Control Theory

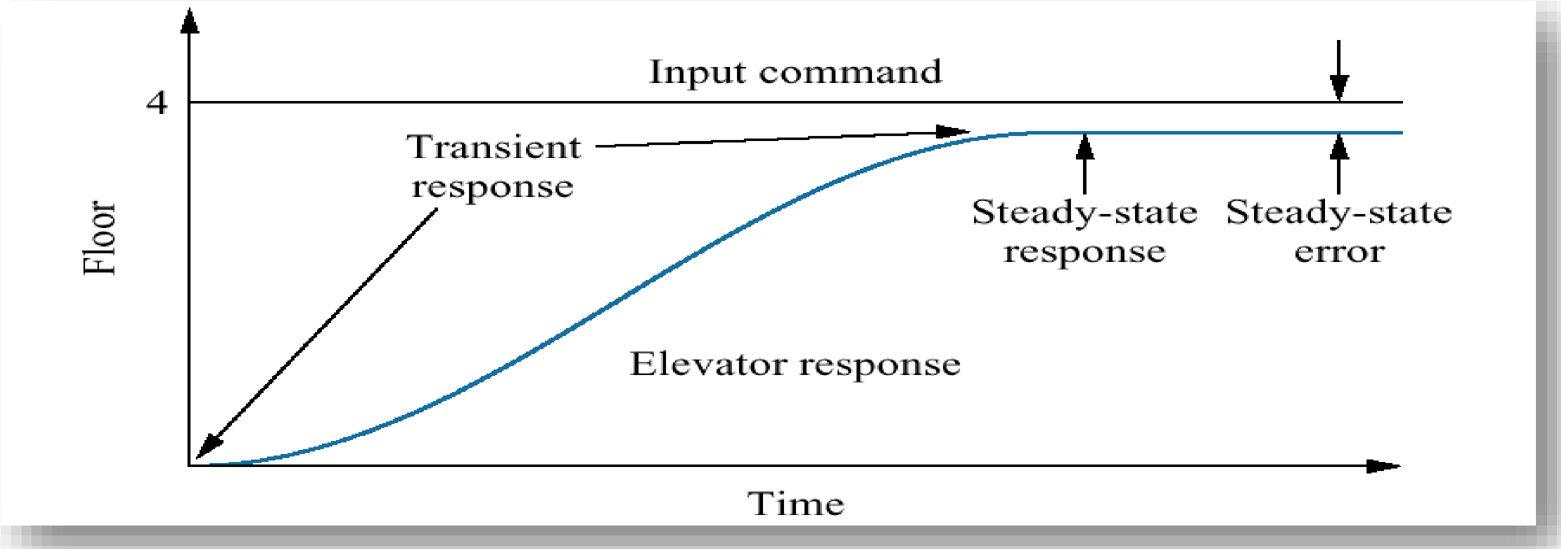
Agenda

- Terms
- Closed loop and open loop systems
- Dynamical systems and feedback
- The PID controller
  - Theory
  - Tuning
  - Implementation

Terms

- System, plant or process
  - To be controlled
- Actuators
  - Converts the control signal to a power signal
- Sensors
  - Provides measurement of the system output
- Reference input (Set-point)
  - Represents the desired output

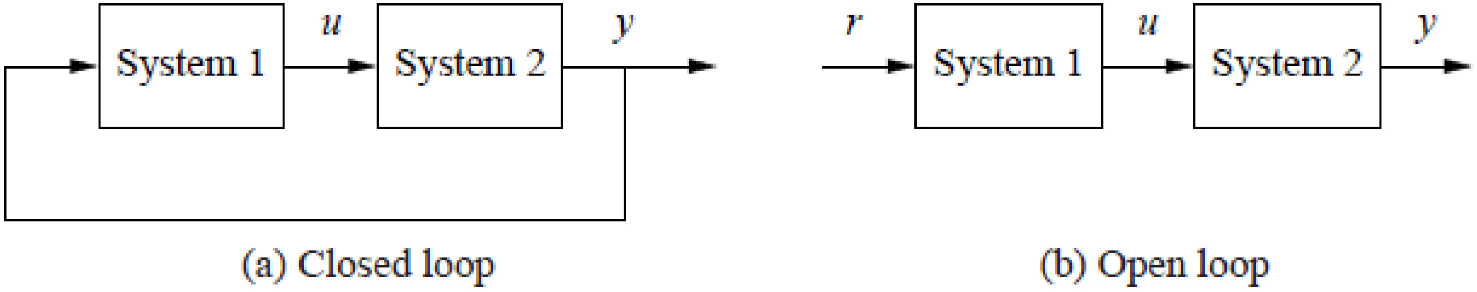
Response characteristics



- Transient response:
  - Gradual change of output from initial to the desired condition
- Steady-state response:
  - Approximation to the desired response
- Steady-state error:
  - Difference between desired condition and steady-state response.

- For example, consider an elevator rising from ground to the 4th floor.

Closed loop and Open loop systems



(a) Closed loop

(b) Open loop

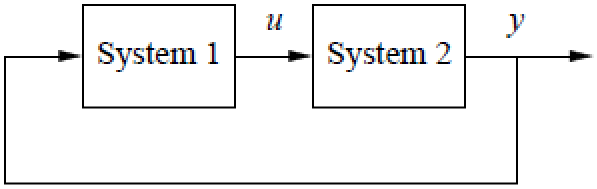
- (a) The output of system 1 is used as the input of system 2, and the output of system 2 becomes the input of system 1, creating a closed loop system.
- (b) The interconnection between system 2 and system 1 is removed, and the system is said to be open loop.

Source: Feedback Systems An Introduction for Scientists and Engineers

AARHUS UNIVERSITY
SCHOOL OF ENGINEERING

Dynamical systems and feedback

- A *dynamical system* is a system whose behavior changes over time, often in response to external stimulation or forcing.
- The term *feedback* refers to a situation in which two (or more) dynamical systems are connected together such that each system influences the other and their dynamics are thus strongly coupled.



(a) Closed loop

Source: Feedback Systems An Introduction for Scientists and Engineers

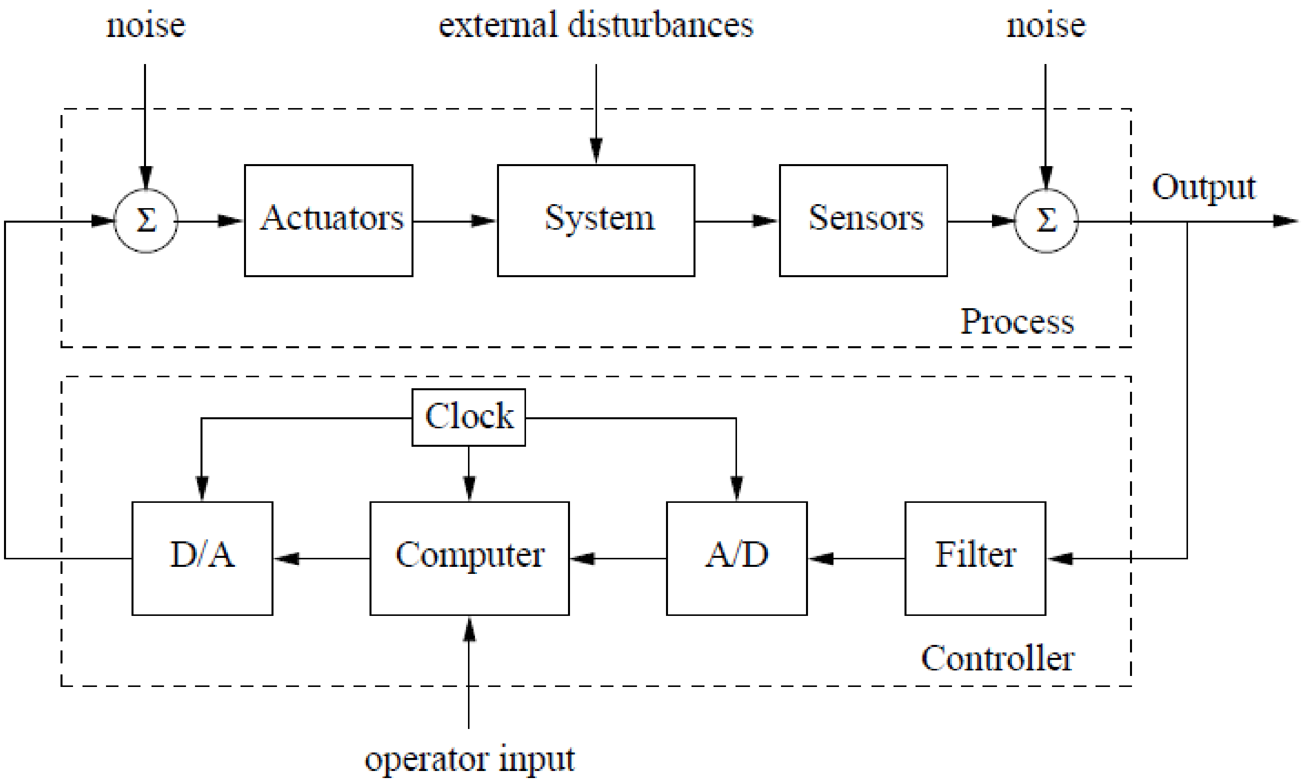
Control

- The use of algorithms and feedback in engineered systems.

- Examples:
  - feedback loops in electronic amplifiers.
  - setpoint controllers in chemical and materials processing.
  - “fly-by-wire” systems on aircraft.
  - router protocols that control traffic flow on the Internet.

Source: Feedback Systems An Introduction for Scientists and Engineers

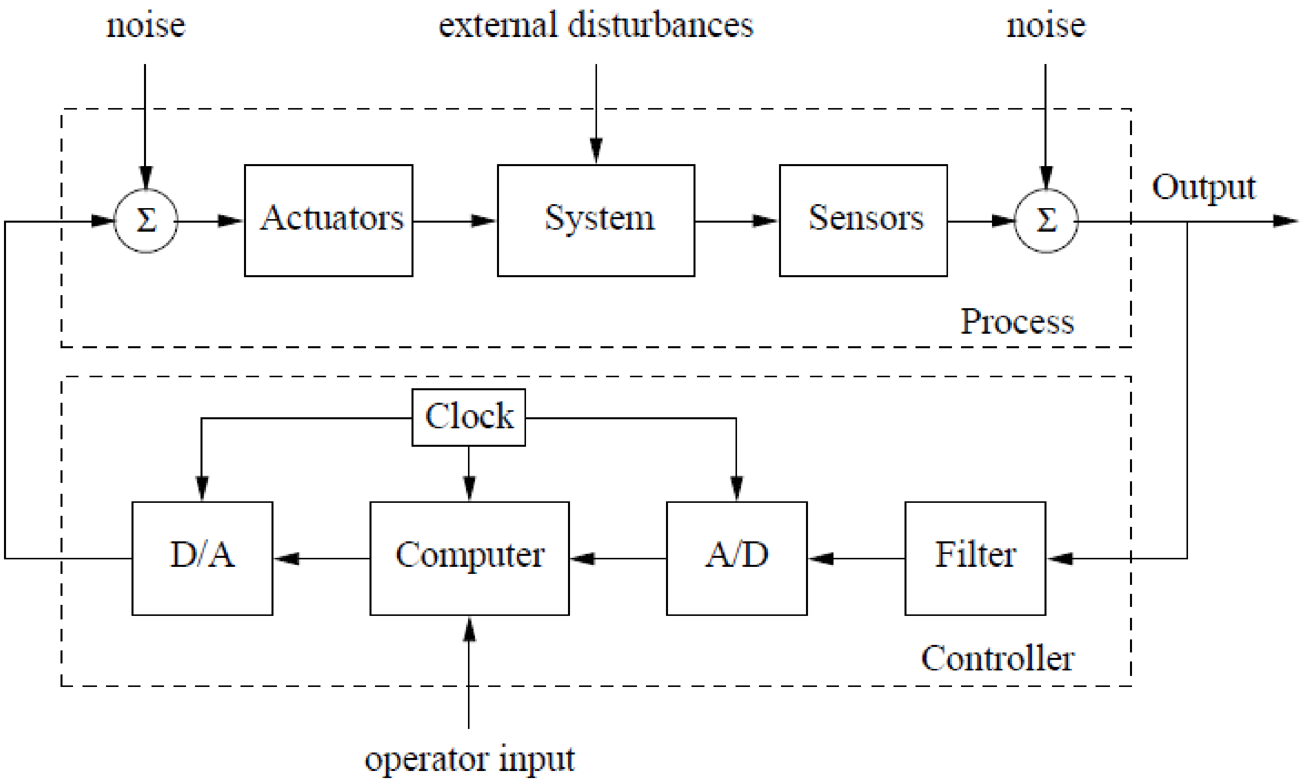
A computer-controlled system



Sensing, Computation and Actuation.

Source: Feedback Systems An Introduction for Scientists and Engineers

A computer-controlled system



- The algorithm that computes the control action as a function of the sensor values is often called a *control law*.
- The system can be influenced externally by an operator who introduces *command signals* to the system.
- External disturbances and noise also influence the system.

Source: Feedback Systems An Introduction for Scientists and Engineers

Negative feedback

- Base correcting actions on the difference between desired and actual performance.

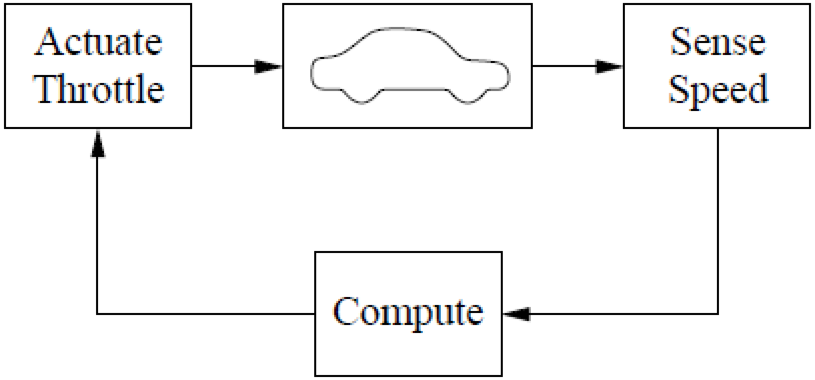
Robustness to uncertainty

- Most real-world systems can not be modelled in all details.

