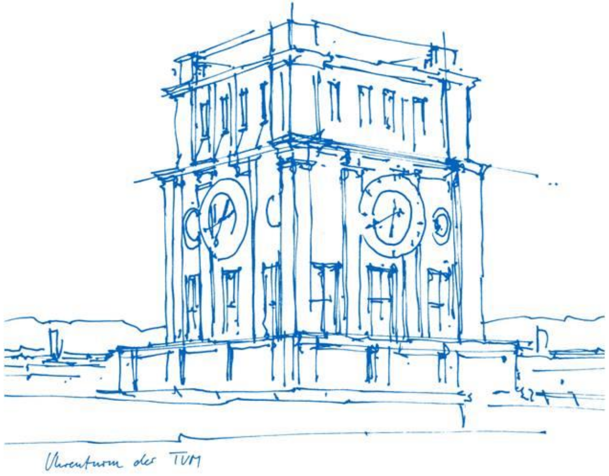
Lehrstuhl für Produktentwicklung und Leichtbau
TUM School of Engineering and Design
Technische Universität München

Numerical optimization for robot design and control – Day 03 Control synthesis

Akhil Sathuluri

Prof. Dr. Markus Zimmermann



Garching, 05th of September 2022

Day-3: Control synthesis: Overview

Part-1: Robot control: Theory and introduction

1. Introduction
2. Basic control system

Part-2: Robot control as an optimization problem

1. Problem formulation
2. Problem solving methods
3. Discussion

Part-1: Robot control: Theory and introduction

Introduction

- We have designed a robot, i.e., the values mechanical design variables that optimizes an objective of our choice

2R-planar robot

5-bar planar robot

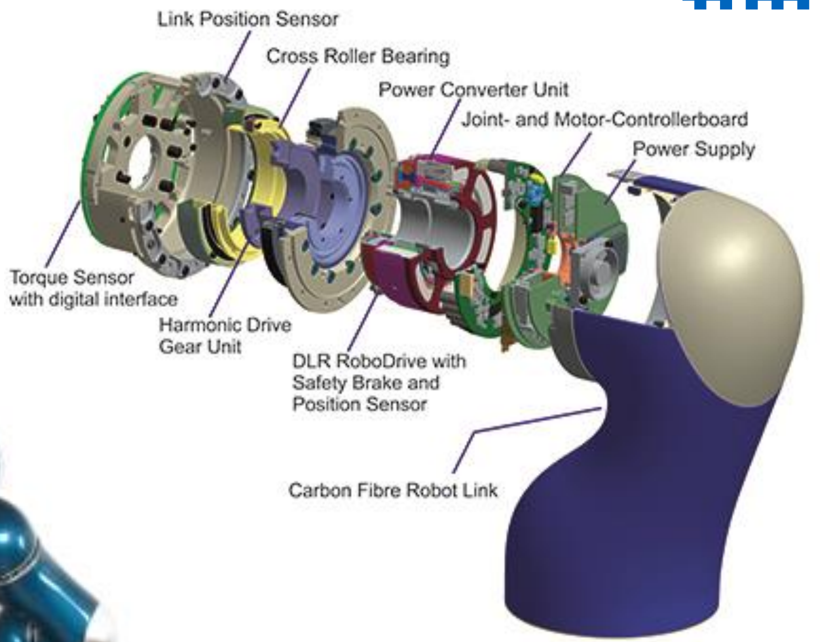
Musculo-skeletal robot

- Link lengths
- Link geometry
- Mechanical properties (mass, density)
- Cable positions …

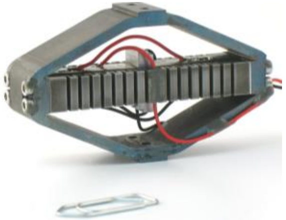
??

Introduction

- How do we make this robot do what we want? How do we move this robot?
- What are the components of a typical robot drive train look like?
- Lets have a look at a robot and its joint from DLR



Introduction

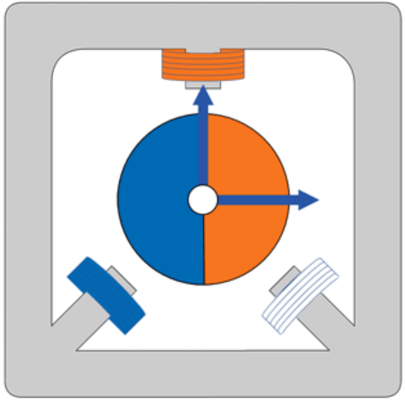


- What are some actuators one could use?

