Optimal control in biomechanics

July 25, 2021

Gil Serrancoli, Antoine Falisse, Friedl De Groote

Overview

- Background
- Hands-on example
- Examples of optimal control simulations of movement in research
- Resources and guidelines for setting up your own simulations

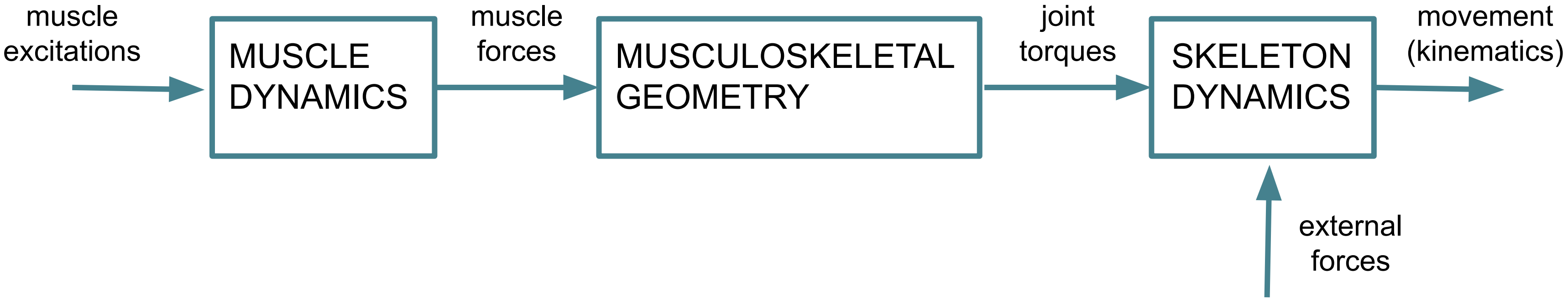
Overview

- Background
    - Simulating movement: system dynamics and optimal control
    - Solving optimal control problems
    - Improving computational efficiency

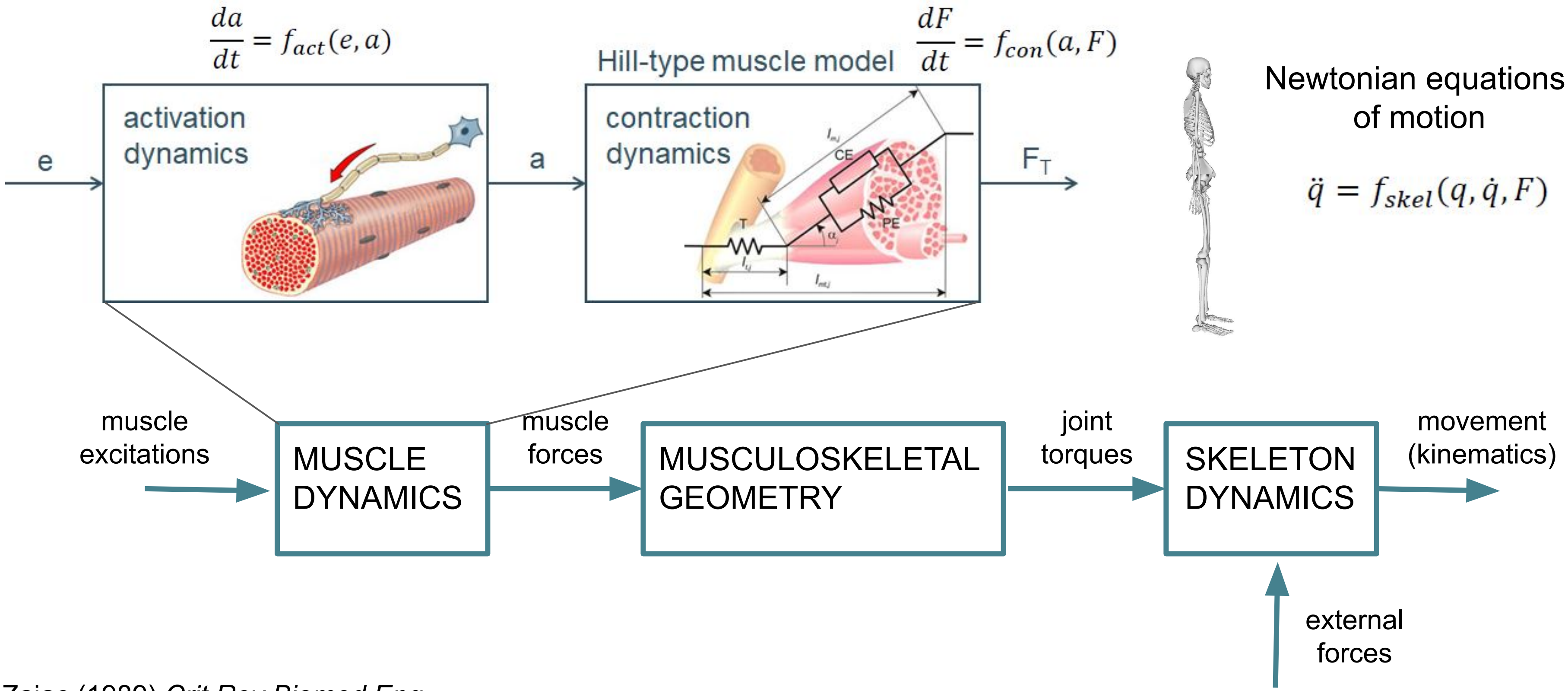
Simulations based on model of system dynamics