Cruise control

- A simple transfer function (the system model):
  - Car speed = throttle_position * x

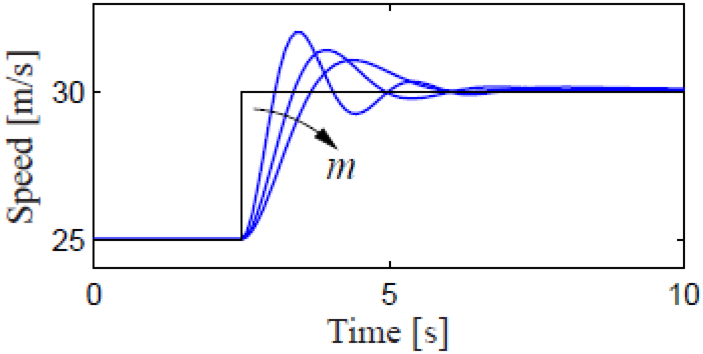
Cruise control with feedback



Actuate Throttle

Sense Speed

Compute



- The figure on the right shows the response of the control system to a commanded change in speed from 25 m/s to 30 m/s.
- The three different curves correspond to differing masses of the vehicle, between 1000 and 3000 kg.

Source: Feedback Systems An Introduction for Scientists and Engineers

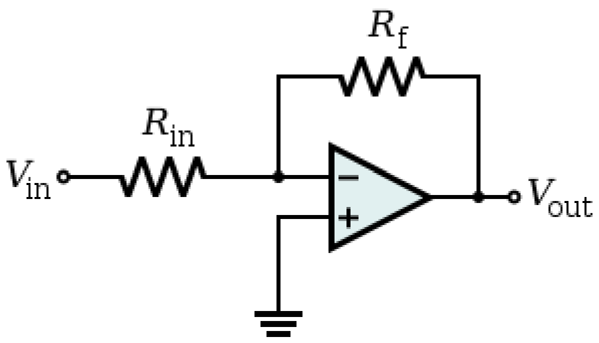
Design of dynamics

- Stabilize an unstable system.
- Make sluggish systems responsive.
- Prevent system drift.

Source: Feedback Systems An Introduction for Scientists and Engineers

Design of dynamics

- By using feedback to create a system whose response matches a desired profile, we can hide the complexity and variability that may be present inside a subsystem.
- This allows us to create more complex systems by not having to simultaneously tune the responses of a large number of interacting components.



Source: Feedback Systems An Introduction for Scientists and Engineers

Unstable systems

- Walking robot
- Segway
- Nuclear reactor
- Airplanes
- ..

Robots!

- https://www.youtube.com/watch?v=rVlhMGQgDkY

Nuclear reactor

Image source:
http://inapcache.boston.com/universal/site_graphics/blogs/bigpicture/chernobyl_25th_anniversary/bp2.jpg

All modern fighter jets are designed to be unstable.

Without a computer to stabilize the jet, it will crash.

Why would you make
a fighter jet unstable?

Drawbacks of feedback

- Possibility of introducing instability.
- Introduction and amplification of measurement noise.
- Increased complexity due to sensors, controller and actuators.

F22 prototype

- https://www.youtube.com/watch?v=faB5bIdksi8

Comparing controllers

- Step response
- Peak related criteria
- Time related criteria

Step response

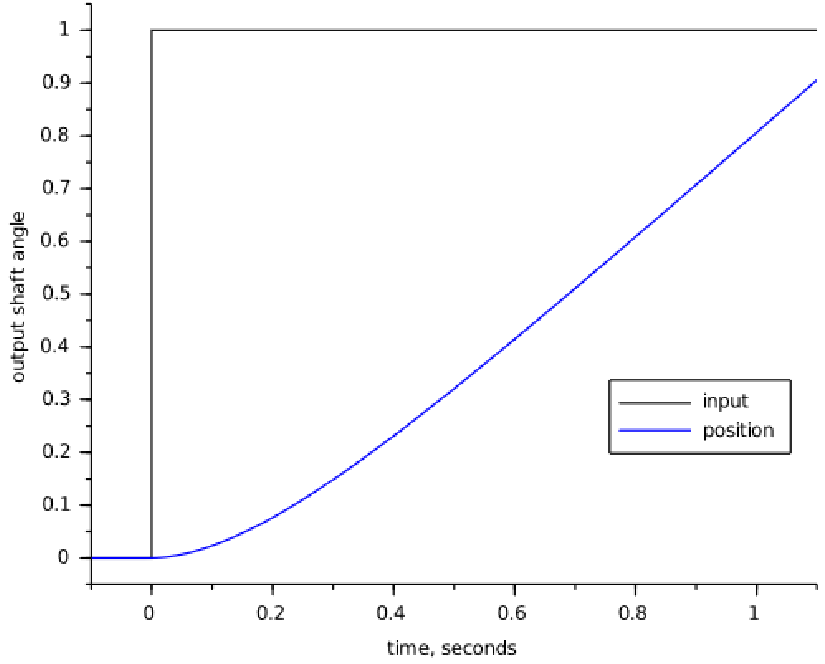
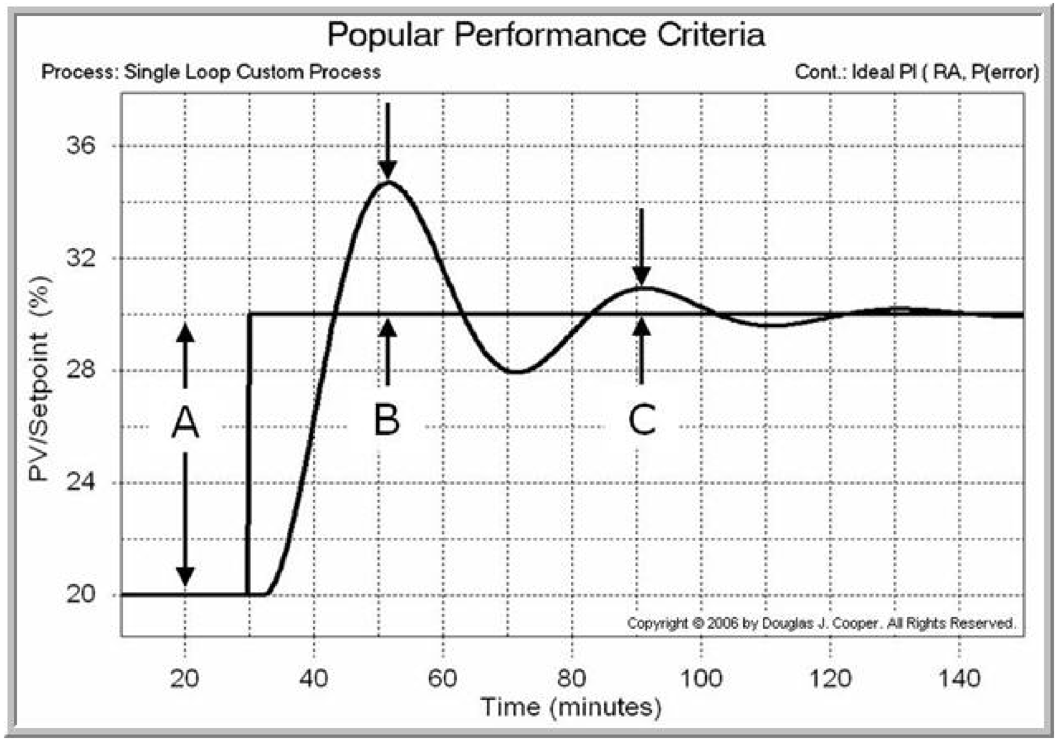


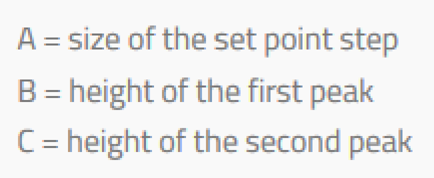
Figure 4: Motor step response.

- The system's step response is the behavior of the system output in response to an input that goes from zero to some constant value at time t = 0.

Source: http://www.wescottdesign.com/articles/pid/pidWithoutAPhd.pdf

Peak related criteria

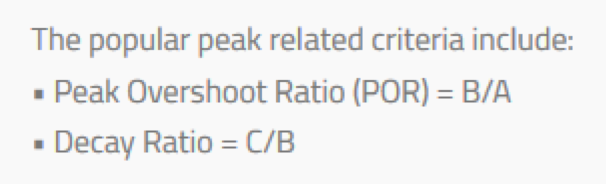




A = size of the set point step

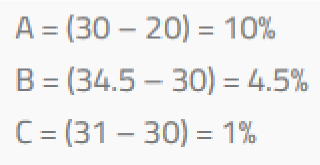
B = height of the first peak

C = height of the second peak



The popular peak related criteria include:

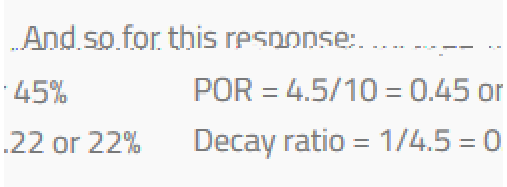
- Peak Overshoot Ratio (POR) = B/A
- Decay Ratio = C/B



A = (30 – 20) = 10%

B = (34.5 – 30) = 4.5%

C = (31 – 30) = 1%



And so for this response:

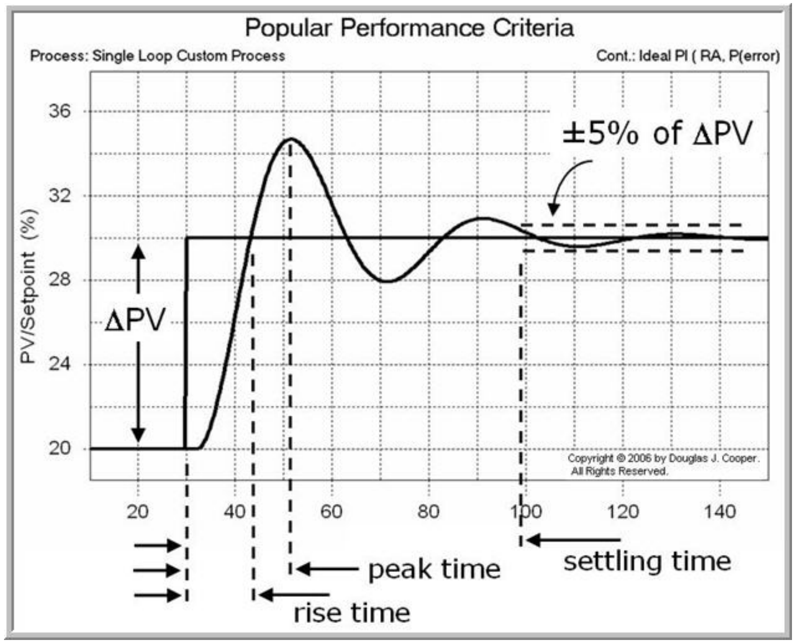
45%   POR = 4.5/10 = 0.45 or

.22 or 22%   Decay ratio = 1/4.5 = 0

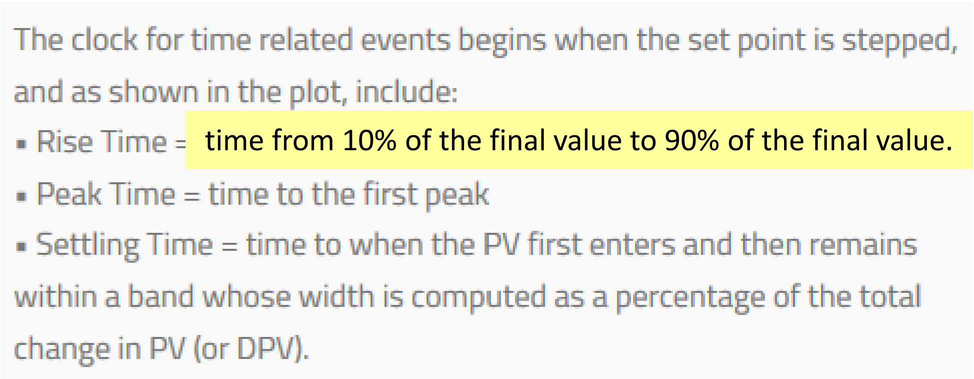
An old rule of thumb is that a 10% POR and 25% decay ratio (sometimes called a quarter decay) are popular values.

Source: http://controlguru.com/comparing-controller-performance-using-plot-data/

Time related criteria

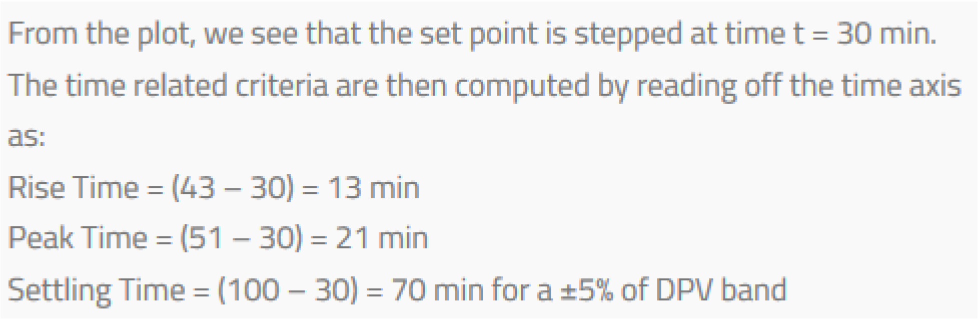


The 5% band used to determine settling time in the plot above was chosen arbitrarily.