- Modes of actuation of robots:
  - Electrical motor
    - Stepper motors
    - Brushed DC motor
    - Blushless DC motor
    - Induction motor
  - Linear
  - Hydraulic
  - Pneumatic
  - Combinations of the above
  - Piezo electric …

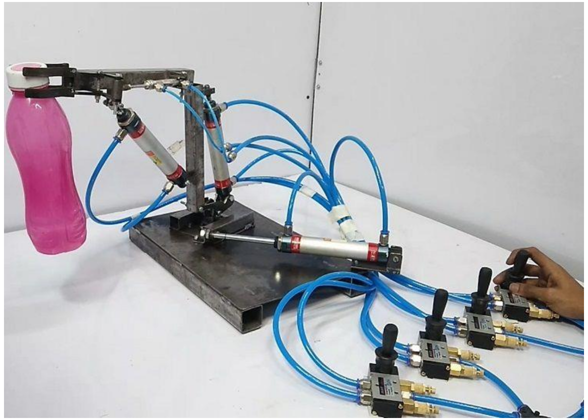
Introduction

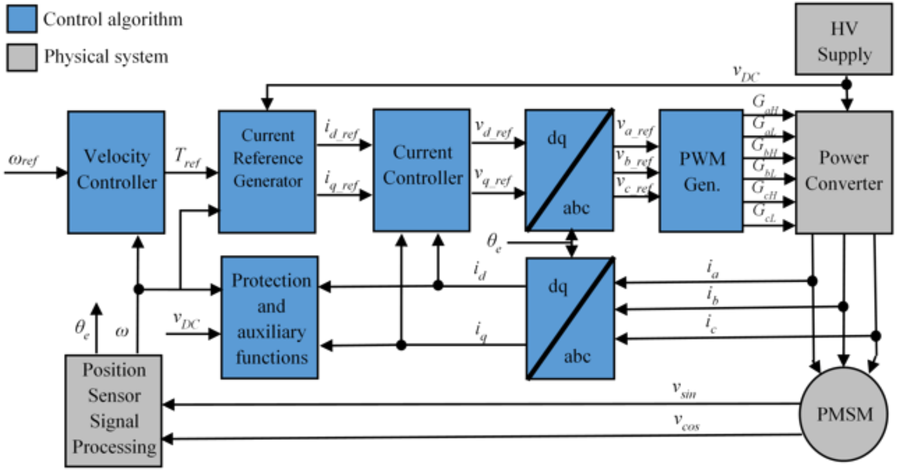
- How does the control strategy change for these different actuators? What do we really have control over?



What is robot control then?

- Modes of actuation of robots:
  - Electrical motor
  - Hydraulic



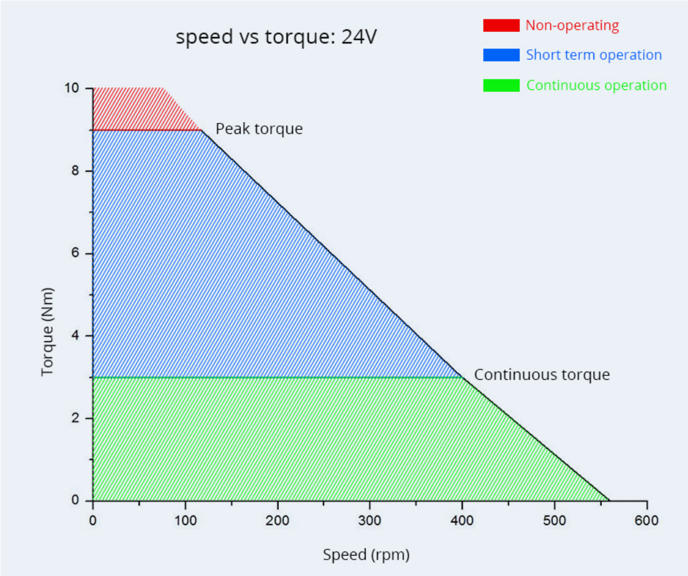


Introduction

- We generally use BLDCs and especially PMSMs for building our robots
  - Compact
  - Can produce high torque
  - High energy density

- More elements of the robot joint like gearing, bearings, encoders etc!

