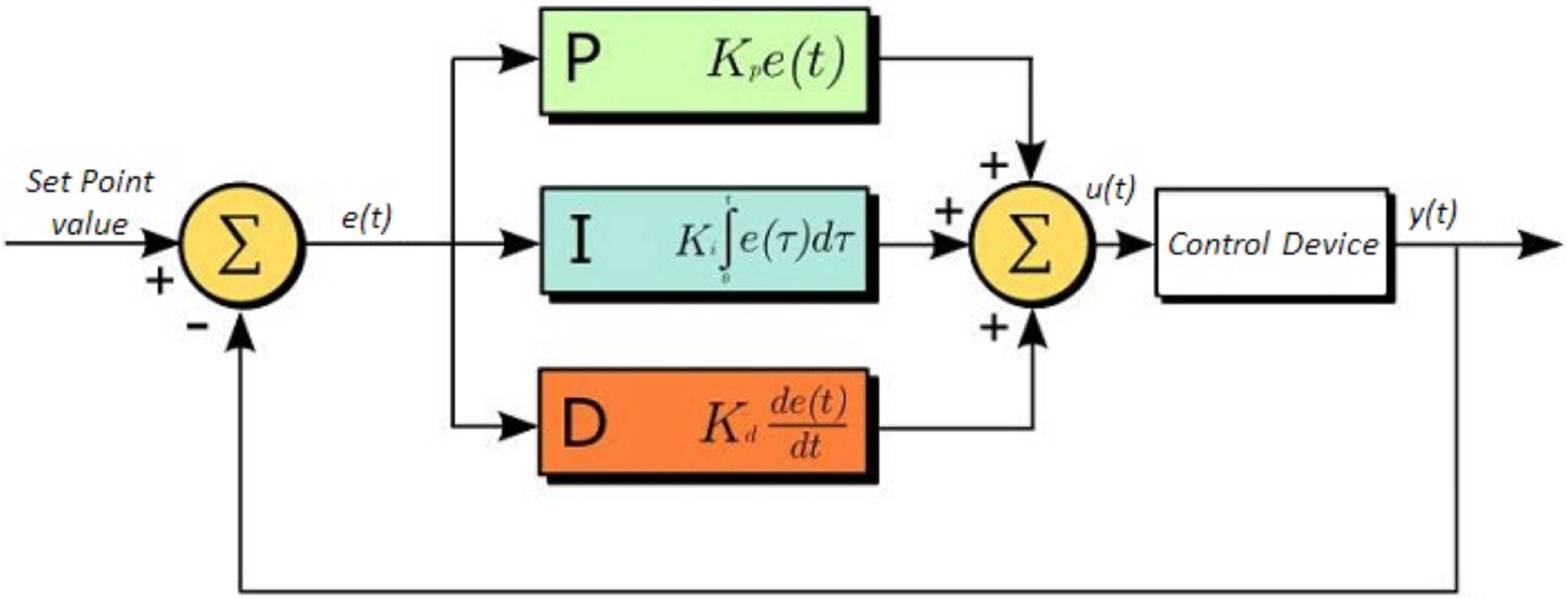
$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau + K_d \dot{e}(t)$$





PID controller

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau + K_d \dot{e}(t)$$



Manual Tuning PID

- Start with Kp, Ki, and Kd at 0.
- Increase Kp until steady-state error is very low.
- Increase Ki until steady-state error is removed entirely.
- Increase Kd until oscillations are removed.

Tuning PID

CL RESPONSE	RISE TIME	OVERSHOOT	SETTLING TIME	S-S ERROR
increase Kp	Decrease	Increase	Small Change	Decrease
increase Ki	Decrease	Increase	Increase	Decrease
increase Kd	Small Change	Decrease	Decrease	No Change

Ziegler–Nichols Tuning

The **Ziegler–Nichols tuning method** is a heuristic method of tuning a PID controller. It was developed by John G. Ziegler and Nathaniel B. Nichols. It is performed by setting the *I* (integral) and *D* (derivative) gains to zero. The "P" (proportional) gain, K_p is then increased (from zero) until it reaches the **ultimate gain** K_u, at which the output of the control loop has stable and consistent oscillations. K_u and the oscillation period T_u are then used to set the P, I, and D gains depending on the type of controller used and behaviour desired:

Ziegler–Nichols method[1]

Control Type	K_p	T_i	T_d	K_i	K_d
P	$0.5K_u$	–	–	–	–
PI	$0.45K_u$	$0.8\bar{3}T_u$	–	$0.54K_u/T_u$	–
PD	$0.8K_u$	–	$0.125T_u$	–	$0.10K_uT_u$
classic PID[2]	$0.6K_u$	$0.5T_u$	$0.125T_u$	$1.2K_u/T_u$	$0.075K_uT_u$
Pessen Integral Rule[2]	$0.7K_u$	$0.4T_u$	$0.15T_u$	$1.75K_u/T_u$	$0.105K_uT_u$
some overshoot[2]	$0.3\bar{3}K_u$	$0.50T_u$	$0.3\bar{3}T_u$	$0.6\bar{6}K_u/T_u$	$0.1\bar{1}K_uT_u$
no overshoot[2]	$0.20K_u$	$0.50T_u$	$0.3\bar{3}T_u$	$0.40K_u/T_u$	$0.06\bar{6}K_uT_u$

The ultimate gain (K_u) is defined as 1/M, where M = the amplitude ratio, $K_i = K_p/T_i$ and $K_d = K_pT_d$.

These guidelines hold in many cases, but not all...

If you truly want to know the effect of tuning the individual gains, you will have to do more analysis, or will have to perform testing on the actual system.

These guidelines hold in many cases, but not all...

If you truly want to know the effect of tuning the individual gains, you will have to do more analysis, or will have to perform testing on the actual system.

Nice video to summarise

https://www.youtube.com/watch?v=4Y7zG48uHRo