The clock for time related events begins when the set point is stepped, and as shown in the plot, include:

- Rise Time = time from 10% of the final value to 90% of the final value.
- Peak Time = time to the first peak
- Settling Time = time to when the PV first enters and then remains within a band whose width is computed as a percentage of the total change in PV (or DPV).



From the plot, we see that the set point is stepped at time t = 30 min. The time related criteria are then computed by reading off the time axis as:

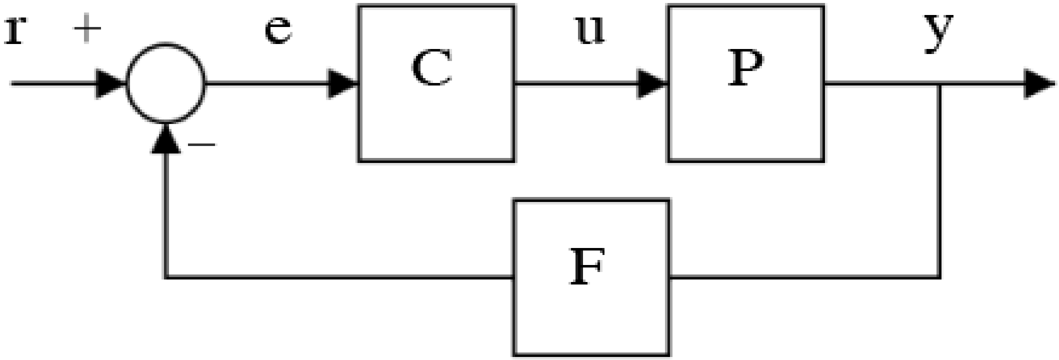
Rise Time = (43 – 30) = 13 min

Peak Time = (51 – 30) = 21 min

Settling Time = (100 – 30) = 70 min for a ±5% of DPV band

Source: http://controlguru.com/comparing-controller-performance-using-plot-data/

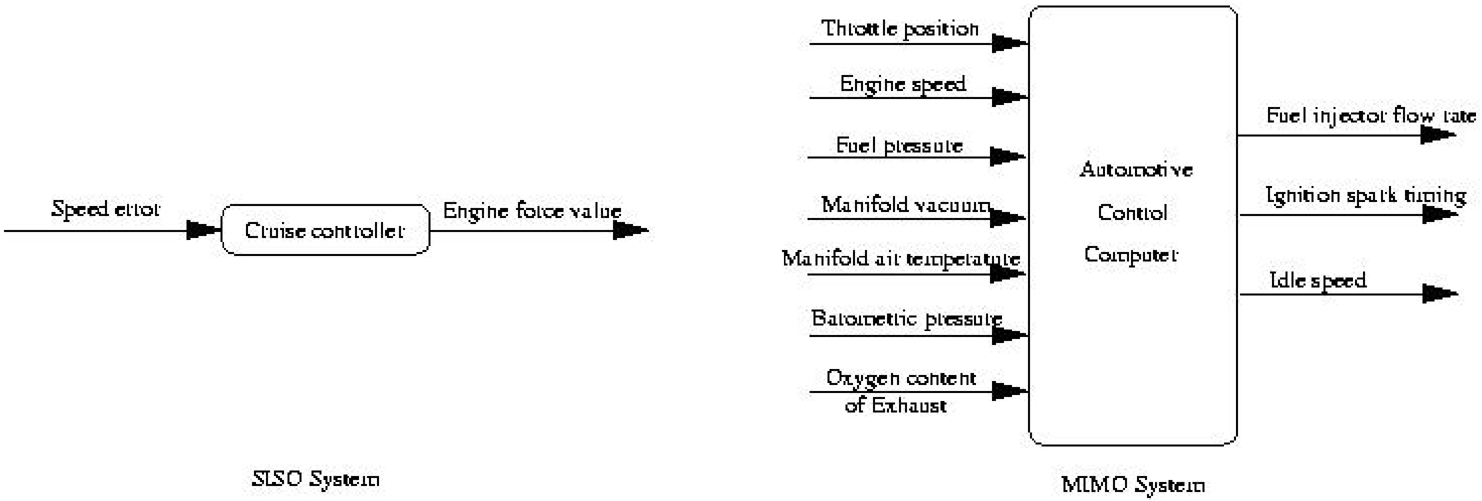
How do you design a feedback controller?



- The output of the system **y** is fed back through a sensor measurement **F** to a comparison with the reference value **r**.
- The controller **C** then takes the error **e** (difference) between the reference and the output to change the inputs **u** to the system under control **P**.
- This kind of controller is a closed-loop controller or feedback controller.

Source: https://en.wikipedia.org/wiki/Control_theory

SISO and MIMO

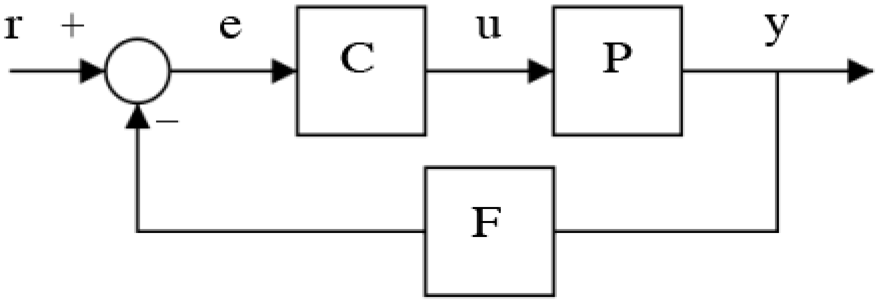


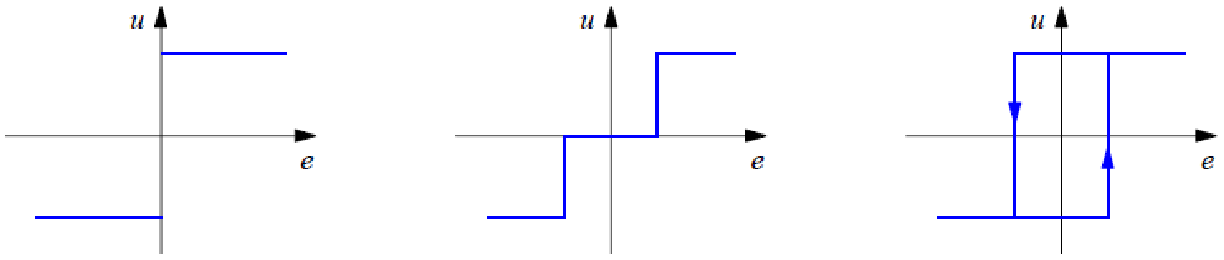
Note: The number of states (internal) of the system is independent of whether the system is SISO or MIMO. The SISO–MIMO terminology is concerned only with the inputs and outputs, not the internal dynamics.

- We will only consider Single-Input, Single-Output (SISO) systems.
- Multiple-Input, Multiple-Output (MIMO) systems are more complex to control (input and output are typically vectors).

Image source: http://www.cds.caltech.edu/~murray/courses/cds101/fa02/faq/02-10-02_freqmimo.html

On/off control





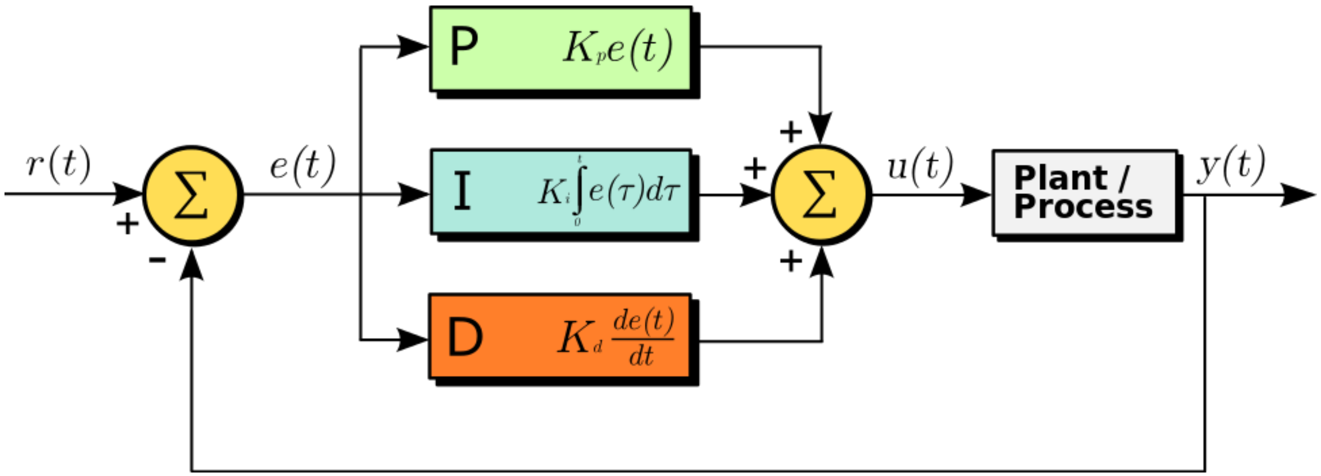
(a) On-off control (b) Dead zone (c) Hysteresis

$$u = \begin{cases} u_{\max} & \text{if } e > 0 \\ u_{\min} & \text{if } e < 0, \end{cases}$$

- where:
  - r = reference
  - y = output
  - e = error = r – y
  - u = control signal

Source: Feedback Systems An Introduction for Scientists and Engineers

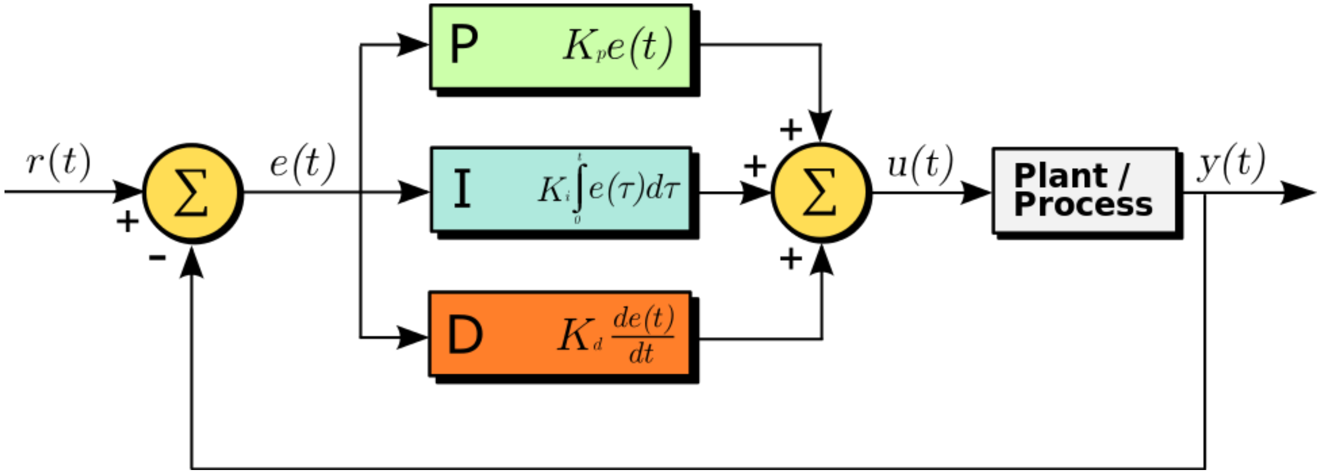
The PID controller



- Simple to understand and implement.
- Widely used.

More than 95% of all industrial control problems are solved by PID control, although many of these controllers are actually *proportional-integral* (PI) *controllers* because derivative action is often not included.
[Source: Feedback Systems An Introduction for Scientists and Engineers]

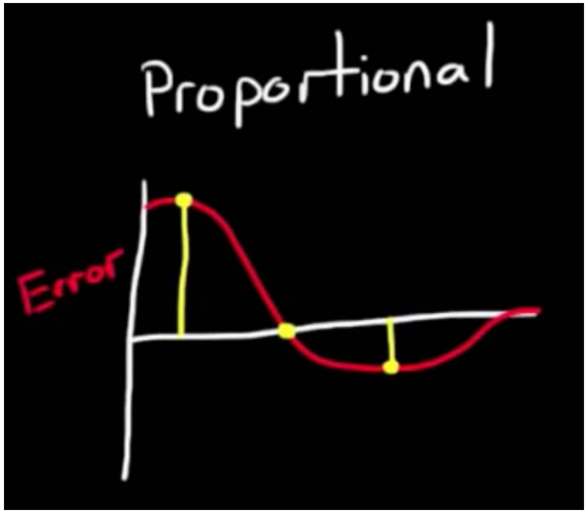
The PID controller



$$u(t) = k_p e(t) + k_i \int_0^t e(\tau)\, d\tau + k_d \frac{de(t)}{dt}.$$

- The three parts of the controller processes the error differently, with different gains:
  - **P**roportional, Kp
  - **I**ntegral, Ki
  - **D**erivative, Kd

Proportional



$$u(t) = k_p e(t)$$

- Generates a control signal based on the present error.
- Increasing the proportional gain may produce overshoot, when a sudden (step) change is made to the input.