Figure: https://anatomyacademy.wordpress.com/chapters/chp-5-6/

Simulations based on model of system dynamics



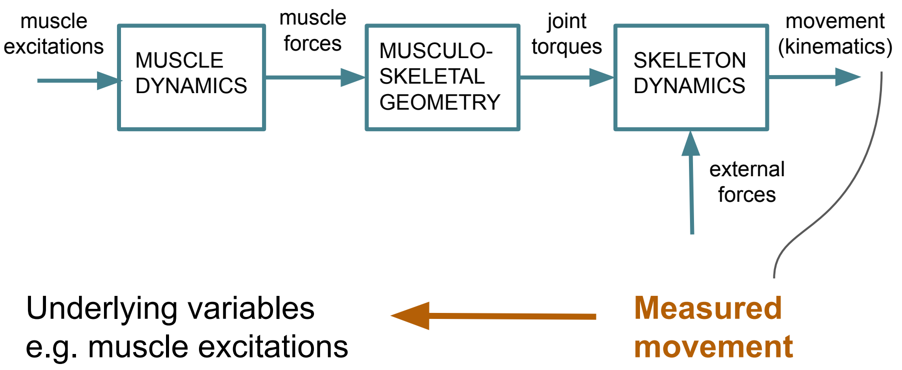
Simulations based on model of system dynamics



Zajac (1989) *Crit Rev Biomed Eng*

Inverse versus forward simulations of movement

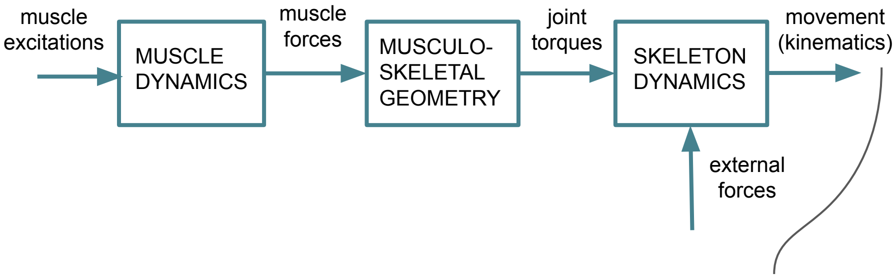
Inverse simulations



→ not possible to evaluate cause-effect relationship between musculoskeletal properties and movement pattern.

Inverse versus forward simulations of movement

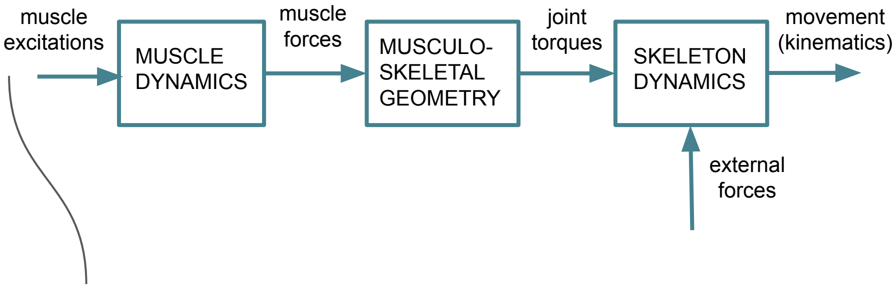
Inverse simulations



Underlying variables
e.g. muscle excitations ← **Measured movement**

→ not possible to evaluate cause-effect relationship between musculoskeletal properties and movement pattern.

Forward simulations



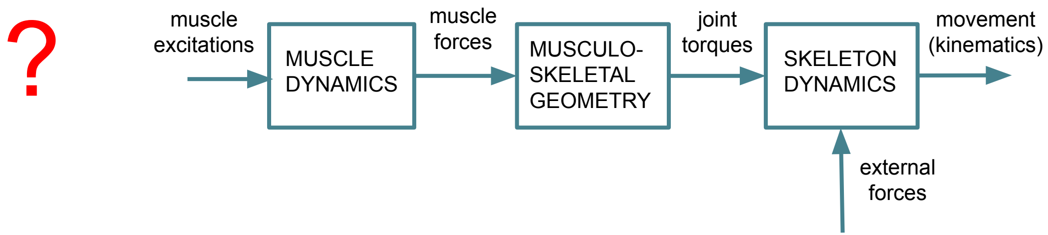
Control policy
e.g. muscle excitions → Movement

Challenges:

- Model of motor control needed
- Computational cost

Assumption of optimal control

dynamic constraints



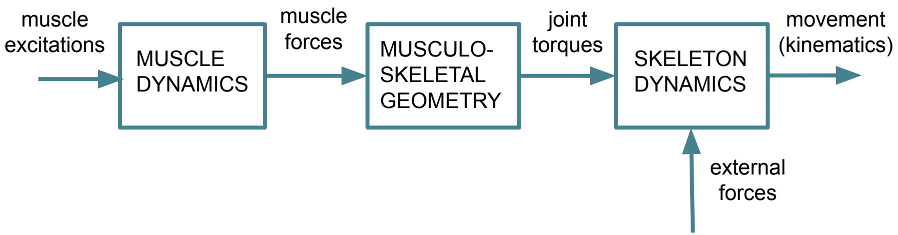
constraints describing task goal

e.g. average forward speed and periodicity for walking

Assumption of optimal control

dynamic constraints

?