- General characteristics of a motor
  - What are the ideal characteristics that we would like to have?
  - How do we capture the behavior of all of these elements?



Introduction: Motor models

Input (in time domain):

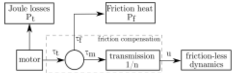
u_j(t) (demanded joint torque)

ω_j(t) (current joint velocity)

Description:

ω_j = ω_m/N, tau_j = tau_m*N

1. ω_j >=0 and u_j >=0, tau_j = m
2. ω_j >0 and u_j <0, tau_j = max
3. ω_j <0 and u_j >0, tau_j = min(tau_max, u_j)
4. ω_j <0 and u_j<0, tau_j = max(kv*(abs(ω_j) - ω_max), u_j)



Power dissipation:

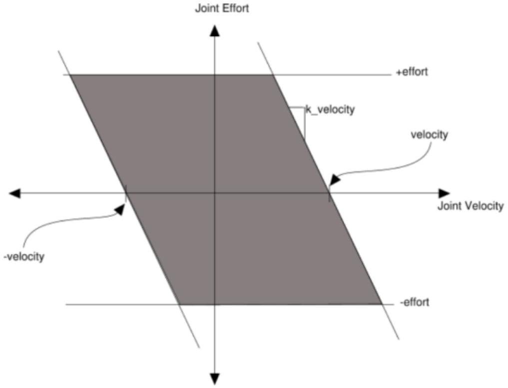
1. tau_f = tau_mu*sign(ω_m) + b*ω_m
2. P_f = tau_f.ω_m
3. tau_t = u_j/N + tau_f
4. P_t = tau_t.K_m.tau_t

Output (in time domain):

tau_j(t) (realised joint torque)

Performance measures (to be used as DVs in system design):

- tau_mu : Coulomb friction parameter
- b : Viscous friction parameter
- K_m : Motor constant
- N : Gear ratio
- Tau_max_m : Maximum torque
- ω_max_m : Maximum velocity
- m_m: : Mass of the motor
- I_m : Rotor inertia of the motor



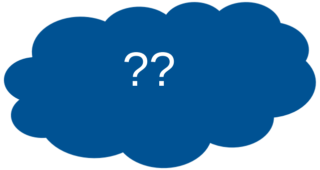
Basic control system

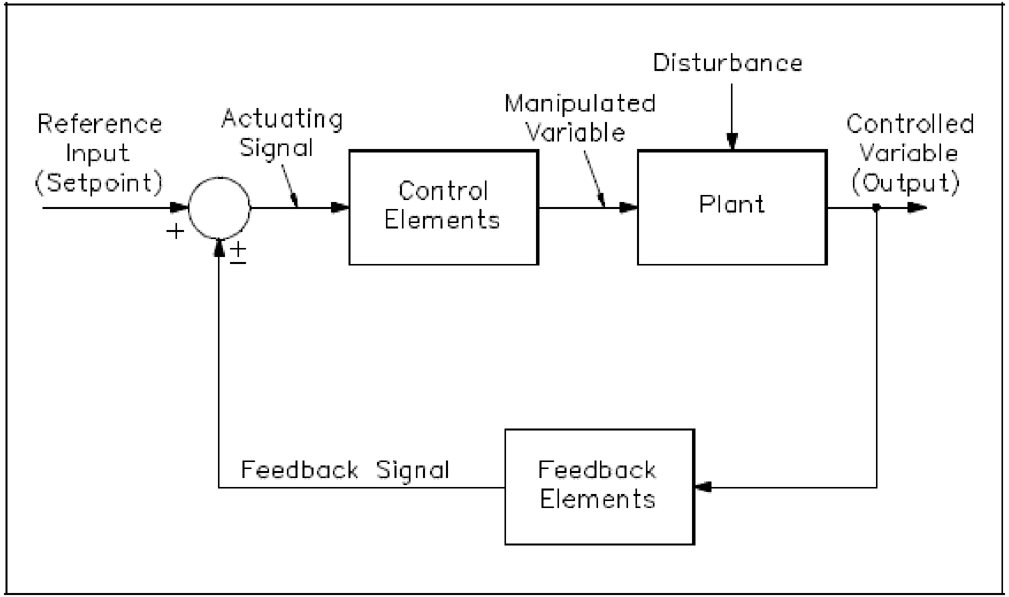
- We have our hardware ready. Now how do we control? Or more importantly what do we control?