Proportional

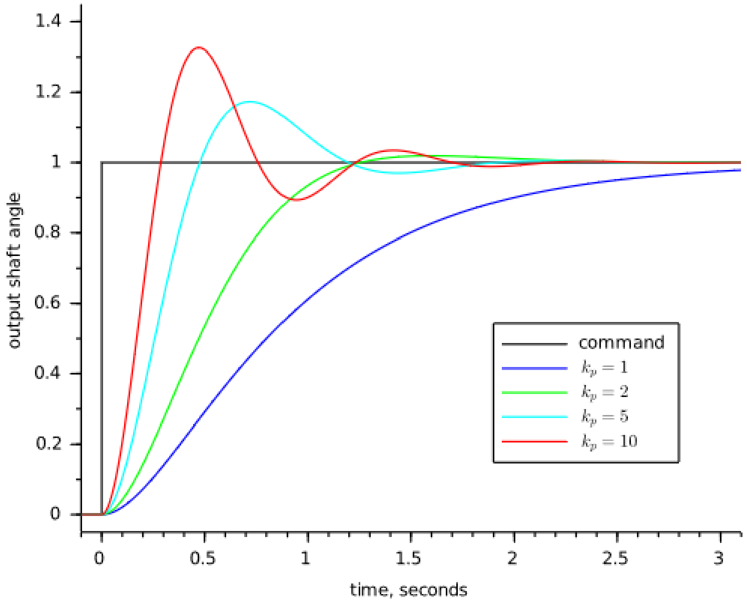
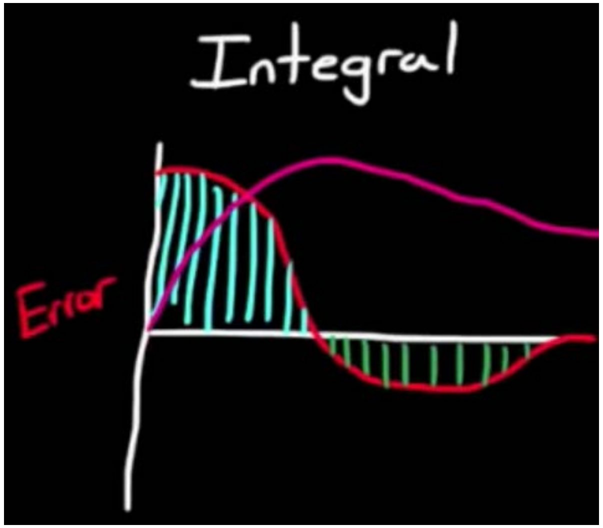


Figure 9: Motor with proportional control.

- The system may settle at a level less than the reference (if a constant control signal is needed to maintain the current state).
- Increasing the gain k brings the response closer to the reference but introduces a damped oscillation.
- To overcome the problem of a constant error at steady state, we introduce the integral controller term.

Image source: http://www.wescottdesign.com/articles/pid/pidWithoutAPhd.pdf

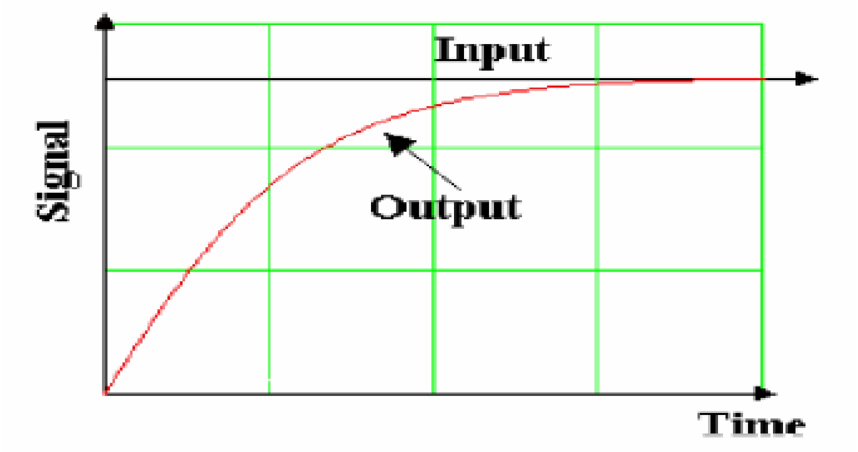
Integral



$$u(t) = k_i \int_0^t e(\tau) d\tau.$$

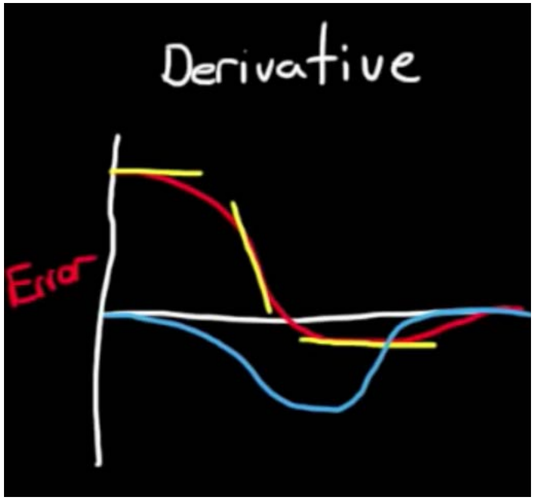
- Generates a control signal based on the sum of the past errors.

Integral



- The integral control action will cause the output to increase over time until the system responds and reduces the error to zero.
- A pure integral controller will reach the setpoint (but probably slowly).
- Integral windup can occur if the control signal goes into saturation.
  - A solution is to limit the integral max and min values.
  - Or stop integrating.

Derivative

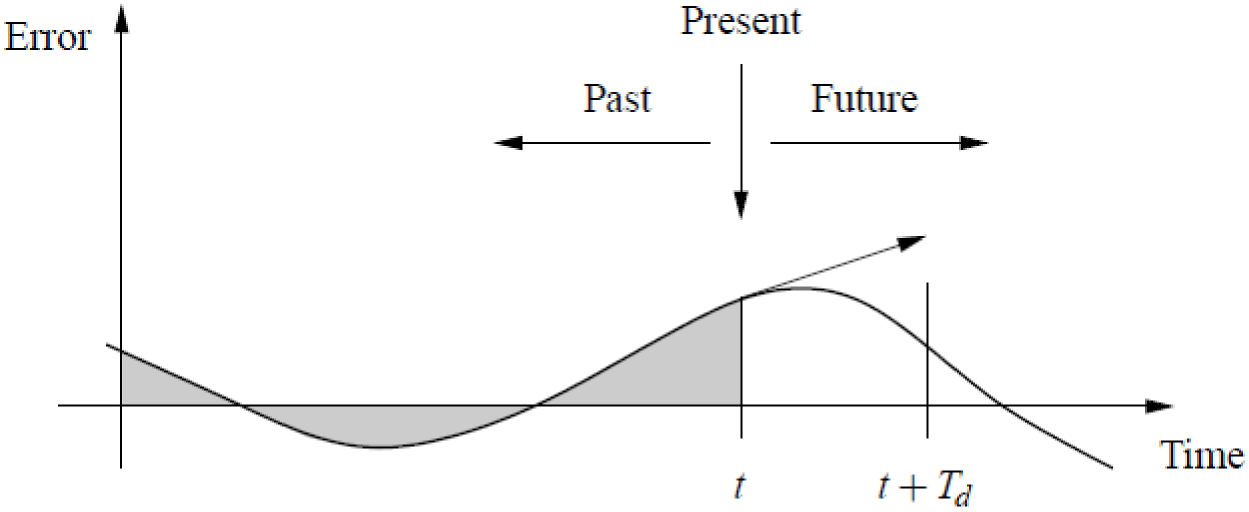


$$e(t+T_d) \approx e(t) + T_d \frac{de(t)}{dt},$$

$$u(t) = k_d \frac{de(t)}{dt}$$

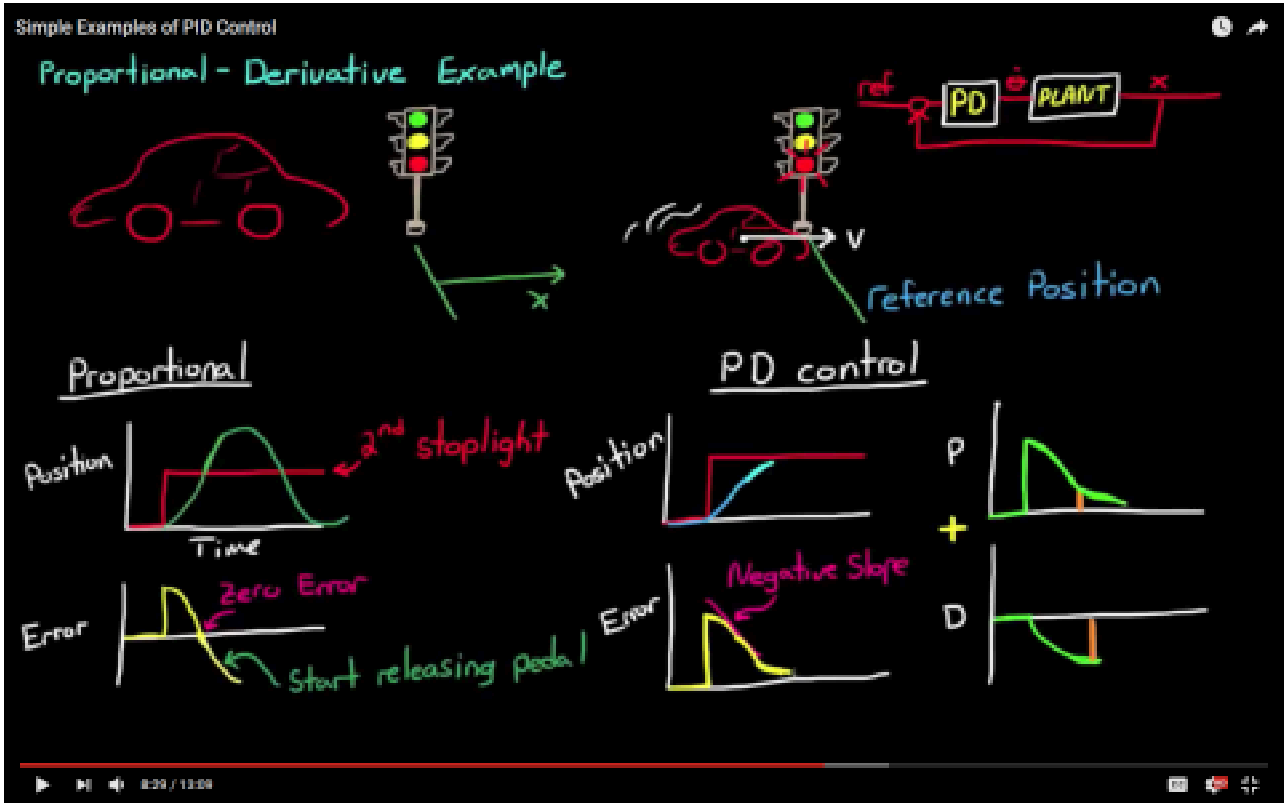
- Generates a control signal based on the rate of change of the error, I.e. the expected future error.
- As the error reduces, the rate of change of error also reduces and the system is slowed down in anticipation of arriving at the correct level. This form of control enables quicker response without overshoot.

PID



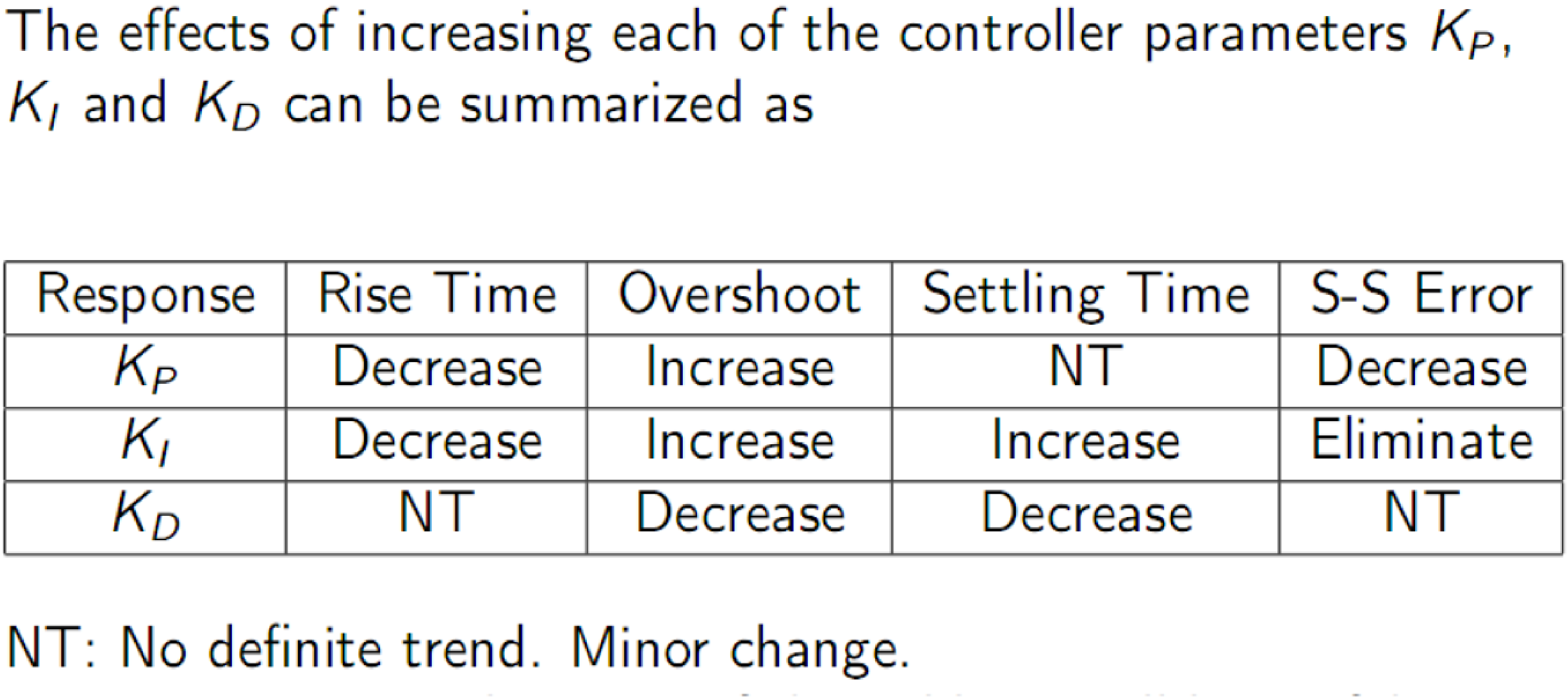
$$u(t) = k_p e(t) + k_i \int_0^t e(\tau)\,d\tau + k_d \frac{de(t)}{dt}.$$

A PID example



You can watch this example at home:
https://www.youtube.com/watch?v=XfAt6hNV8XM

PID tuning



The effects of increasing each of the controller parameters K_P, K_I and K_D can be summarized as

Response	Rise Time	Overshoot	Settling Time	S-S Error
K_P	Decrease	Increase	NT	Decrease
K_I	Decrease	Increase	Increase	Eliminate
K_D	NT	Decrease	Decrease	NT

NT: No definite trend. Minor change.