constraints describing task goal

e.g. average forward speed and periodicity for walking

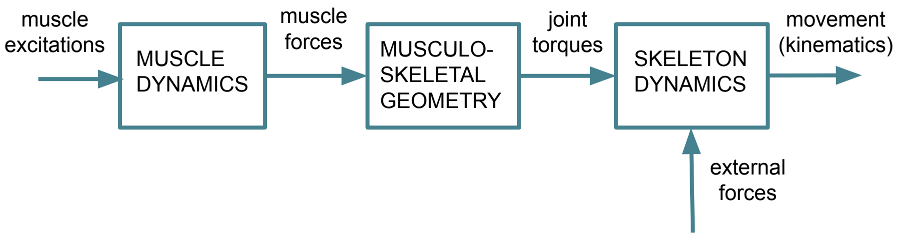
Monty Python: Ministry of silly walks

Assumption of optimal control

minimize cost

subject to

dynamic constraints



constraints describing task goal

e.g. average forward speed and periodicity for walking

with respect to muscle excitations

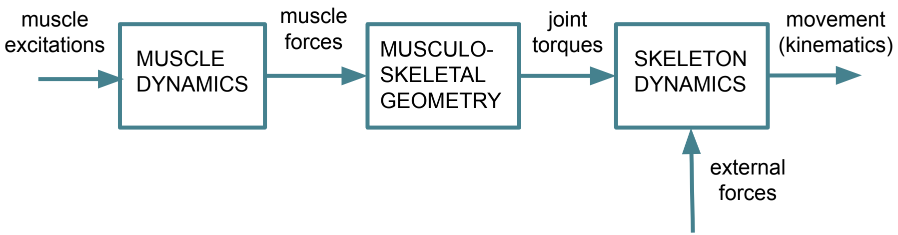
Assumption of optimal control

→ Trajectory optimization problem

minimize cost

subject to

dynamic constraints



constraints describing task goal

e.g. average forward speed and periodicity for walking

with respect to muscle excitations

Optimal control based on 3D musculoskeletal model yields physiologically-plausible gait patterns

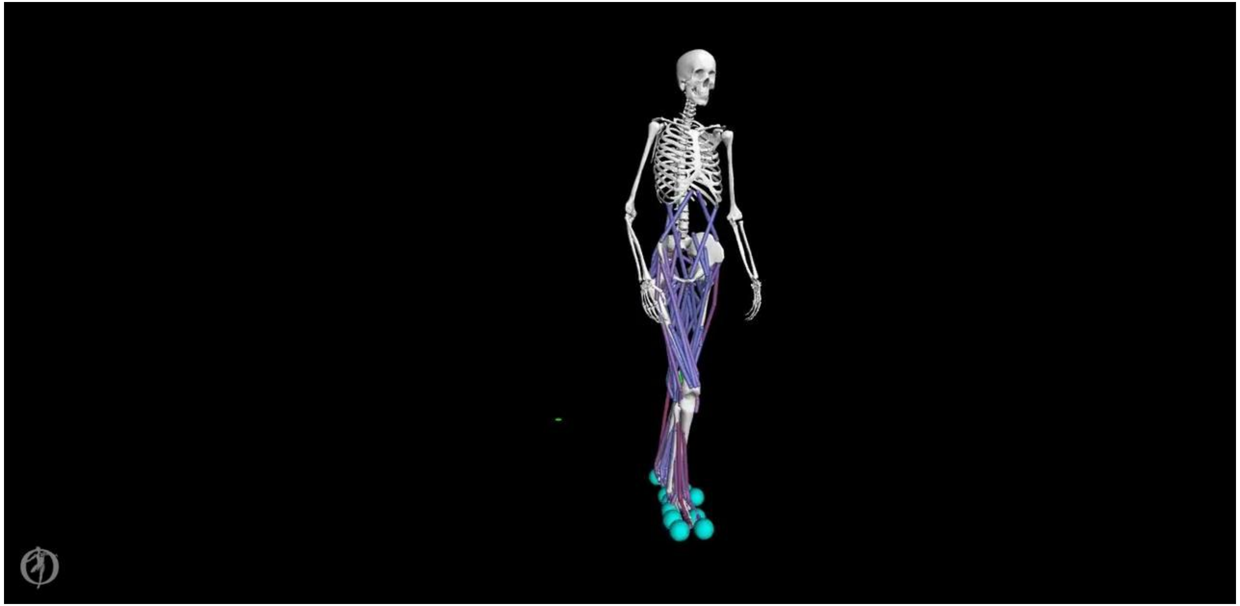
minimize

- metabolic energy rate,
- muscle activity,
- joint accelerations,
- passive joint torques,

subject to

- musculoskeletal dynamics
- desired speed
- symmetry

with respect to excitations



Falisse et al. (2019) *JRSI*

General formulation of trajectory optimization problem

minimize cost function

subject to dynamic constraints

path constraints (at each instant in time)

boundary constraints (initial and final time)