- Open loop vs closed loop control

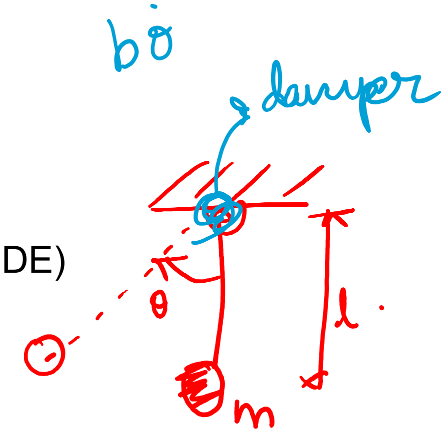
- Sensing and feedback

- Sensing or “observing” the entities we would like to control





Basic mathematics



Simple pendulum

- Understanding the system dynamics
  - Formulate the dynamics of the system as an ordinary differential equation (ODE)
  - Plot the phase portrait of the system
  - Understanding the phase portrait
- Simulating a system as a representative of a physical system
- Understanding the effects of the controller function

$u = k_p(ref - \theta)$

θ vs $\dot{\theta}$

dynamics $\rightarrow \ddot{\theta} = g(\theta, \dot{\theta})$

$\dot{x} = f(x)$

plot x vs $\dot{x}$

$l_2 \geq l_1$

$l_1 - l_2 \leq 0$

$\tau = I\alpha$

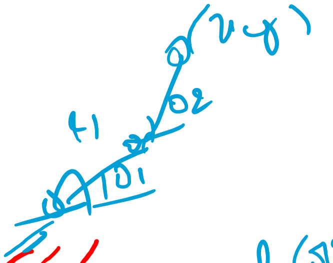
torque moment $\leftarrow \tau = (ml^2)\ddot{\theta}$

$$\underbrace{(mgl)\sin\theta}_{g} + \underbrace{b\dot{\theta}}_{\text{dan}} + u = ml^2\ddot{\theta}$$

$xy = l$

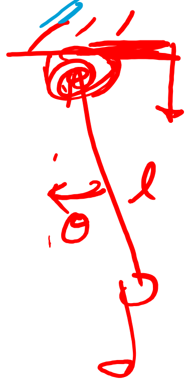
$xy - l = 0$

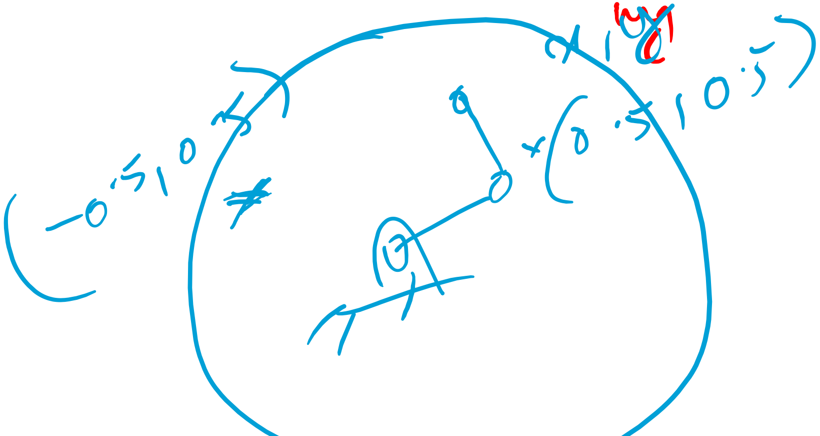
$$\begin{bmatrix} 1 & 0 & 0 \\ \cancel{2} & \cancel{0} & \cancel{0} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} l \\ 0 \\ 0 \\ 0 \end{bmatrix}$$