Source: http://faculty.mercer.edu/jenkins_he/documents/TuningforPIDControllers.pdf

PID tuning

Typical steps for designing a PID controller are

1. Determine what characteristics of the system needs to be improved.
2. Use K_P to decrease the rise time.
3. Use K_D to reduce the overshoot and settling time.
4. Use K_I to eliminate the steady-state error.

This works in many cases, but what would be a good starting point? What if the first parameters we choose are totally crappy? Can we find a good set of initial parameters easily and quickly?

Source: http://faculty.mercer.edu/jenkins_he/documents/TuningforPIDControllers.pdf

PID tuning

- Brute force aka. trial and error
- Ziegler-Nichols
- Lambda tuning

- Mathematical methods:
  - Mathematical representation of the plant
  - Root locus methods
  - State space equations
  - Laplace transforms
  - S-domain calculations
- Others...

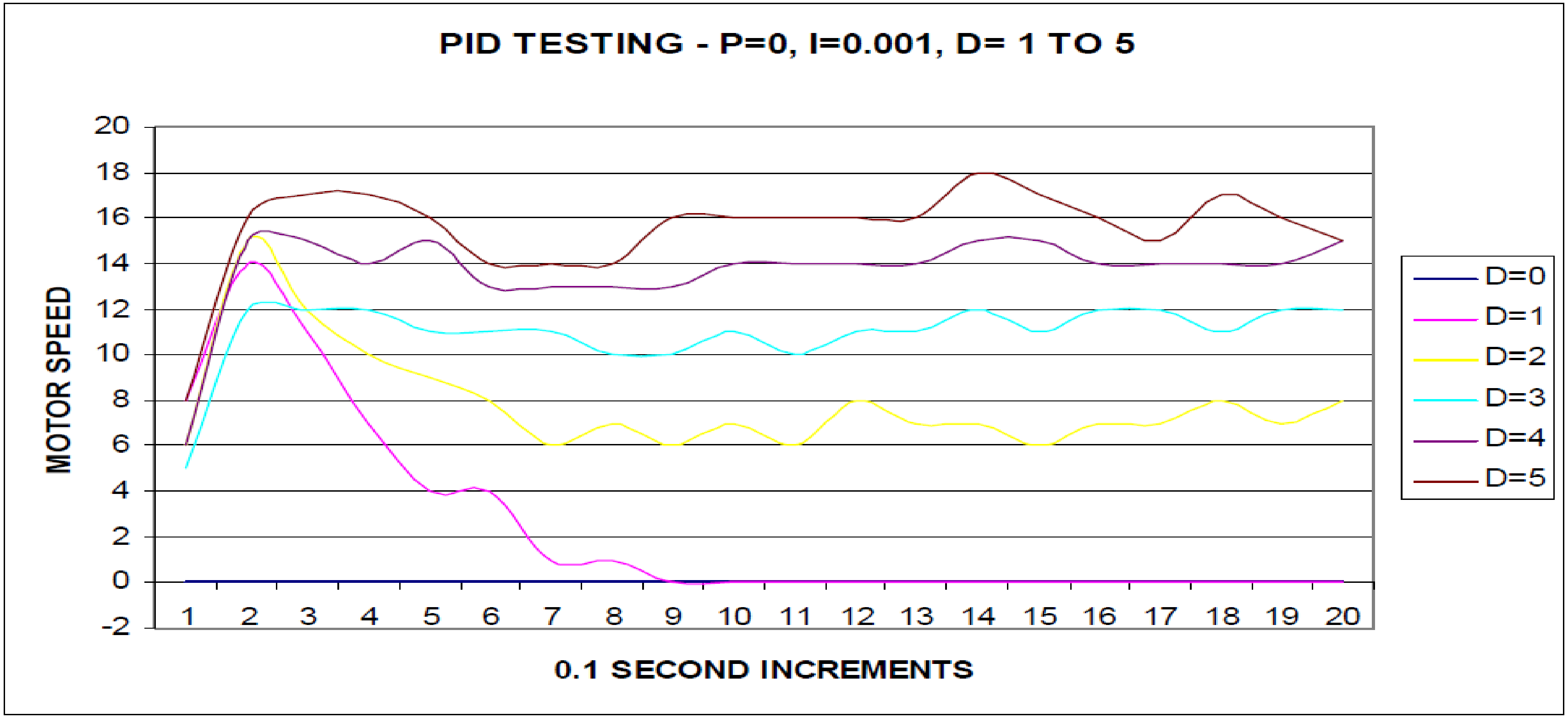
PID Tuning (Brute Force)

- Add code to monitor the output of the PID algorithm (i.e. encoder speed feedback, counts per PID)
- Store the feedback speed value into an array element for the first 20 PID executions. (2 seconds)
- Change the set speed from 0 to 60% of the motor’s maximum speed. (30 counts per PID) This is equivalent to a step function.
- After 2 seconds, stop the motor and print the array data to the serial port.
- This allows the response of the platform to be determined numerically.

PID Tuning (Brute Force) results

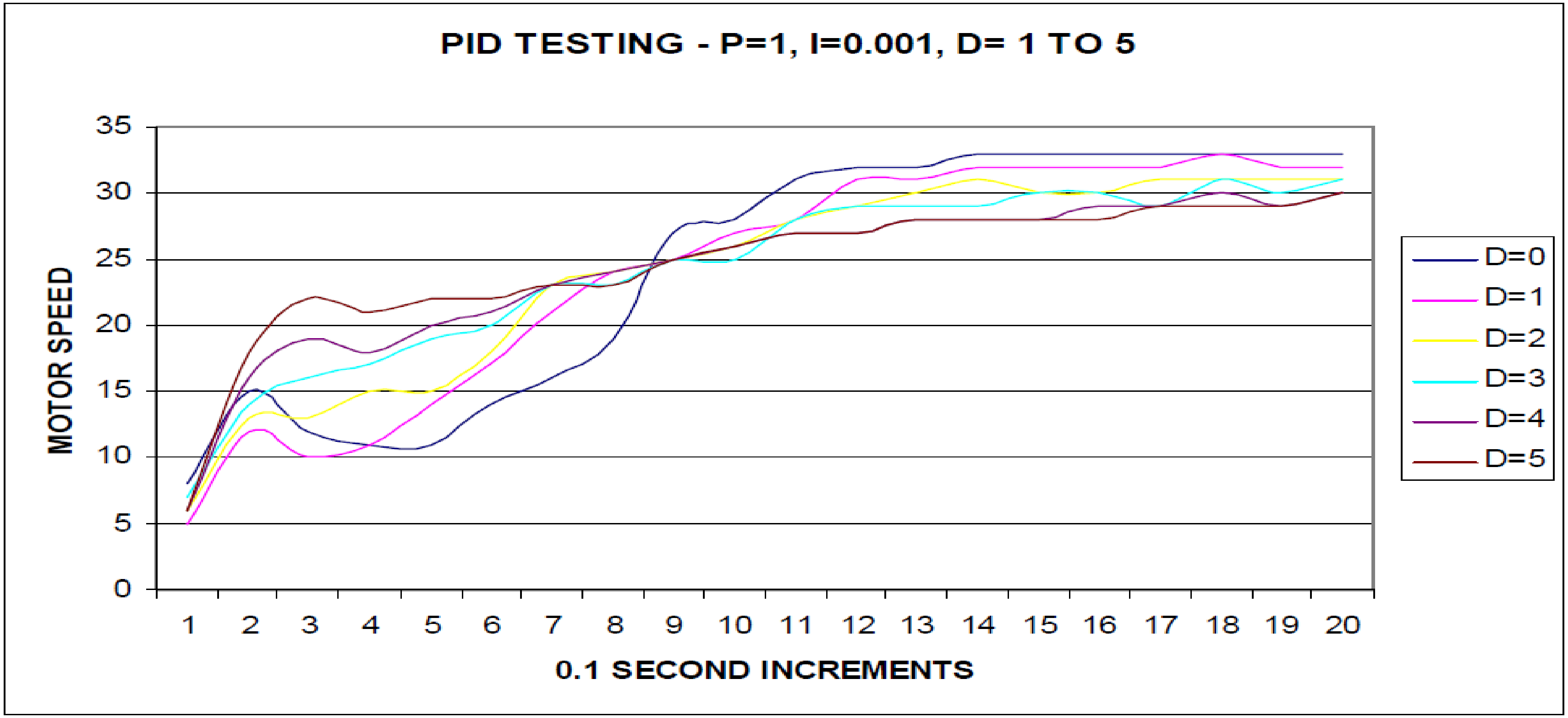
- Now the results of all PID values within the test range are plotted with respect to time.
- The values which yield the best curve will be used for the PID controller.