bounds

with respect to inputs

Example: prediction of periodic walking

Prediction of walking using a planar 5-dof model

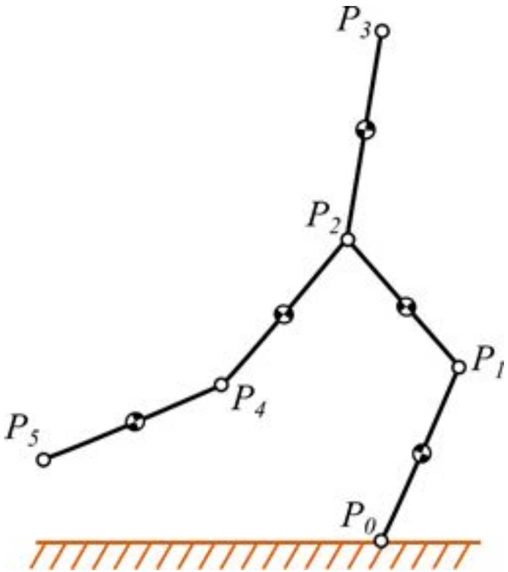
Kelly (2017) *SIAM Review*

Prediction of half a gait cycle

Imposing periodicity, step duration and length

Example: system dynamics

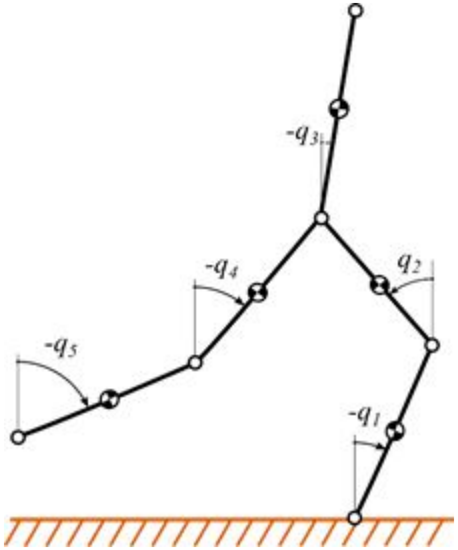
5-dof torque driven model



P_0 fixed to ground during stance phase

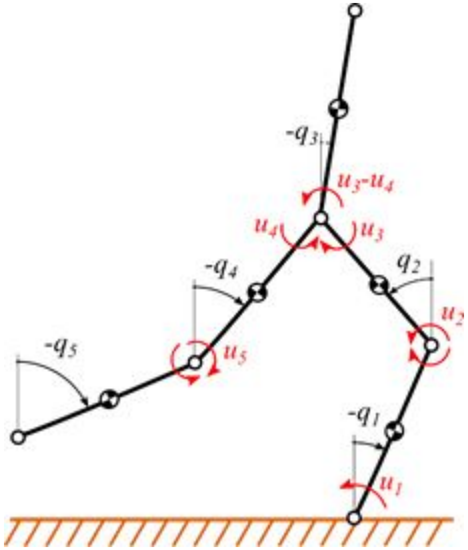
Example: system dynamics

5-dof torque driven model



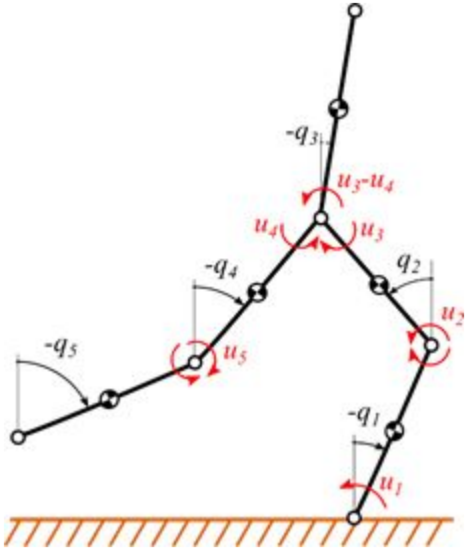
Example: system dynamics

5-dof torque driven model



Example: system dynamics

5-dof torque driven model



Dynamics

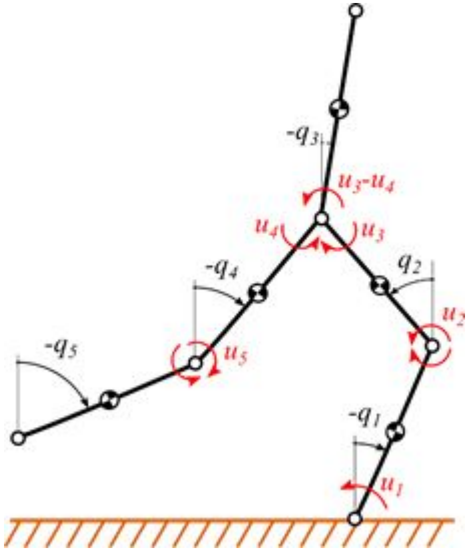
Equations of motion

joint torques

$$\left[M(q)\right]\ddot{q}+C(q,\dot{q})=\mathcal{F}(q,\dot{q},u)$$

Example: system dynamics

5-dof torque driven model



Dynamics

Equations of motion

joint torques

$$[M(q)]\ddot{q} + C(q,\dot{q}) = \mathcal{F}(q,\dot{q},u)$$

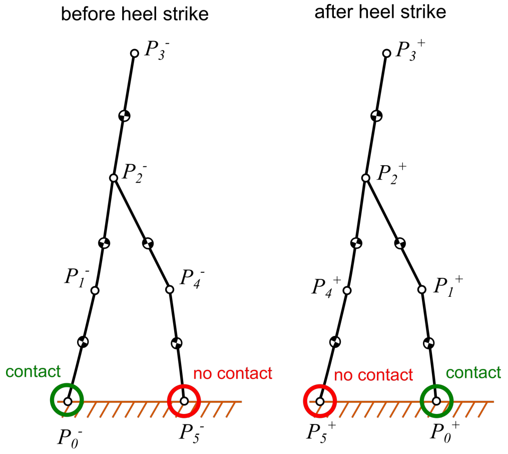
Impulsive collision

(conservation of angular momentum L)

