Lecture notes: Intro to control and a motivating example

Jie Fu

Department of Electrical and Computer Engineering
Robotics Engineering Program
Worcester Polytechnic Institute

RBE502, 2017

Table of Contents

The lecture is based on and adapted from

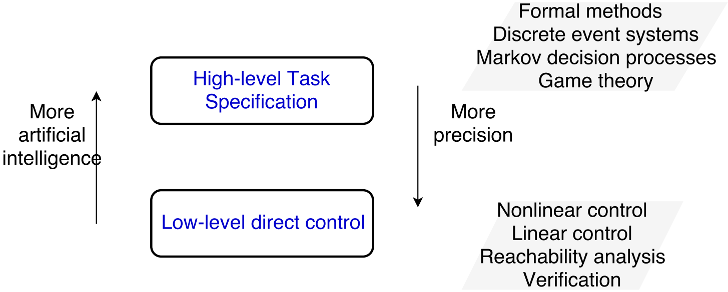
- Robotics 2, taught by Prof. Alessandro De Luca. Dipartimento di Ingegneria informatica, automatica e gestionale Antonio Ruberti (DIAG), Sapienza Universita di Roma.

The goal of control

Control has different meanings under different contexts:

- successfully complete a task or work program.
- accurate execution of a motion trajectory.
- minimizing a positioning error under disturbances.

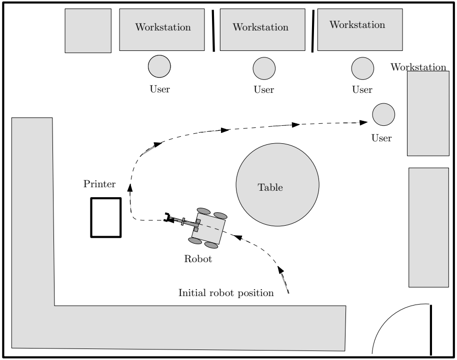
A hierarchical view



Example

Mobile manipulator RB-1 — Robotnik

How to accomplish the task of picking a printout?

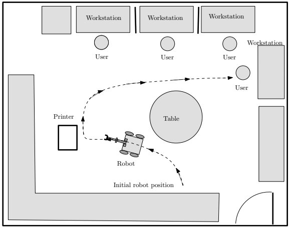


Example

design motion primitives: Navigate, pick, place

high-level:

a seq of motion: Go to the printer
pick up the paper
Go to the user
place the paper
(Handover the paper to user)



low-level:

Go to: avoid obs.
goal s.t.: can reach the paper.
Time optimal.
Energy

pick

Evaluation of control performance

quality of execution in **nominal** conditions

- task completion time. ✓ optimal control.
- energy consumption. ✓
- accuracy/repeatability (in static and dynamic terms).

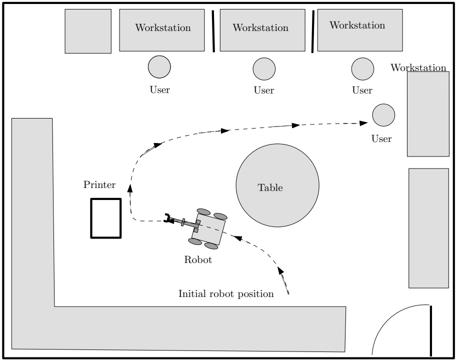
robustness and **safety** in **perturbed** conditions

- repeatability despite disturbances and modeling errors.
- adapting to the changing environment, or system parameters.
- safety guarantee against external disturbances.

- Feedback on task completion. → stochastic control.
  planning in MDP
- under-actuated system. [Differantial flat]

Example: Control performance

if you were given the task of control design, what would you like to achieve in the performance?

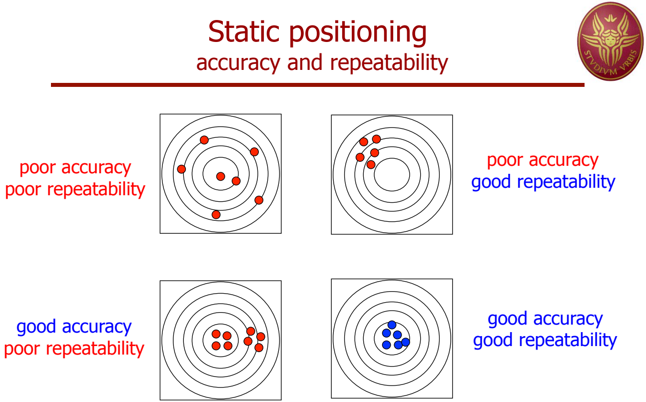


Hybrid system.