PID Tuning (Brute Force) chart 1



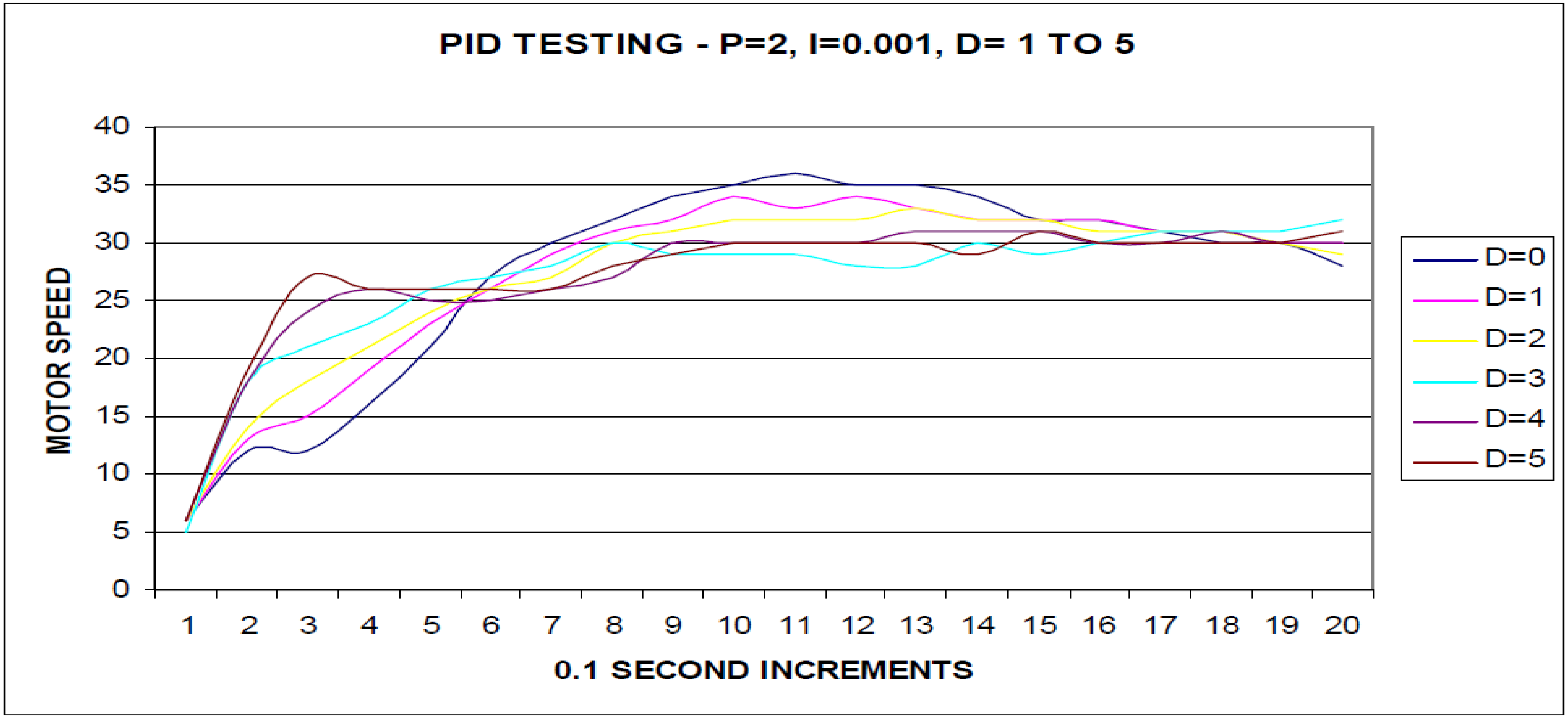
AARHUS UNIVERSITY
SCHOOL OF ENGINEERING

PID Tuning (Brute Force) chart 2



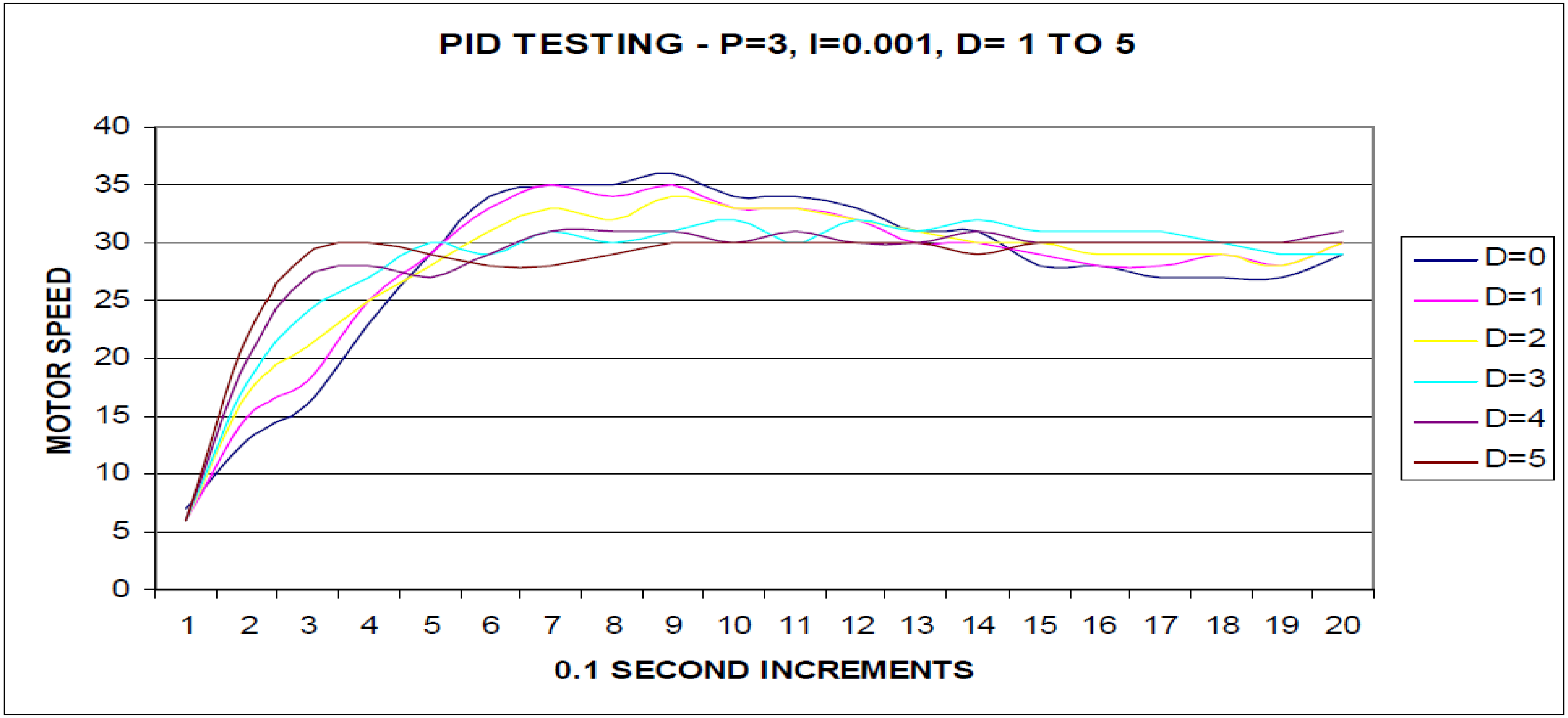
AARHUS UNIVERSITY
SCHOOL OF ENGINEERING

PID Tuning (Brute Force) chart 3

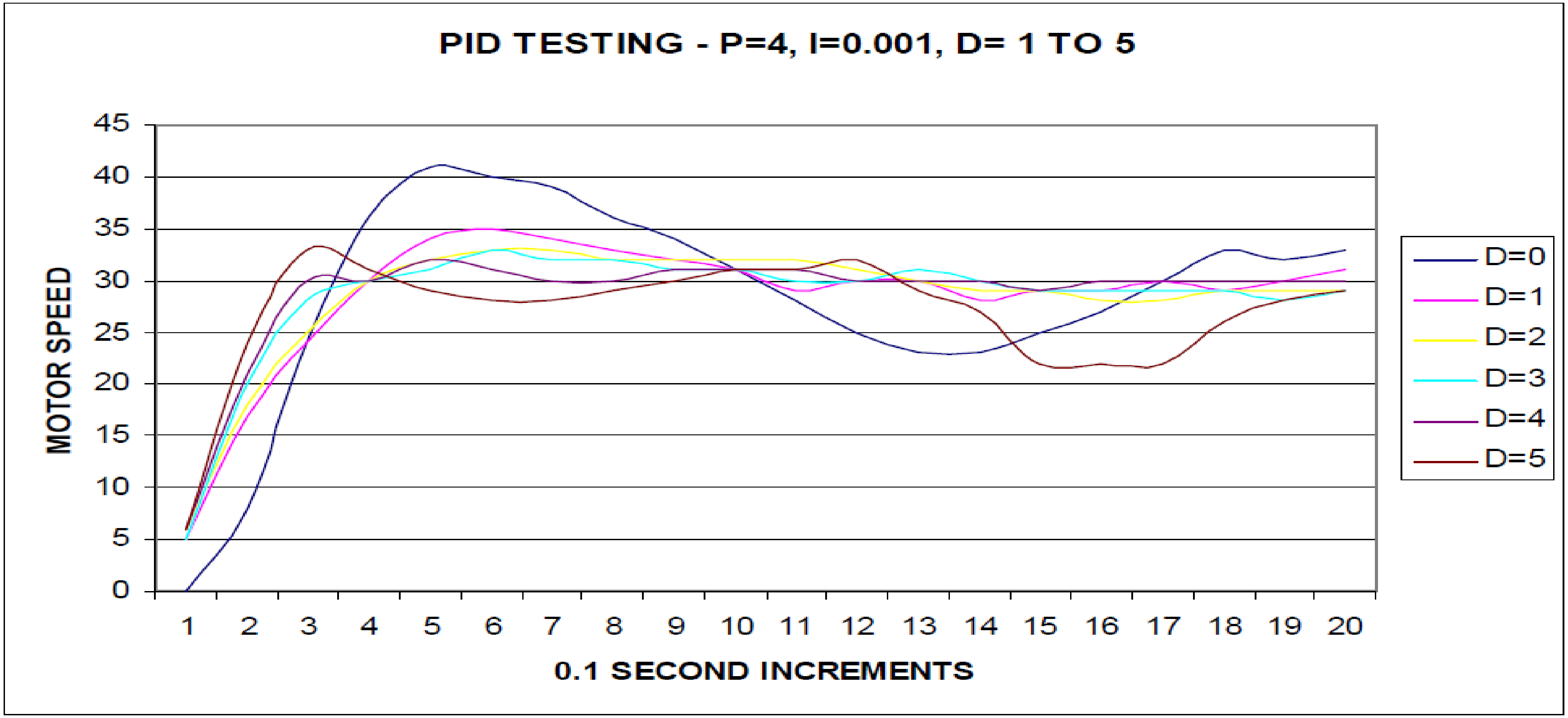


AARHUS UNIVERSITY
SCHOOL OF ENGINEERING

PID Tuning (Brute Force) chart 4

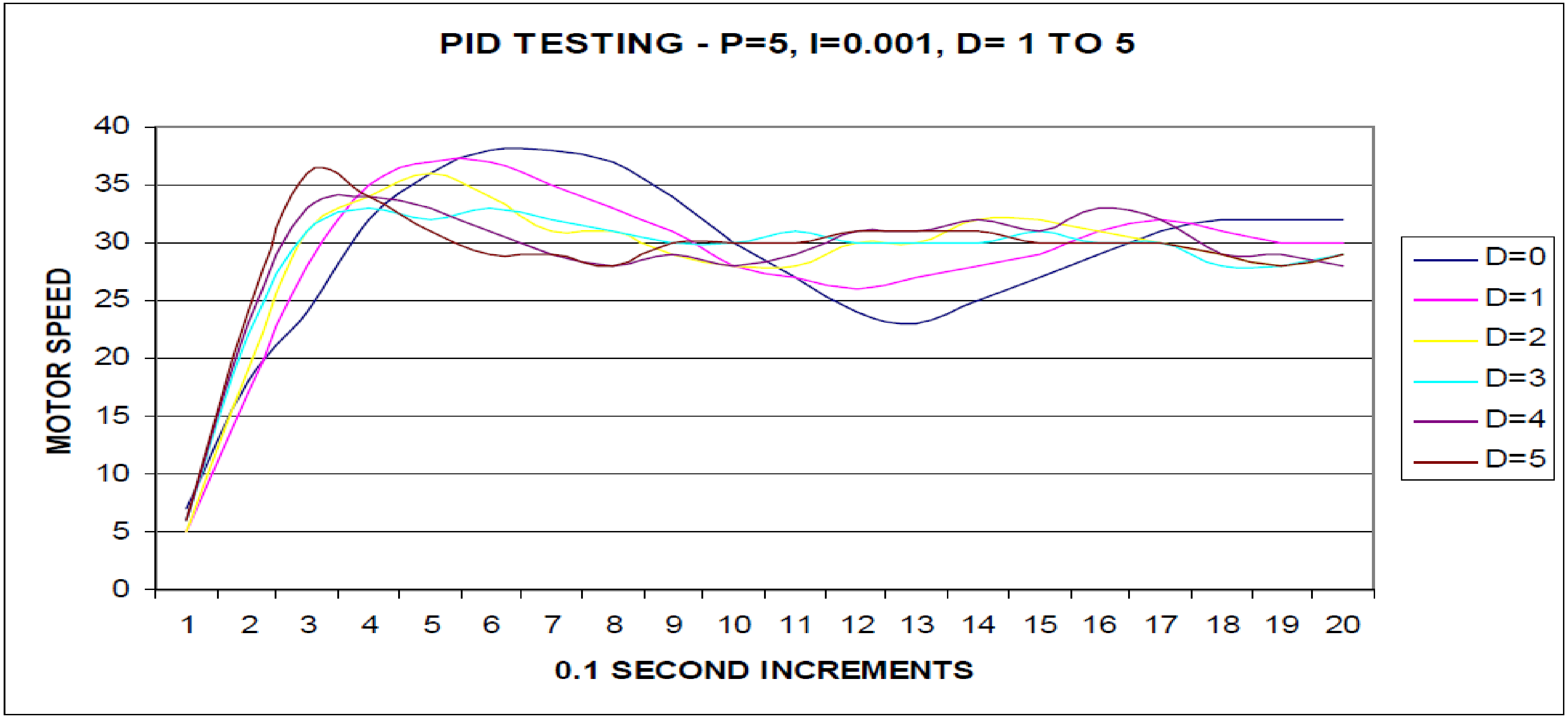


PID Tuning (Brute Force) chart 5



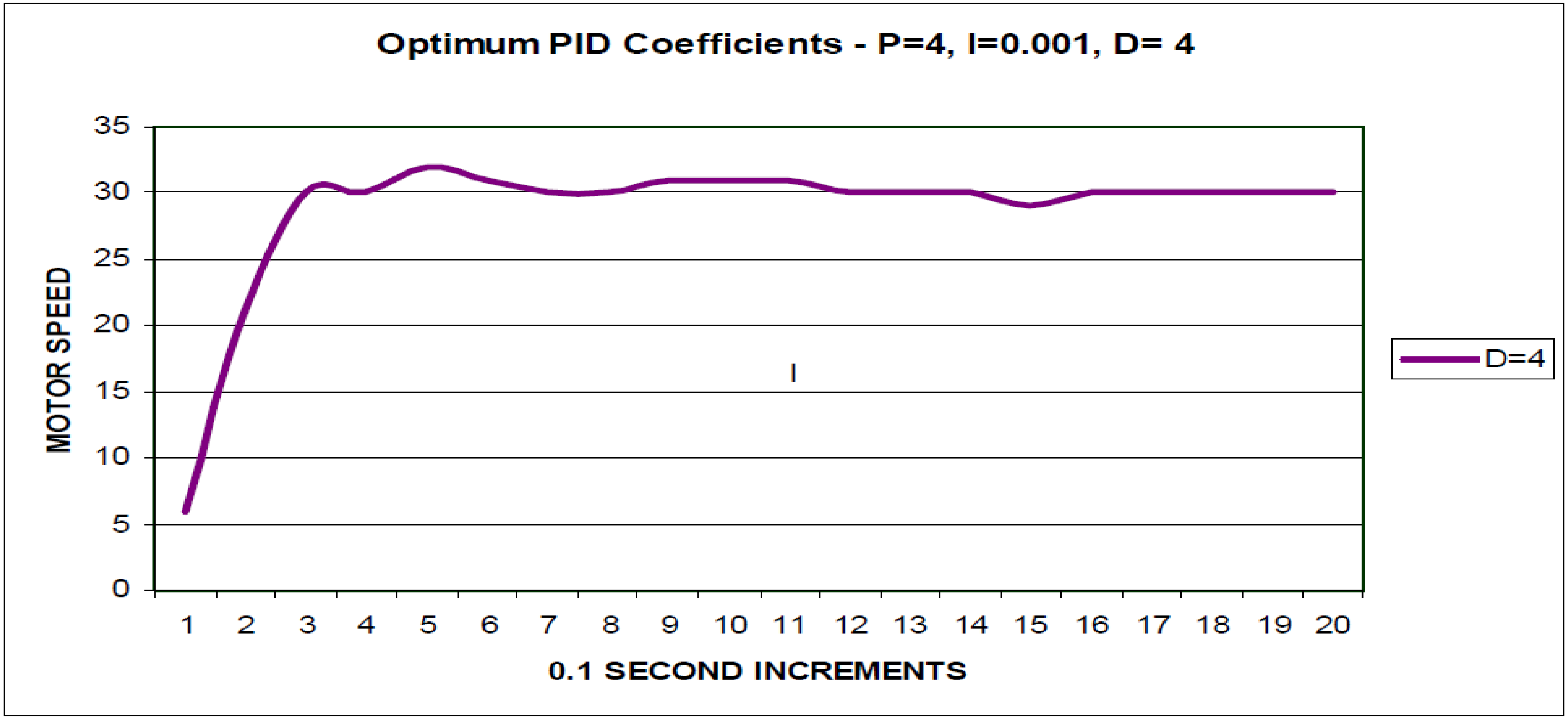
AARHUS UNIVERSITY
SCHOOL OF ENGINEERING

PID Tuning (Brute Force) chart 6



AARHUS UNIVERSITY
SCHOOL OF ENGINEERING

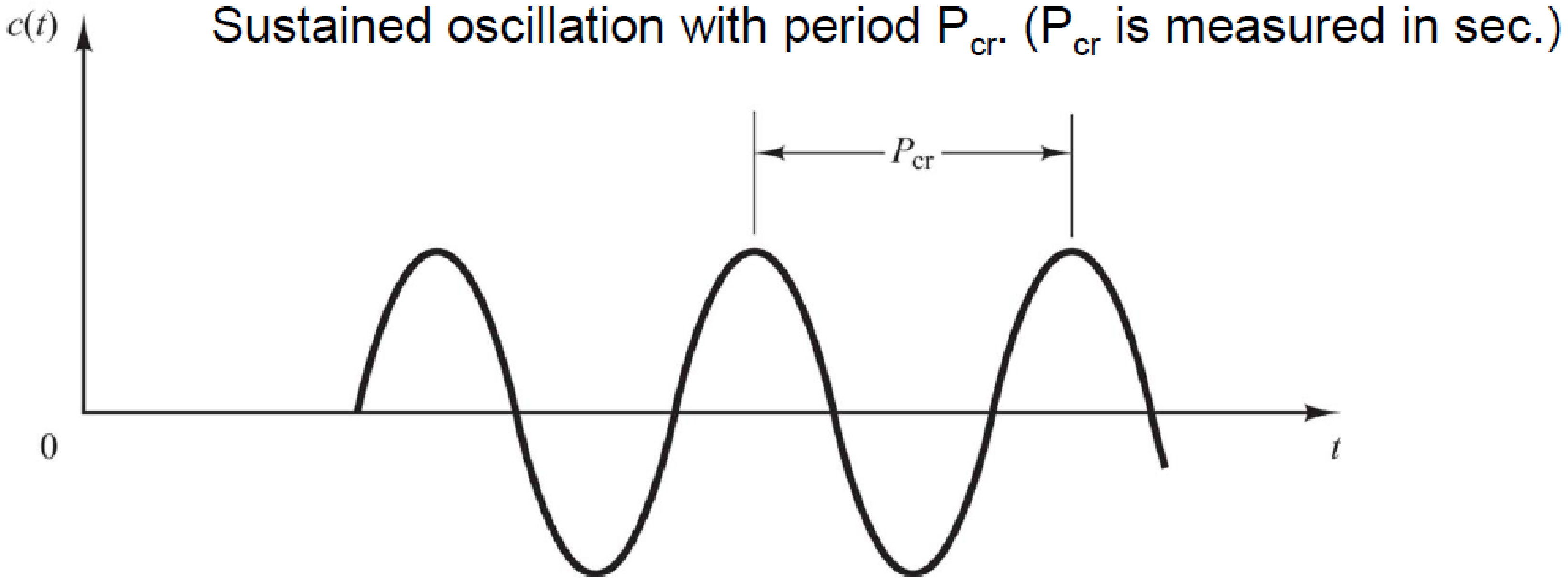
PID Tuning (Brute Force) optimum ☺



AARHUS UNIVERSITY
SCHOOL OF ENGINEERING

Ziegler – Nichols Tuning

- Begin with a low/zero value of gain Kp, set Ki and Kd to zero.
- Increase until a steady-state oscillation occurs, note this gain as Kcr

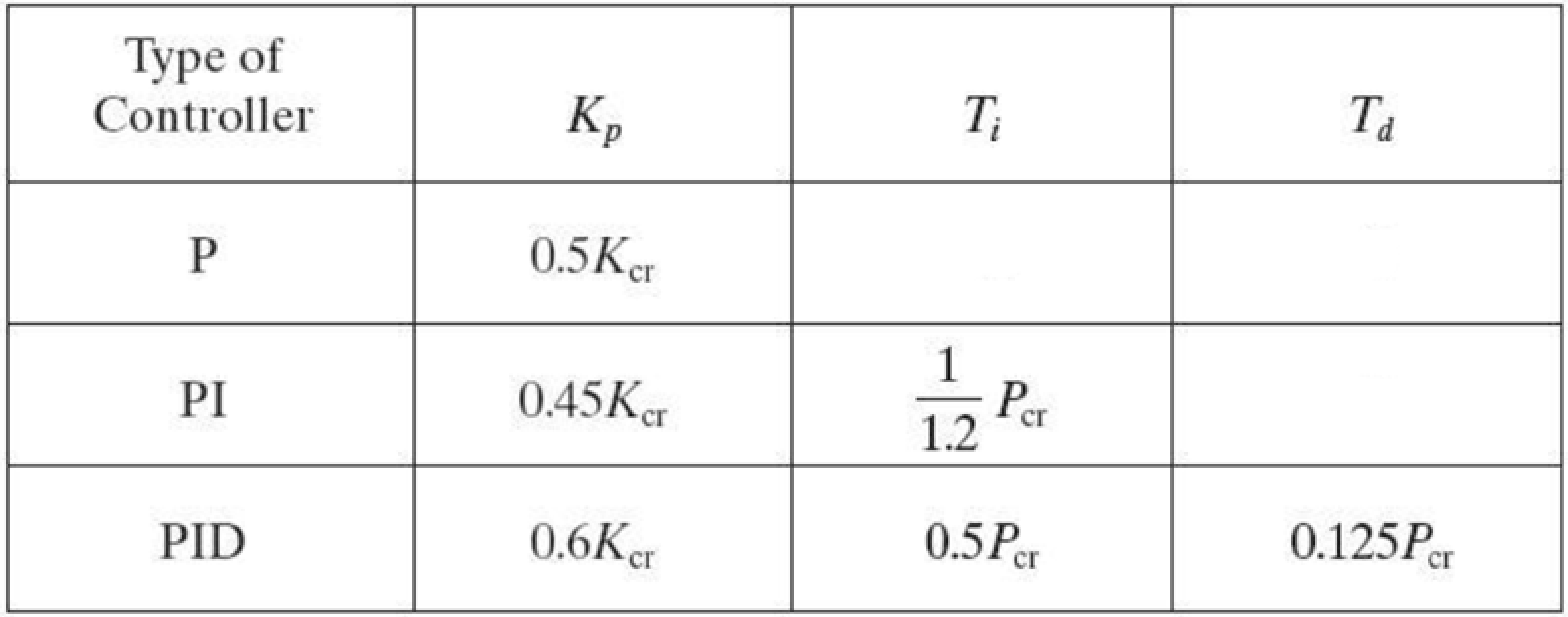


Sustained oscillation with period P_{cr}. (P_{cr} is measured in sec.)

Source: http://faculty.mercer.edu/jenkins_he/documents/TuningforPIDControllers.pdf

Ziegler – Nichols Tuning

Gain estimator chart



Type of Controller	K_p	T_i	T_d
P	$0.5K_{cr}$		
PI	$0.45K_{cr}$	$\frac{1}{1.2}P_{cr}$	
PID	$0.6K_{cr}$	$0.5P_{cr}$	$0.125P_{cr}$

Source: http://faculty.mercer.edu/jenkins_he/documents/TuningforPIDControllers.pdf