$l_1\cos\theta_1 + l_2\cos\theta_2 = x$

$l_1\sin\theta_1 + l_2\sin\theta_2 = y$

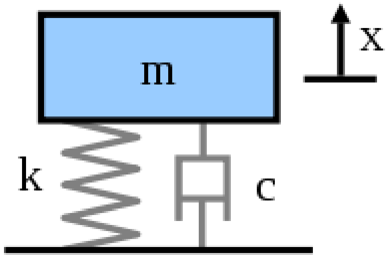




Basic control system

Getting acquainted with the Control Systems Toolbox:

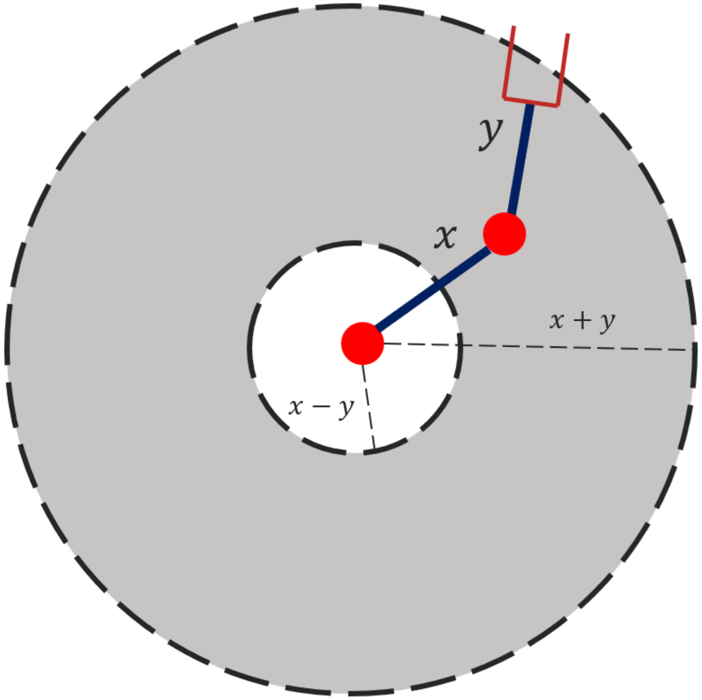
Playing with a PID controller using the Control Systems Toolbox



Basic control system

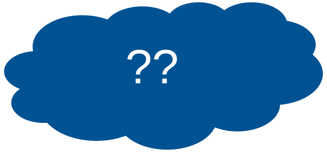
- Dealing with a more complicated robot control
- PID control of a two-link robot (Ghosal)
- Inverse dynamics controller [3]

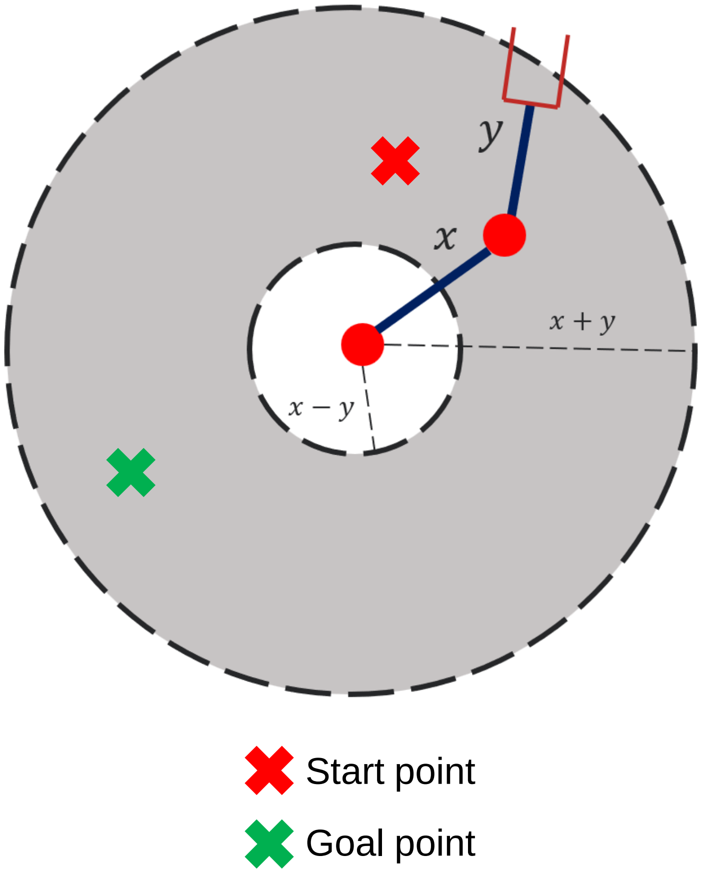
- Defining the control problem
  - If PID like the last case => Asymptotic convergance
  - Other ways to demand a control
    - Energy
    - Time
    - Staying far from an object
    - Achieve stability
    - Achieve a required external forces