Discrete

Continuous.

for industrial robot, accuracy and repeatability have been the key performance criteria.



repeatability:

`https://youtu.be/UR6YkdAP8Jk`

Control schemes

feedback control:

- insensitivity to mild disturbances and small variations of parameters.
- feeback control+ feedforward compensation: Tracking trajectory and insensitivity to mild disturbances.

robust control

- tolerates relatively large uncertainties of known range

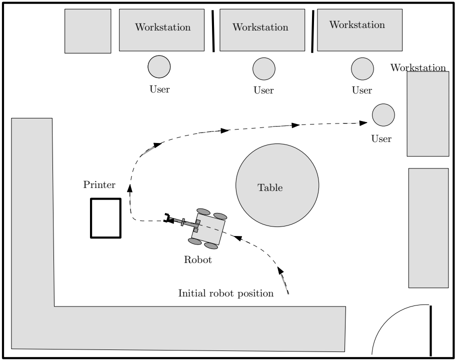
adaptive control

- improves performance on line, adapting the control law to a priori unknown range of uncertainties and/or large (but not too fast) parameter variation.

intelligent control

- Learning-based control, and reinforcement learning.
- self-organizing behavior in swarm robotics.

Q: Can you picture some scenarios for these control scheme to apply in the pick print out task?



feedback control: Nav.

robust control pick up.

adaptive control X

intelligent control

Functional structure of a control unit

task program

↓

trajectory planning

↓

direct control algorithms

↓

actuators

↓

robot

↓

environment

modeling of tasks

geometric and kinematic models
coordinate transformations

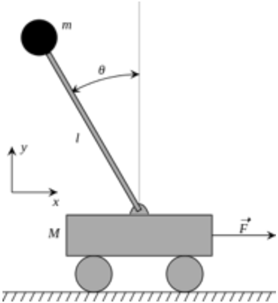
nonlinear methods
dynamic control

(electrical and mechanical)
dynamic models

structured and unstructured
world modeling

Table of Contents

Dynamic modeling of a pendulum



Dynamic modeling of a pendulum

Kinetic Energy

$$K = \frac{1}{2} M v_1^T v_1 + \frac{1}{2} m v_2^T v_2$$

under the assumption of mass-less link.

Potential energy

$$P = mg\ell \cos(\theta)$$

The Lagrangian:

$$\begin{aligned} L &= K - P & (1) \\ &= \frac{1}{2} M v_1^T v_1 + \frac{1}{2} m v_2^T v_2 - mg\ell \cos(\theta) & (2) \\ &= \frac{1}{2}(M+m)\dot{x}^2 - m\ell\dot{x}\dot{\theta}\cos(\theta) + \frac{1}{2} m\ell^2\dot{\theta}^2 - mg\ell\cos(\theta) & (3) \end{aligned}$$

Dynamic modeling of a pendulum

Using Eular-Lagrangian mdodeling method:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = \tau.$$

which gives

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{x}} - \frac{\partial L}{\partial x} = F \tag{4}$$

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}} - \frac{\partial L}{\partial \theta} = 0 \tag{5}$$

$$(M+m)\ddot{x} - m\ell\ddot{\theta}\cos(\theta) + m\ell\dot{\theta}^2\sin(\theta) = F \tag{6}$$

$$\ell\ddot{\theta} - g\sin(\theta) - \ddot{x}\cos(\theta) = 0 \tag{7}$$

State space form

The state space representation of a system replaces n-th order differential equations with **first order** differential equations

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$$

where x is the **state vector**. u: input vector
Question: How to select the state vector?