$$L(P_0^-) = L(P_5^+)$$

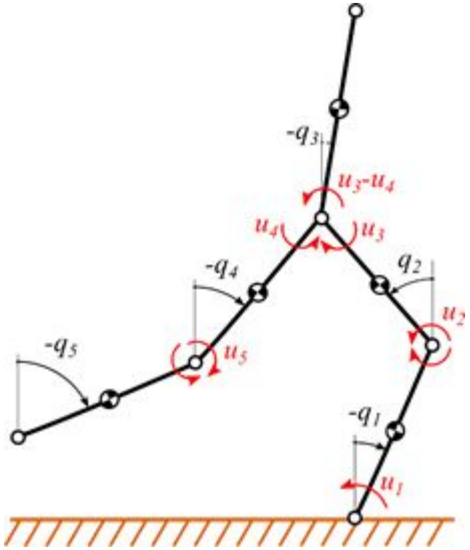
$$\dots$$

$$L(P_5^-) = L(P_0^+)$$



Example: system dynamics

5-dof torque driven model

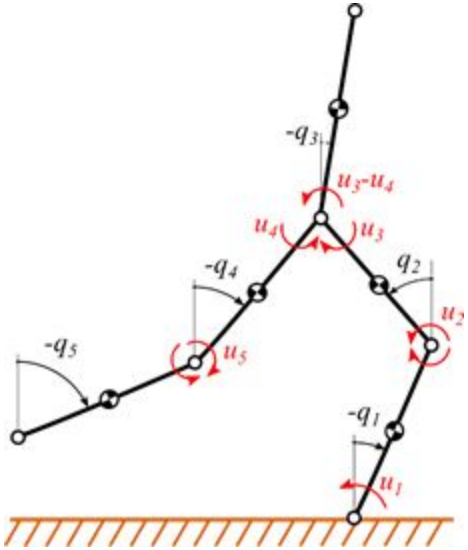


States and controls

States x: segments angles q and angular velocities $\dot{q}$

Example: system dynamics

5-dof torque driven model



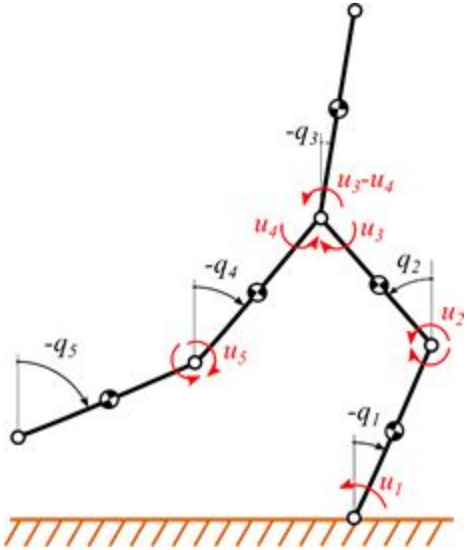
States and controls

States x: segments angles q and angular velocities $\dot{q}$

Controls u: joint torques

Example: system dynamics

5-dof torque driven model



States and controls

States x: segments angles q and angular velocities $\dot{q}$

Controls u: joint torques

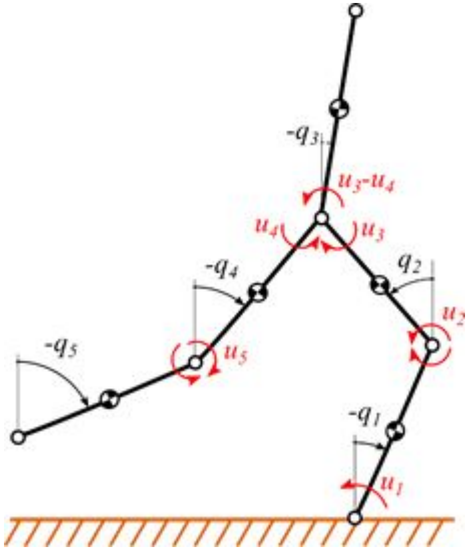
Bounds

$$x_{lb} \leq x \leq x_{ub}$$

$$u_{lb} \leq u \leq u_{ub}$$

Example: system dynamics

5-dof torque driven model



States and controls

States x: segments angles q and angular velocities $\dot{q}$

Controls u: joint torques

Bounds $x_{lb} \leq x \leq x_{ub}$

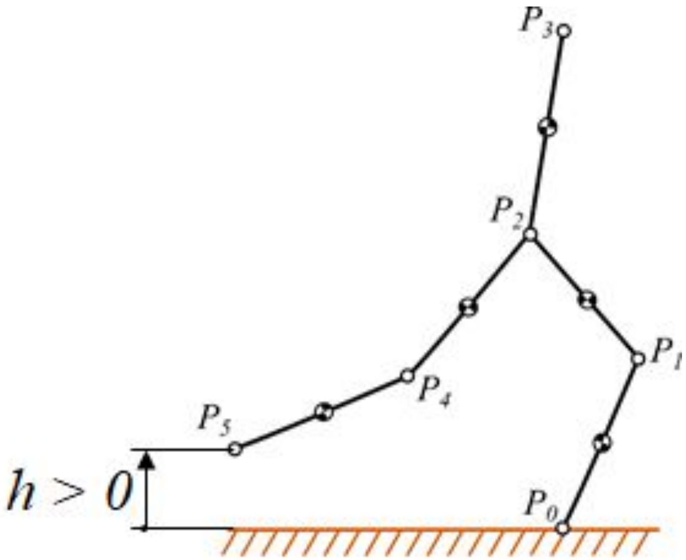
$u_{lb} \leq u \leq u_{ub}$

Constraints

Dynamic constraints: equations of motion

Example: system dynamics

5-dof torque driven model



States and controls

States x: segments angles q and angular velocities $\dot{q}$

Controls u: joint torques

Bounds $x_{lb} \leq x \leq x_{ub}$

$u_{lb} \leq u \leq u_{ub}$

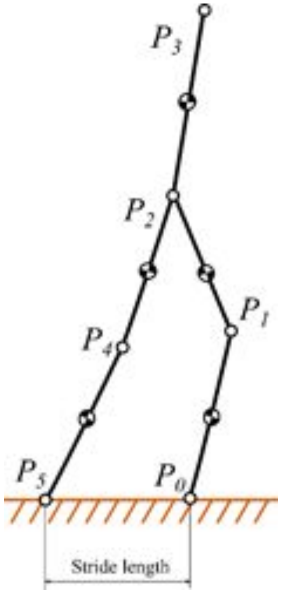
Constraints

Dynamic constraints: equations of motion

Path constraints: swing foot height (h > 0)

Example: system dynamics

5-dof torque driven model



States and controls

States x: segments angles q and angular velocities $\dot{q}$

Controls u: joint torques

Bounds $x_{lb} \leq x \leq x_{ub}$

$u_{lb} \leq u \leq u_{ub}$

Constraints

Dynamic constraints: equations of motion

Path constraints: swing foot height (h > 0)