Basic control system

- Posing the robot control as an optimization problem to minimize consumed energy as it moves between two points in the workspace





Part-2: Robot control as an optimization problem

Trajectory optimization: Formulation

- Formulating robot control as a general optimization problem

Objective

$$\min_{t_f,\, \boldsymbol{x}(\boldsymbol{t}),\, \boldsymbol{u}(\boldsymbol{t})} \boxed{J(t_0, t_f, x(t_0), x(t_f))} + \int_{t_0}^{t_f} \boxed{L(\tau, \big(x(\tau), u(\tau)\big) d\tau}$$

Lagrange term/ running cost

Mayer term/ terminal cost

Constraints

$\dot{x}(t) = f(t, x(t), u(t))$ → System dynamics

$g_1(t, x(t), u(t)) \leq 0$ → Path constraints

$g_2\big(t_0, t_f, x(t_0), x(t_f)\big) \leq 0$ → Boundary constraints

$h(t, x(t), u(t)) = 0$ → Equality constraints

Bounds

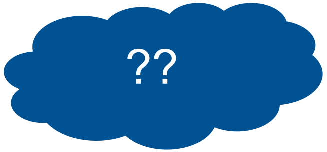
$x_{\text{low}} \leq x(t) \leq x_{\text{upp}}$ → State bounds

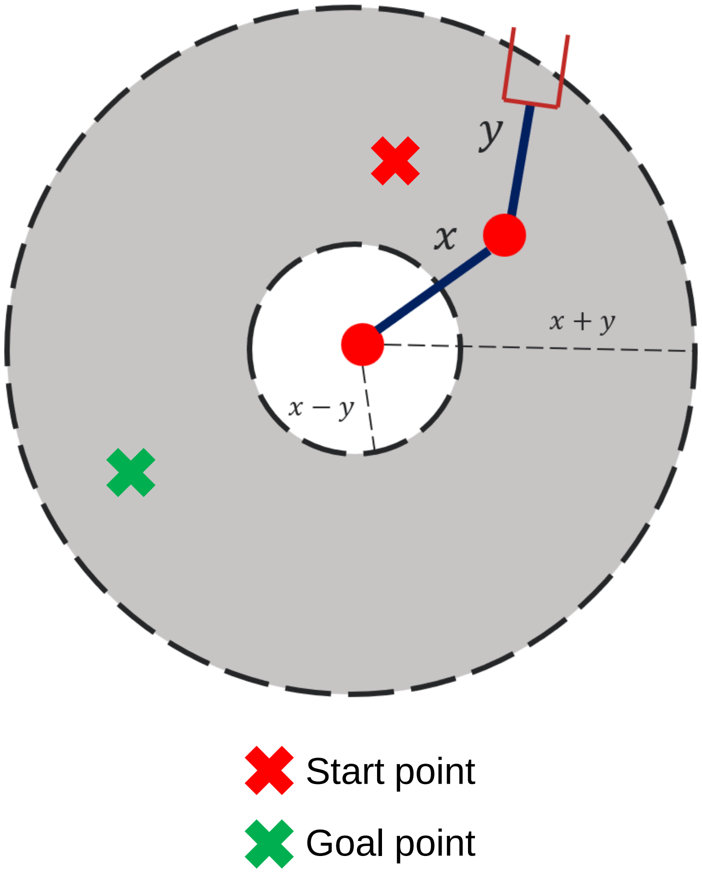
$u_{\text{low}} \leq u(t) \leq u_{\text{upp}}$ → Control bounds

Trajectory optimization: Formulation

- What kind of optimization problem does the formulation lead to?
  - **Direct transcription:** Solving the trajectory optimization problem by converting it into a non-linear program
  - What do you have available from the toolkit for such problems?

- Posing the robot control as an optimization problem to minimize consumed energy as it moves between two points in the workspace





Trajectory optimization: Formulation

- How do we complete the formulation such that we can implement it?
  - Discretize everything!