- NOT UNIQUE.
- One choice: Let n be the number of the highest derivative in the generalized coordinates. Each state $x(i)$ in the state vector x represents a variable or the k-th order derivative of that variable for some $0 \leq k \leq n-1$

State space form of linear system

Example:

$$\dddot{y} + a_1\ddot{y} + a_2\dot{y} + a_3 y = b_0 u$$

Recall: The state space form is $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$.

n th : $n = 3$

$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \end{bmatrix}$

$k \le n-1 = 3-1 = 2$

$x_1 = y$
$x_2 = \dot{y}$
$x_3 = \ddot{y}$

$\dot{x} = f(x, u) = \begin{bmatrix} f_1(x, u) \\ f_2(x, u) \\ f_3(x, u) \end{bmatrix}$

$\boxed{\dot{X} = AX + Bu}$

$\dot{x}_1 = f_1(x, u) = x_2$

$\dot{x}_2 = f_2(x, u) = \ddot{y} = x_3$

$\dot{x}_3 = f_3(x, u) = \dddot{y} = b_0 u - a_1\ddot{y} - a_2\dot{y} - a_3 y$

$= b_0 u - a_1 x_3 - a_2 x_2 - a_3 x_1$

$A \in R^{n\times n}$

$B \in R^{1\times n}$

in general

$u \in R^m$

$B \in R^{m\times n}$

$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_3 & -a_2 & -a_1 \end{bmatrix}$

$B = \begin{bmatrix} 0 \\ 0 \\ b_0 \end{bmatrix}$

State space form of a nonlinear system

Example:

$$\dddot{y} + a_1 y \ddot{y} + a_2 \dot{y} + a_3 y^3 = b_0 u$$

Recall: The state space form is $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$.

$x_1 = y \quad x_2 = \dot{y} \quad x_3 = \ddot{y}$

$\dot{x}_1 = x_2 \quad \dot{x}_2 = x_3 \quad \dot{x}_3 = b_0 u - a_1 y \ddot{y} - a_2 \dot{y} - a_3 y^3$

$= b_0 u - a_1 x_1 x_3 - a_2 x_2 - a_3 x_1^3$

$\dot{x} = \underbrace{f(x)}_{} + g(x)\, \underline{\underline{u}}$

control-affine system

$$f(x) = \begin{bmatrix} x_2 \\ x_3 \\ -a_1 x_1 x_3 - a_2 x_2 - a_3 x_1^3 \end{bmatrix} \qquad g(x) = \begin{bmatrix} 0 \\ 0 \\ b_0 \end{bmatrix}$$

State space form of the inverted pendulum

$F(\ddot{\theta})$

$$(M+m)\ddot{x} - m\ell\ddot{\theta}\cos(\theta) + m\ell\dot{\theta}^2\sin(\theta) = F \quad (8)$$

$$\ell\ddot{\theta} - g\sin(\theta) - \ddot{x}\cos(\theta) = 0 \quad (9)$$

- The highest order $n =$ 2
- Variables and their k-th order derivatives: $k \leq n-1 =$ 1

x, θ

$z_1 = x$ $\quad$ $\dot{z}_1 = z_2$

$z_2 = \dot{x}$ $\quad \Rightarrow \quad$ $\dot{z}_2 = ? = \ddot{x} = ? = f_3(z, u)$

$z_3 = \theta$ $\quad$ $\dot{z}_3 = z_4$

$z_4 = \dot{\theta}$ $\quad$ $\dot{z}_4 = \ddot{\theta} = ?$ $\quad f_4(z, u)$

$u = F$

equilibrium: $F = 0$

$\dot{z} = f(z)$ $\quad f(z_e) = 0$ $\quad z_e = \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix}$

$\dot{x} = 0, \quad \dot{\theta} = 0$

Using matlab: symbolic computation

```
% inverted pendulum control: Example

syms x theta F x_dot theta_dot x_ddot theta_ddot g
syms M m l
X  = sym ('X', [4,1]); % create a 4 by 1 vector for state
X(1)=x;
X(2)= x_dot;
X(3) = theta;
X(4) = theta_dot;

eq1 = (M+m)*x_ddot - m*l*theta_ddot*cos(theta) + m*l*theta_dot^2 *sin(theta)-F;
eq2 = l*theta_ddot - g*sin(theta)  - x_ddot *cos(theta);

sol = solve([eq1==0, eq2==0], [x_ddot, theta_ddot]);

% display the solution using the following commands:
% sol.x_ddot
%
% ans =
```

An informal introduction of stability

Equilibrium is a state of a system which does not change.

$\dot{x} = f(x)$ $\qquad\qquad$ $f(x_e) = 0$

Stable equilibrium: An equilibrium is considered **stable** (for simplicity we will consider asymptotic stability only) if the system **always returns to it after small disturbances.** If the system moves away from the equilibrium after small disturbances, then the equilibrium is **unstable.**

Q: What are the equilibria of the cart-pole system? Without input force F, can you identify any stable equilibrium?