Boundary constraints:

- Impulsive collision
- Stride length $\left(\dot{y}_{foot} < 0\right)$ $\left(\dot{y}_{foot} > 0\right)$
- Swing foot velocities at heel-strike and toe-off

Example: system dynamics

5-dof torque driven model

Cost function

$$J = \int_{t_0}^{t_f} \sum_{i}^{nq} u_i^2(t)dt$$

Sum of squared joint torques

States and controls

States x: segments angles q and angular velocities $\dot{q}$

Controls u: joint torques

Bounds $x_{lb} \leq x \leq x_{ub}$

$u_{lb} \leq u \leq u_{ub}$

Constraints

Dynamic constraints: equations of motion

Path constraints: swing foot height (h > 0)

Boundary constraints:

Impulsive collision

Stride length $(\dot{y}_{foot} < 0)$ $(\dot{y}_{foot} > 0)$

Swing foot velocities at heel-strike and toe-off

Example: system dynamics

5-dof torque driven model

Cost function

$$J = \int_{t_0}^{t_f} \sum_{i}^{nq} w_i u_i^2(t)\,dt$$

Weighted sum of squared joint torques

States and controls

States x: segments angles q and angular velocities $\dot{q}$

Controls u: joint torques

Bounds $x_{lb} \leq x \leq x_{ub}$

$u_{lb} \leq u \leq u_{ub}$

Constraints

Dynamic constraints: equations of motion

Path constraints: swing foot height (h > 0)

Boundary constraints:

- Impulsive collision
- Stride length
- Swing foot velocities at heel-strike and toe-off $(\dot{y}_{foot} < 0)$ $(\dot{y}_{foot} > 0)$

Overview

- Background
  - Simulating movement: system dynamics and optimal control
  - Solving optimal control problems
  - Improving computational efficiency

Solving trajectory optimization problems

minimize $$\int_{t_0}^{t_f} u(t)^2 dt$$ cost function

subject to $$\dot{x} = f(x,u)$$ dynamic constraints

$$c(x,u) = 0$$ path constraints

continuous → discrete time

$u(t)$, $t = t_0 \dots t_f$ → $u_0\ u_1 \dots u_i \dots u_{Nu}$

$x(t)$, $t = t_0 \dots t_f$ → $x_0\ x_1 \dots x_k \dots x_N$

Direct shooting

guess u_i,x_0

Direct shooting

guess u_i, x_0 → simulate ("shoot")

for example using forward Euler: $$\frac{x_{k+1} - x_k}{\Delta t} = f(x_k, u_k)$$

$$x_1 = x_0 + \Delta t * f(x_0, u_0)$$

$$x_2 = x_1 + \Delta t * f(x_1, u_1)$$

$$x_3 = x_2 + \Delta t * f(x_2, u_2)$$

...

Direct shooting

guess u_i,x_0 → simulate (“shoot”) → evaluate cost function and constraints

for example using forward Euler: $$\frac{x_{k+1} - x_k}{\Delta t} = f(x_k, u_k)$$

$$x_1 = x_0 + \Delta t * f(x_0, u_0)$$

$$x_2 = x_1 + \Delta t * f(x_1, u_1)$$

$$x_3 = x_2 + \Delta t * f(x_2, u_2)$$

…

Direct shooting

guess u_i,x_0 → simulate ("shoot") → evaluate cost function and constraints

for example using forward Euler: $\frac{x_{k+1} - x_k}{\Delta t} = f(x_k, u_k)$

$$x_1 = x_0 + \Delta t * f(x_0, u_0)$$

$$x_2 = x_1 + \Delta t * f(x_1, u_1)$$

$$x_3 = x_2 + \Delta t * f(x_2, u_2)$$

...

generate new guess (NLP solver)