Trapezoidal collocation method

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

$$x_{k+1} - x_k \approx \frac{1}{2} h_k (f_{k+1} + f_k)$$

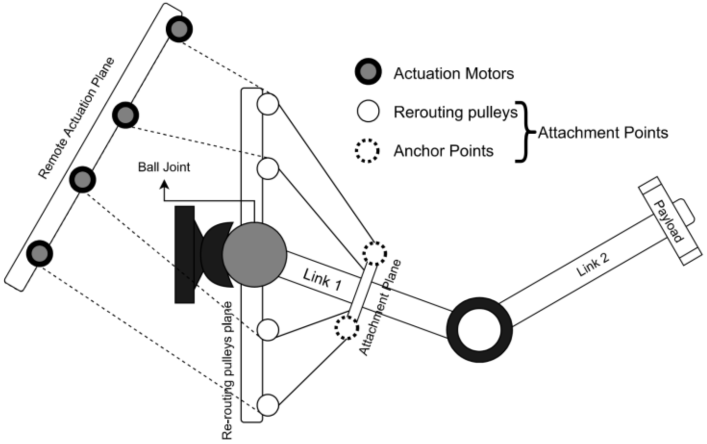
$$\int_0^{t_f} w\big(t, x(t), u(t)\big) dt \approx \frac{1}{2} \sum h_k (w_k + w_{k+1})$$

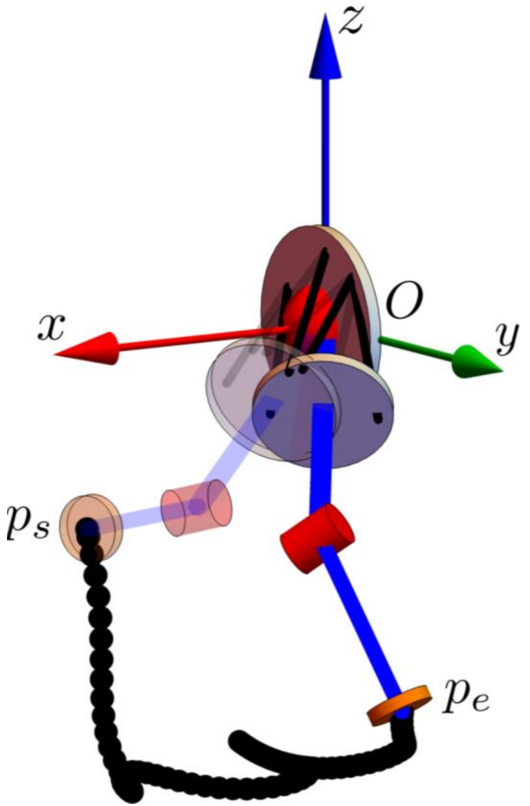
$$g\big(t, x(t), u(t)\big) < 0 \;\rightarrow g(t_k, x_k, u_k) < 0$$

Trajectory optimization: Formulation

- Formulating a controller for the musculo-skeletal robot we designed in the last class
- Be clever about the formulation to simplify computation!

Cable driven -> Joint based modeling!





Trajectory optimization: Discussion

- What happens when we try to control the system with the trajectory optimized controller?
  - Imperfect system modeling
  - Discretization of control and states
  - Imperfect sensing
  - Unknow objects in the environment
  - Its still open loop!
  - How do we then use trajectory optimization?
  - Read more about controllers like MPC, iLQR, RL etc.

Questions?

End of Day-3

References

[1] Kelly, Matthew. "An introduction to trajectory optimization: How to do your own direct collocation." *SIAM Review* 59.4 (2017): 849-904.

[2] Tedrake, Russ. "Underactuated robotics: Learning, planning, and control for efficient and agile machines course notes for MIT 6.832." *Working draft edition* 3 (2009) [https://underactuated.mit.edu/trajopt.html]

[3] Ghosal, Ashitava. *Robotics: fundamental concepts and analysis*. Oxford university press, 2006.

[4] Murray, Richard M., Zexiang Li, and S. Shankar Sastry. *A mathematical introduction to robotic manipulation*. CRC press, 2017.

[5] Task-Space Inverse Dynamics (TSID) [https://andreadelprete.github.io/]