Ziegler – Nichols problems

Are you able to get sustained oscillation?

Is it safe to make the process oscillate?

What if the oscillation increases?

Lambda tuning

Lambda tuning gives non-oscillatory response with the response time (Lambda) required by the plant.

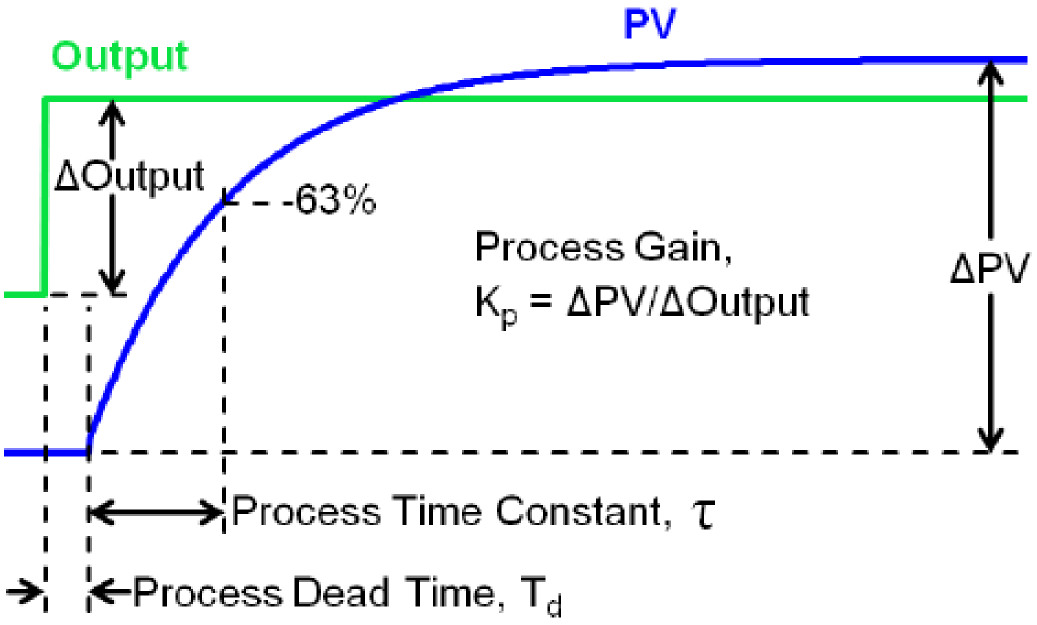
This is a good **starting point** if you want to do manual tuning.

The next slides shows how to tune a system with **1st order dynamics**, like the temperature regulator we will implement in the lab exercise.

Source: https://www.controlglobal.com/assets/13WPpdf/131022-coughran-controllers.pdf

Lambda tuning

a) **First-Order Process Dynamics:** if the process has the following self-regulating open loop response

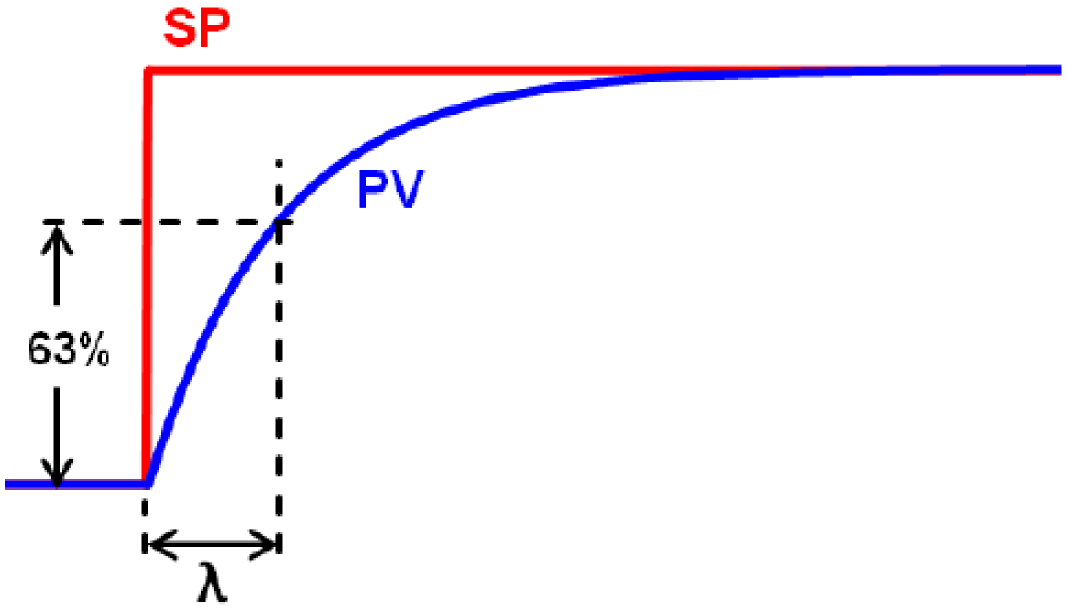


then we define Lambda (λ) as the closed loop time constant after a setpoint step. Time constant has the familiar meaning of the time to reach 63% of the final value.

Source: https://www.controlglobal.com/assets/13WPpdf/131022-coughran-controllers.pdf

Lambda tuning

For simplicity, we show here the closed loop response with negligible dead time:



To accomplish this in the PID controller, we set

Proportional Gain = $\dfrac{\tau}{K_P(\lambda+T_d)}$

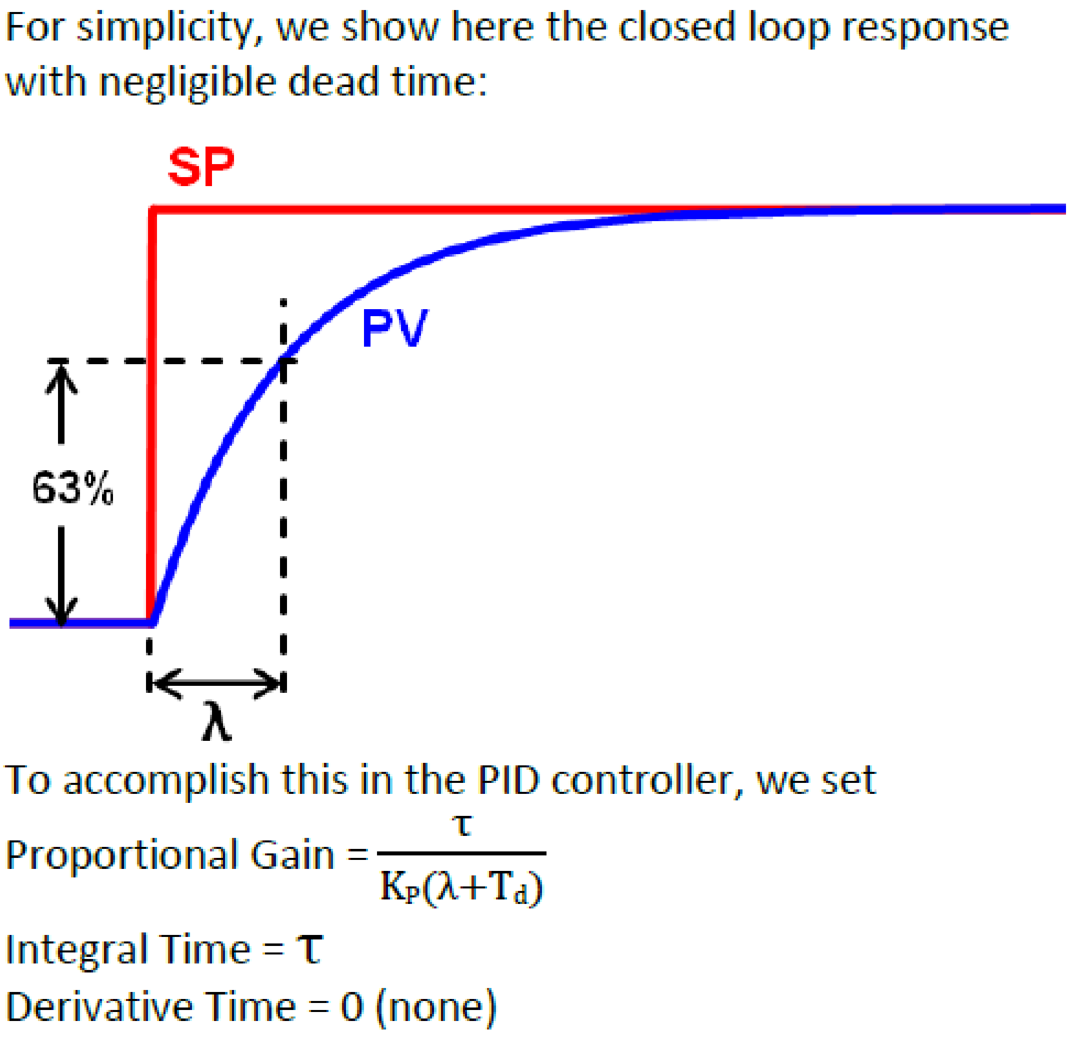
Integral Time = τ

Derivative Time = 0 (none)

The process response, in particular the dead time Td, limits how small you can make λ.