From nonlinear to linear system

Linear control theory: Control design for stabilizing and trajectory tracking in linear dynamical systems.

Many nonlinear systems can be **linearized**.

- **Simplification of dynamics**: Cart-pole system
- Feedback linearization: Rigid-body dynamics, Dubins car system.
- Jacobian linearization: General nonlinear system.

Small-angle assumption

Assumption: $\theta \to 0$, $\dot{\theta} \to 0$, $\sin(\theta) \to \theta$, and $\cos(\theta) \to 1$.

$\dot{\theta}, \dot{x} \to 0$

$x ? = \text{const.}$

$$\dot{z}(2) = \ddot{x} = \frac{-\ell m \sin(\theta)\dot{\theta}^2 + F + gm\cos(\theta)\sin(\theta)}{M + m - m\cos(\theta)^2}$$

$$\dot{z}(4) = \ddot{\theta} = \frac{-\ell m \cos(\theta)\sin(\theta)\dot{\theta}^2 + F\cos(\theta) + gm\sin(\theta) + Mg\sin(\theta)}{\ell(M + m - m\cos(\theta)^2)}$$

equilibrium: $\sin\theta\cos\theta = 0$, $gm\sin\theta + Mg\sin\theta = 0$ $\Rightarrow$ $\theta = 0$, x.

Control goal: keep the pole at $\theta = 0$, $x = \text{const}$. $\dot{\theta}, \dot{x} = 0$.

$\sin\theta \to \theta$. $\cos\theta \to 1$:

$$\dot{z}(2) = \frac{gm\,\theta}{M} + \frac{F}{M} \quad (\theta = z(3))$$

$$\dot{z}(4) = \frac{(M+m)g\,\theta}{m\ell} + \frac{F}{M\ell} \quad (\theta = z(3))$$

$$z \to 0: \quad \begin{pmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{pmatrix} \begin{matrix} \to \text{const} \\ \to 0 \\ \to 0 \\ \to 0 \end{matrix}$$

Feedback controller for linear system

The system after simplification is

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

Assume (A, B) controlable, then the system can be stabilized to the origin with a controller

$$u = -Kx$$

such that $A - BK$ has eigenvalues on the left-hand side of the complex plane.

ODE Solver

We need to validate our control design — by numerically solve ordinary differential equation (ODE):

$$\frac{dx}{dt} = f(t, x, u)$$

Under a given control input $u : [0 : T] \rightarrow R^m$, if the system is Lipchitz continuous, the solution of the ode is unique $x : [0, T] \rightarrow R^n$.

δt :

$$\frac{dx}{dt} = \frac{X(t+\delta t) - X(t)}{\delta t} = f(x(t), t, u(t))$$

$$X(t+\delta t) = f(x(t), t, u(t)) \cdot \delta t + x(t)$$

ode45

ode15

ODE Solver

```
[t,y]=ode45(myode,[0,T],[y0(1);y0(2)])
```

Inputs:

- ODE function name (or anonymous function). This function takes inputs (t, y), and returns dy/dt
- Time interval $[0, T]$: 2-element vector specifying initial and final time
- Initial conditions $[y0(1); y0(2)]$: column vector with an initial condition for each ODE.

Outputs:

- t contains the time points
- y contains the corresponding values of the integrated variables.

Demonstration

```
function dz = ode_pendulum(t,z,K)
m = 5; M = 100; g = 9.8; l = 0.5;
dz =zeros(4,1);
z=num2cell(z);
[x, x_dot, theta, theta_dot] = deal(z{:});
if abs(theta)>2*pi %
theta  = mod(theta, 2*pi); % wrap the angle to [0, 2pi]
end
F = - K*[x, x_dot, theta, theta_dot]' ;
dz(1) = x_dot;
dz(2) = (- l*m*sin(theta)*theta_dot^2 + F + g*m*cos(theta)*sin(
    theta))/(M + m - m*cos(theta)^2);
dz(3) = theta_dot;
dz(4) = (- l*m*cos(theta)*sin(theta)*theta_dot^2 + F*cos(theta)
     + g*m*sin(theta) + M*g*sin(theta))/(l*(M + m - m*cos(theta
    )^2));

end
```