Direct shooting

guess u_i,x_0 → simulate (“shoot”) → evaluate cost function and constraints

for example using forward Euler: $\frac{x_{k+1} - x_k}{\Delta t} = f(x_k, u_k)$

$$x_1 = x_0 + \Delta t * f(x_0, u_0)$$

$$x_2 = x_1 + \Delta t * f(x_1, u_1)$$

$$x_3 = x_2 + \Delta t * f(x_2, u_2)$$

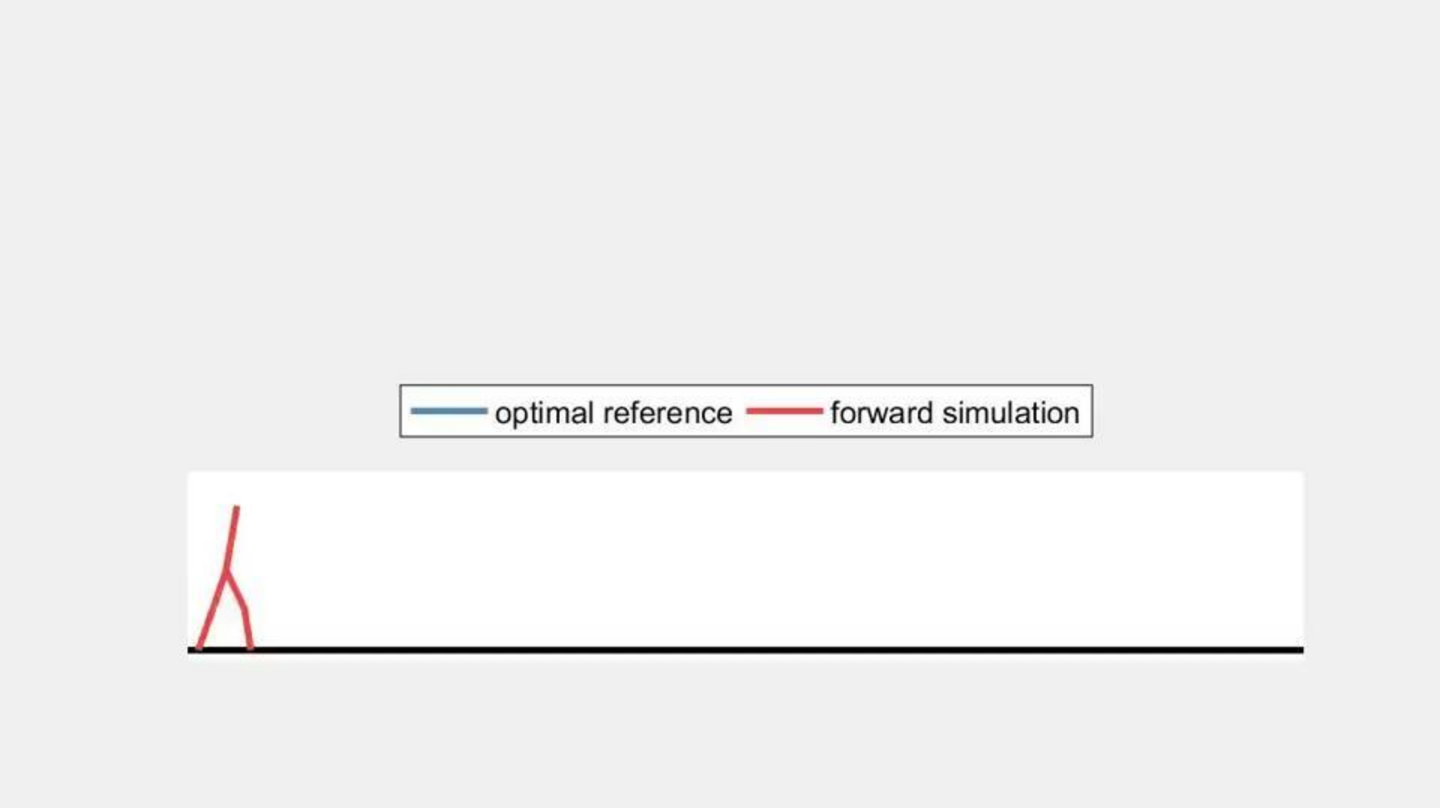
...

generate new guess (NLP solver)

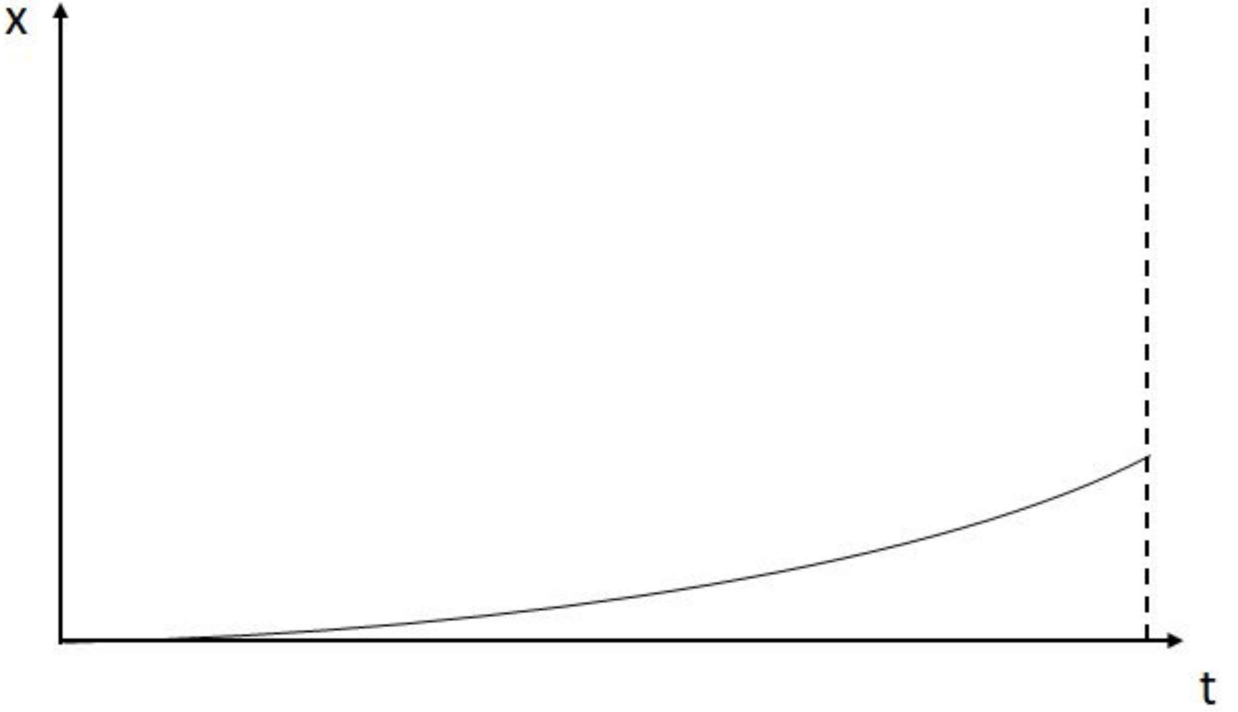
Problem: When system dynamics is stiff, the solution is very sensitive to the controls/initial state.
⇒ Hard to generate ‘better’ guess and long computational times.

