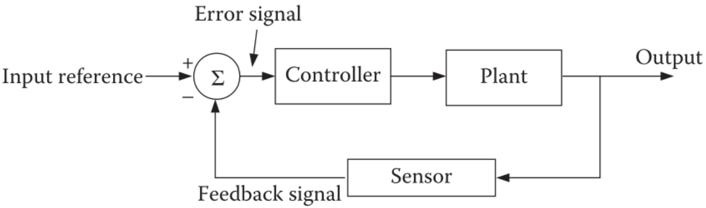
5ELEN018W - Robotic Principles
Lecture 7: Control - Part 2

Dr Dimitris C. Dracopoulos

Feedback Controllers



- Many types of controllers and control algorithms
- The simplest (and one of the most commonly used for certain systems in industrial process control) is *PID* control

PID Controllers

Proportional + **Integral** + **Derivative**

- Can have just the P component
- or both P and I (PI)
- or both P and D (PD)
- or all 3 of them (PID)

Proportional Control (P)

The controller applies an action which is proportional to the error $e(t)$:

$$action = K_p \cdot e(t) \tag{1}$$

where K_p is the gain of the proportional controller.

A Simple Example: A Robot following a Path

Consider a robot that we would like to keep following a given path at a distance of $d1$.

- The error at any time t is measured by $e(t) = d1 - d$ where d is the minimum current distance from the given path.
- if $e(t) > 0$: the action would be turn by $K_p \cdot e$ degrees away from the path.
- if $e(t) < 0$: the action would be turn by $K_p \cdot e$ degrees towards the path.

Integral Control (I)

Proportional control (P) might not be able to extinguish a steady-state error.

- The integral error calculates the accumulated error over time, i.e. it introduces some memory for the error.

$$integral\ error = K_i \cdot \int_0^t e(\tau)d\tau \tag{2}$$

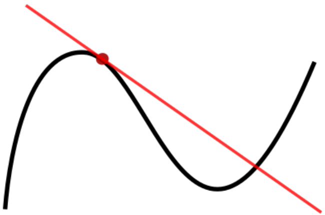
where K_i is the integral gain.

$\longrightarrow$ The integral error helps in reducing the steady-steady error, but it can lead to overshooting.

Derivative Control - What is a Derivative?

The derivative of a function measures the sensitivity of changes to the output (value) of the function, based on changes of the input (independent variable) of the function.

- The slope of the tangent line is equal to the derivative.



$\longrightarrow$ It is also used to show the direction we need to follow and the magnitude of the step we need to take, in order to reduce an error (machine learning, etc).

Derivative Control (D)

It takes into account how fast the error changes.

$$\textit{derivative error} = K_d \cdot \frac{de(t)}{dt} \tag{3}$$

- It can correct some of the problems introduced by the integral error, e.g. saturation (the real life physical mechanisms can never be linear - a motor has always upper bounds, how fast it can rotate, etc)

$\longrightarrow$ It reduces overshooting.

PID Control

Combining 2 or all 3 components (individual gains can be set to 0 to leave out one or two components from I, D).

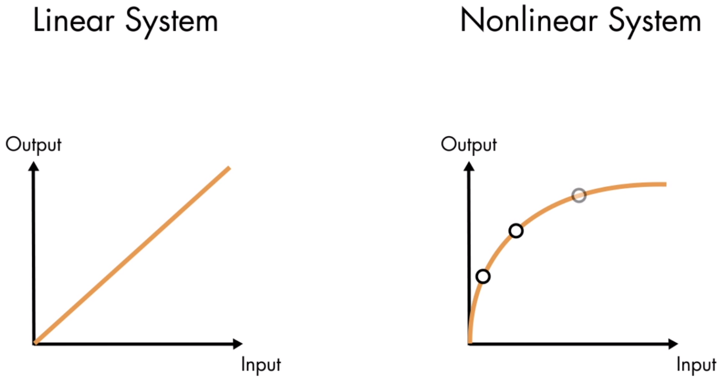
$$action = K_p \cdot e(t) + K_i \cdot \int_0^t e(\tau) d\tau + K_d \cdot \frac{de(t)}{dt} \tag{4}$$

PID Tuning

A difficult task is to find an appropriate set of the gain parameters K_p, K_i, K_d of the controller so as to perform satisfactorily (or optimise) with respect to:

- steady-state error
- transient response
- overshooting
- settling time

Linear vs Nonlinear Systems



- In real life all systems are nonlinear, however many of them can be linearised about their operation point.

Linear systems are easier to analyse and prove mathematically their behaviour and properties.

Discrete vs Continuous Dynamic Systems

- Discrete dynamic systems are described by difference equations.

$$x(n) = 5 * x(n-1) + 6 * x(n-2) + 2$$

- Continuous dynamic systems are described by differential equations.

$$\ddot{x} = 5 * \dot{x} + 10 * x + 10$$

The Laplace Transform

Linear differential equations describing physical dynamic systems (including robots) can be transformed to algebraic equations which can be more easily solved (and also analyse) using the Laplace transform.

- The robotic arm performing a surgery (mass spring damper example), seen last week:

$$m\ddot{x} + b\dot{x} + kx = f \tag{5}$$

assuming all initial conditions are set to 0, then its Laplace transform is:

$$ms^2 + bs + k = F \tag{6}$$

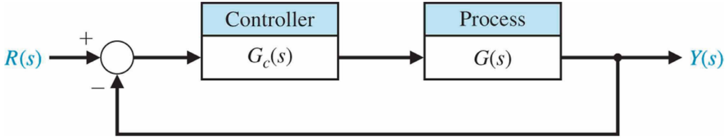
Transfer Functions

The transfer function of a *linear, time-invariant* system is defined as the ratio of the Laplace transform of the output variable $Y(s)$ to the Laplace transform of the input variable $R(s)$, with all initial conditions assumed to be 0:

$$G(s) = \frac{Y(s)}{R(s)} \tag{7}$$

- These are used to make easier the modelling and analysis of dynamic systems.