Source: https://www.controlglobal.com/assets/13WPpdf/131022-coughran-controllers.pdf

Lambda tuning



For simplicity, we show here the closed loop response with negligible dead time:

To accomplish this in the PID controller, we set

Proportional Gain = $\dfrac{\tau}{K_P(\lambda+T_d)}$

Integral Time = τ

Derivative Time = 0 (none)

For a PID controller with the same response time as the open-loop system, we choose:

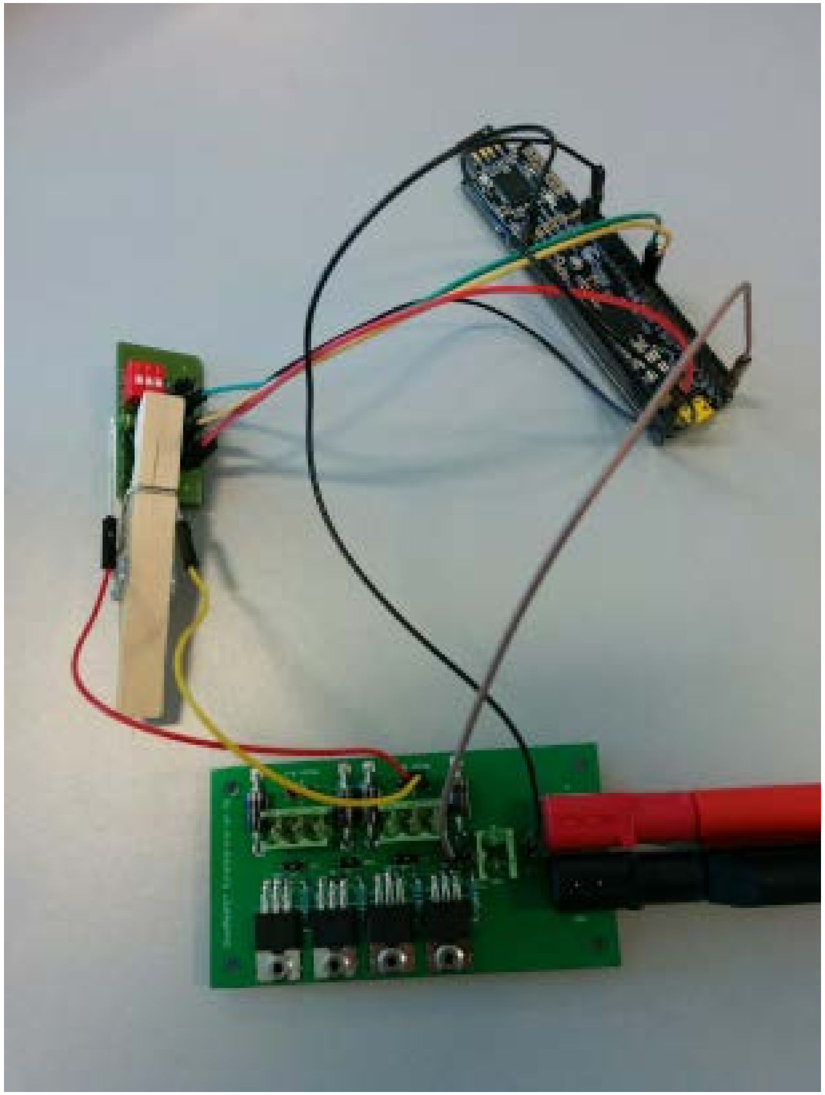
$K_{proportional} = 1/K_p$

$K_{integral} = 1/(\tau * K_p)$

$K_{derivative} = 0$

Source: https://www.controlglobal.com/assets/13WPpdf/131022-coughran-controllers.pdf

Lab – Temperature regulator



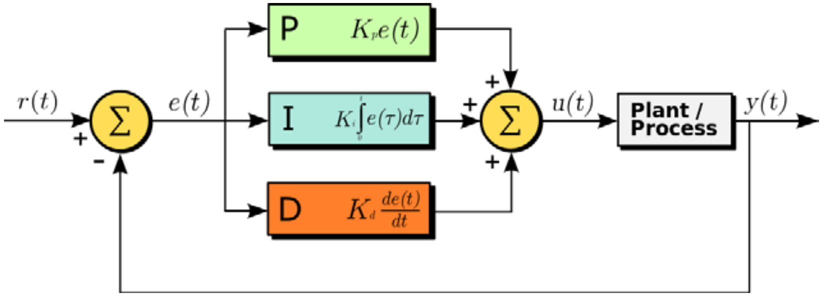
Starting point for the lab experiment:

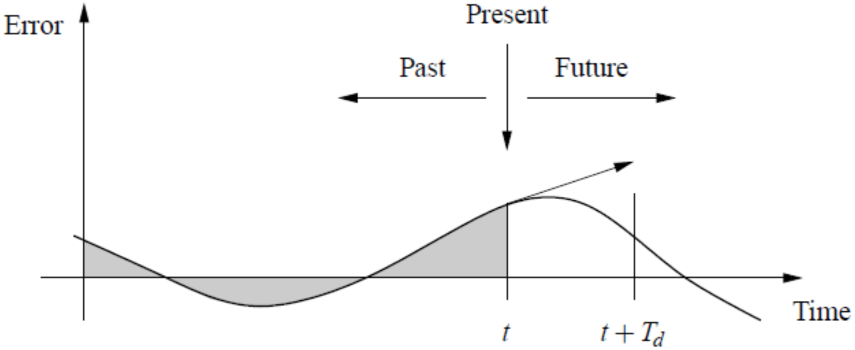
Kp = 2

Ki = 1/30

Kd = 0

Implementation (in discrete time)





- Measure the present error.
- P: Multiply the error with Kp
- I: Sum the present and all previous errors and multiply with Ki
- D: Substract the last previous error from the present error and multiply with Kd

Implementation, proportional

```
// calculate current error
currentError = setPoint - systemOutput;

// calculate proportional part
proportional = currentError;











correction = proportional * Kp
```

- Measure the present error.
- P: Multiply the error with Kp

Implementation, integral

```
// calculate current error
currentError = setPoint - systemOutput;




// calculate integral part
integral = integral + (currentError * dt);

// limit the integral
if (integral > integralMax) integral = integralMax;
if (integral < integralMin) integral = integralMin;




correction = integral * Ki
```

- Measure the present error.
- I: Sum the present and all previous errors and multiply with Ki

Implementation, derivative

```
// calculate current error
currentError = setPoint - systemOutput;




// calculate derivative part
derivative = (currentError - previousError) / dt;

correction = derivative * Kd;
previousError = currentError;
```

- Measure the present error.
- D: Substract the last previous error from the present error and multiply with Kd

Implementation, complete

```
// calculate current error
currentError = setPoint - systemOutput;

// calculate proportional part
proportional = currentError;

// calculate integral part
integral = integral + (currentError * dt);

// limit the integral
if (integral > integralMax) integral = integralMax;
if (integral < integralMin) integral = integralMin;

// calculate derivative part
derivative = (currentError - previousError) / dt;

correction = proportional * Kp + integral * Ki + derivative * Kd;
previousError = currentError;
```

- Remember to limit the output signal to what is physically possible.
- E.g. PWM = 110% is impossible.. output saturation

Sampling rate

- The sampling rate affects both the integral term and the derivative term.
- Increasing the sampling rate produces a more precise integral and a more responsive controller output.
- But increasing the sampling rate increases noise in the output, because the derivative part can change often, unless the input is filtered.

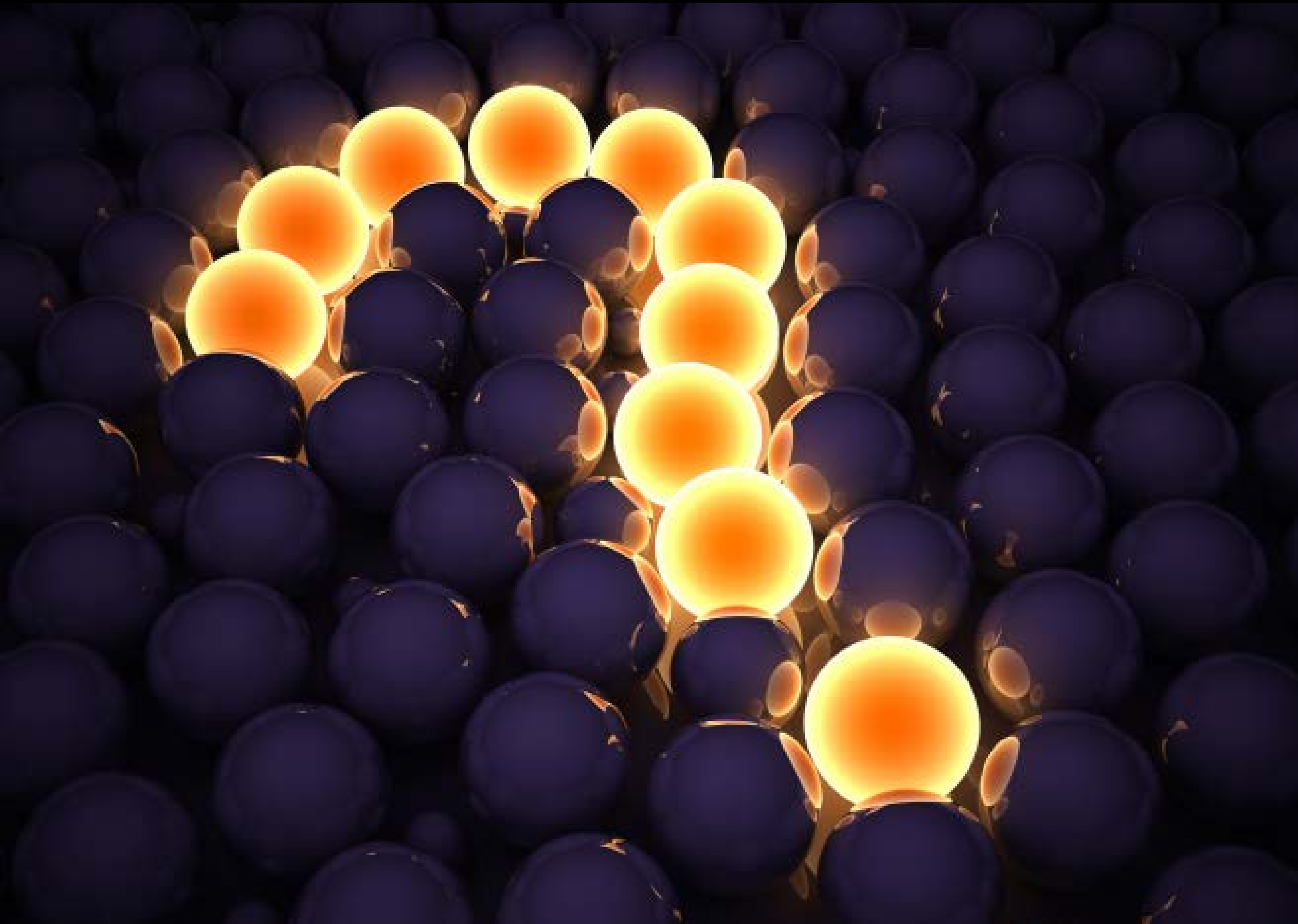


Image resources

- Outrun gif: https://media.giphy.com/media/IEqEhKiyw6cnK/200w_d.gif
- op-amp: https://en.wikipedia.org/wiki/Operational_amplifier#/media/File:Op-Amp_Inverting_Amplifier.svg
- PID controller block diagram: https://en.wikipedia.org/wiki/Control_theory#/media/File:PID_en.svg
- PID hand drawings: https://www.youtube.com/watch?v=UR0hOmjaHp0
- PID formulas and past, present, future: Feedback Systems An Introduction for Scientists and Engineers
- F16: https://allwallpapers.info/wp-content/uploads/2016/05/8948-general-dynamics-f-16-fighting-falcon-1920x1080-aircraft-wallpaper.jpeg
- Child being punished: http://arkiv.norskfolkemuseum.no/no/Forskning/Norsk-etnologisk-gransking/Alle-undersokelser/Oppdragelse-for-og-na-Sporreliste-nr-236/
- Windmill: https://wattsupwiththat.files.wordpress.com/2009/08/windmills_tx-ok-panhandle-1024.jpg

- Question mark: https://wall.alphacoders.com/big.php?i=437563