Lecture 4:
Linearization.
Basis of PID controller.

Outline

- Example Control Systems
- Deviation variables
- Linearization
- PID controller

Examples

DARPA Grand Challenge:

- System = Robot Car.
- Controllable Inputs = Orientation wheels.
- Output = Position of car.
- Reference = Planned trajectory.
- Feedback = Deviation from planned trajectory.
- Disturbances system = Mechanical friction in car.
- Measurement error = Deviation measured location.
- Controller = Output feedback.

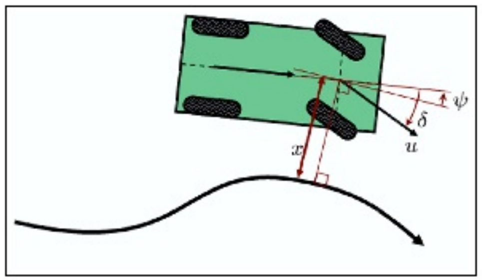


Figure 24: Illustration of the steering controller. With zero cross-track error, the basic implementation of the steering controller steers the front wheels parallel to the path. When cross-track error

Deviation variables

- The output response of a system is affected by the initial conditions and input signals.
- The effect of the initial conditions can be eliminated by assuming that initial conditions are at steady state.
- If the initial conditions are at steady state, it implies that the initial values of the time derivatives are zero, but not the initial value of the signals (inputs, outputs, states).
- Deviation variables (X(t)) are defined as

$$X(t) = x(t) - x_{ss}$$

where $x(t)$ is the total value of the signal and x_{ss} is the steady state value of the signal

- The initial condition of a deviation variable is always zero.

Linearization

- Linear models can be seen as approximations of nonlinear systems.
- Linear approximation to the nonlinear equations is valid for a region near some point around which the linearization is made.
- The point (x_0,u_0) is an equilibrium for

$$\dot{x}(t) = f\left(x(t), u(t)\right) \qquad \text{if} \quad f\left(x_0, u_0\right) = 0.$$

- The nonlinear system is described in the vicinity of the equilibrium point (x_0,u_0) by

$$\dot{z}(t) = Az(t) + Bv(t)$$ where $z=x-x_0$ and $v=u-u_0$.

- The matrices A and B have elements a_{ij} and b_{ij} given by

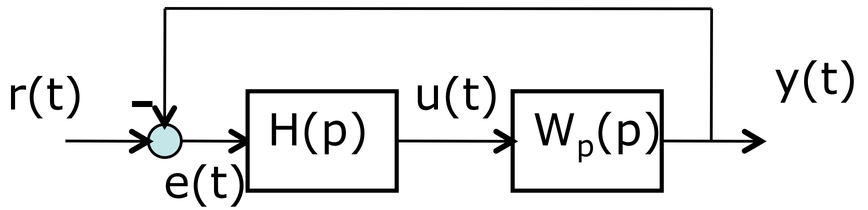
$$a_{ij} = \left.\frac{\partial f_i}{\partial x_j}\right|_{(x,u)=(x_0,u_0)} \qquad b_{ij} = \left.\frac{\partial f_i}{\partial u_j}\right|_{(x,u)=(x_0,u_0)}$$

PID controller

Franklin et al., Ch. 4

PID controller – Continuous time domain

- Feedback controllers make corrective actions based on the difference between the desired and the actual values of output signals.



Control error:
e(t)=r(t)-y(t)

- In time domain

$$u(t) = u_{ss} + k_P e(t) + k_I \int_0^t e(\tau)d\tau + k_D \frac{de(t)}{dt}$$

$$U(t) = k_P e(t) + k_I \int_0^t e(\tau)d\tau + k_D \frac{de(t)}{dt}$$

Control parameters:
k_P, k_I, k_D

- Proportional term – related to the present value of *e(t)*
- Integral term – related to the history of *e(t)*
- Derivative term – related to the velocity (signed) of *e(t)*

PID controller – Transfer function

- The PID output can written as

$$U(t) = k_P \left(e(t) + \frac{1}{T_I} \int_0^t e(\tau) d\tau + T_D \frac{de(t)}{dt} \right)$$

Control parameters: k_P, T_I, T_D

- The transfer function of the PID controller is

$$H(s) = k_P + \frac{k_I}{s} + k_D s$$

$$U(s) = H(s)E(s)$$
$$E(s) = R(s) - Y(s)$$

- Integral term eliminates the static error $\lim_{t \to \infty} e(t)$
- For noisy signals, the use of the derivative action is undesirable (noise effects are amplified)

Discrete PID

- The continuous PID is modified in order to execute the PID algorithm periodically using the sampled values of the controlled variables to determine the values of the controller output.
- The differentiation operator can be replaced by (using the Euler's classic approximation) ($p \longrightarrow \frac{(q-1)}{T}$)

$$\dot{x}(t) \approx \frac{x(t+T) - x(T)}{T}$$

where T is the sampling time

- After some manipulations, the discrete controller output can be rewritten as

$$u(t+T) = u_{ss} + k_P \left[e(t) + \frac{1}{T_I} \sum_{k=0}^{t+T} e(k) + \frac{T_D}{T} \left(e(t) - e(t-T) \right) \right]$$

PID controller - Tuning

- PID controllers are implemented in discrete time but tuned using a continuous formulation
- PID tuning involves the selection of the best values of k_P, k_I and k_D (or T_P, T_I and T_D). It depends on the process.
- There are many methods for tuning a PID:
  - Based on experiments with the process: Most of the tuning methods for PID controllers are based on experiments with the process plant (step response, parameters of oscillations, etc.) e.g. Ziegler Nichols.
  - Based on a mathematical model of the process: There are also formal methods for the design of PID controllers that demand a mathematical model of the plant.

PID controller

- The PID controller is the most commonly used controller.
- The derivative term is usually implemented with a filter. Only those controllers have state space representation.

$$H(s) = k_P + \frac{k_I}{s} + \frac{k_D s}{1 + \varepsilon k_D s} \qquad \text{for a small } \varepsilon$$

- PID controllers have typically bad performance when there is a significant time delay in the loop. They work though nicely for relatively small time delays.
- PID algorithms can be different depending on manufactures. It is important to know the configuration of the PID algorithm before tuning.

Conclusions

- Linearization
- Output/Error feedback
- PID laws