Feedback (Closed-Loop) Controllers Transfer Function



The transfer function of the closed-loop system is:

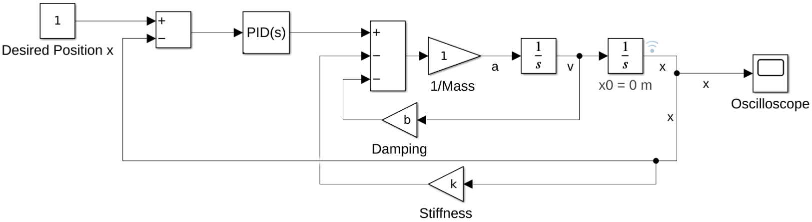
$$T(s) = \frac{G(s)G_c(s)}{1 + G(s)G_c(s)} \tag{8}$$

- A system is unstable where the closed loop transfer function diverges for s (e.g. where $G(s)G_c(s) = -1$).
- Stability is guaranteed when $G(s)G_c(s) < -1$.

PID Control for the Robot Arm Surgeon

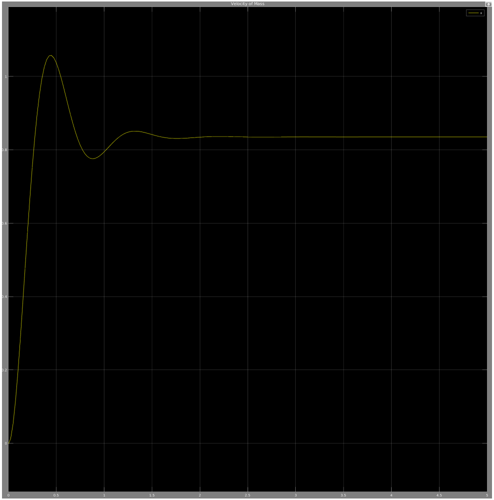
$$m\ddot{x} + b\dot{x} + kx = f \tag{9}$$

The desired position is 1, starting at position $x = 0$. For all the simulations, the following parameters were used:
$m = 1, b = 6, k = 9.86960.$



P-Controller

$K_p = 50$

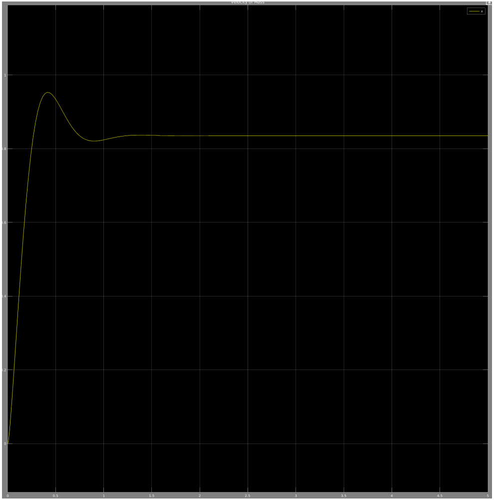


P-Controller (cont'd)

- large steady-steady error
- large overshoot
- large settling time

PD-Controller

$K_p = 50, K_d = 2.5$



PD-Controller (cont'd)

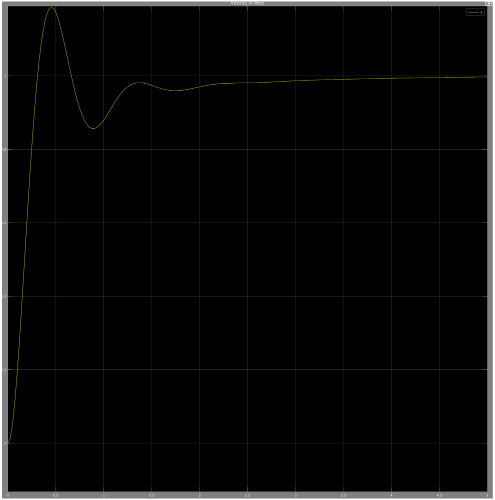
The *D*-component has reduced:

- the overshoot
- the settling time

Still large steady-state error.

PI-Controller

$K_p = 50, K_i = 40$

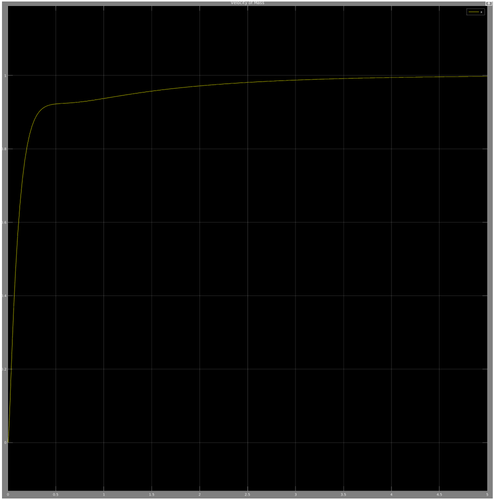


PI-Controller

- 0 steady-steady error
- still large overshoot

PID-Controller

$$K_p = 50, K_i = 40, K_d = 8$$



PID-Controller (cont'd)

- no steady steady error
- no overshoot
- faster rise time