Gait simulations are sensitive to controls

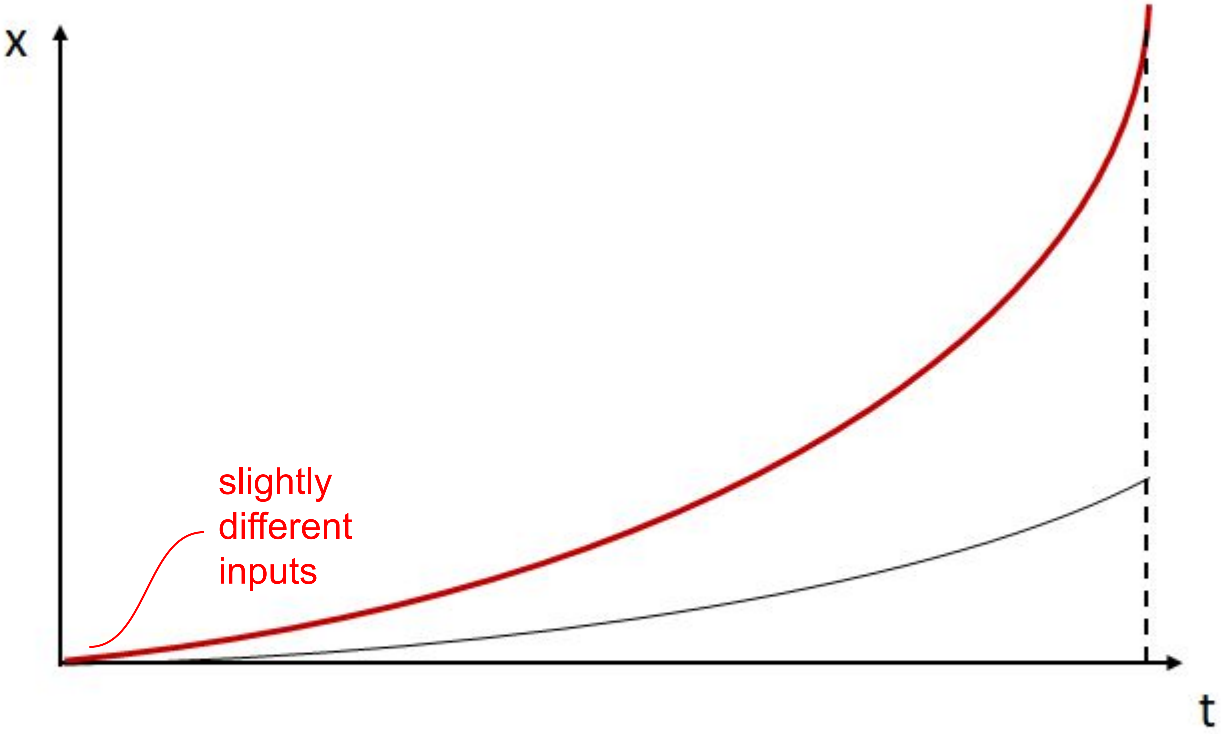
small change in joint torques → large change in kinematics



Direct shooting versus direct collocation: a cartoon

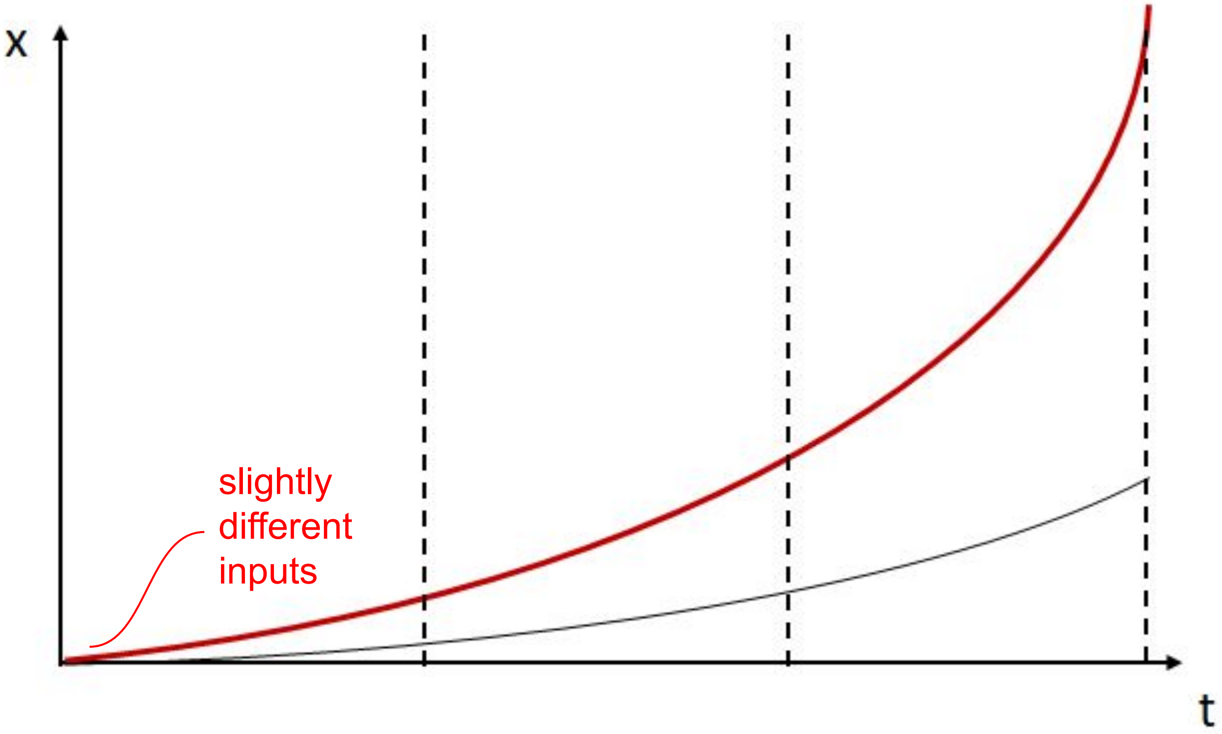


Direct shooting versus direct collocation: a cartoon



Shooting
⇒ state trajectory sensitive to controls

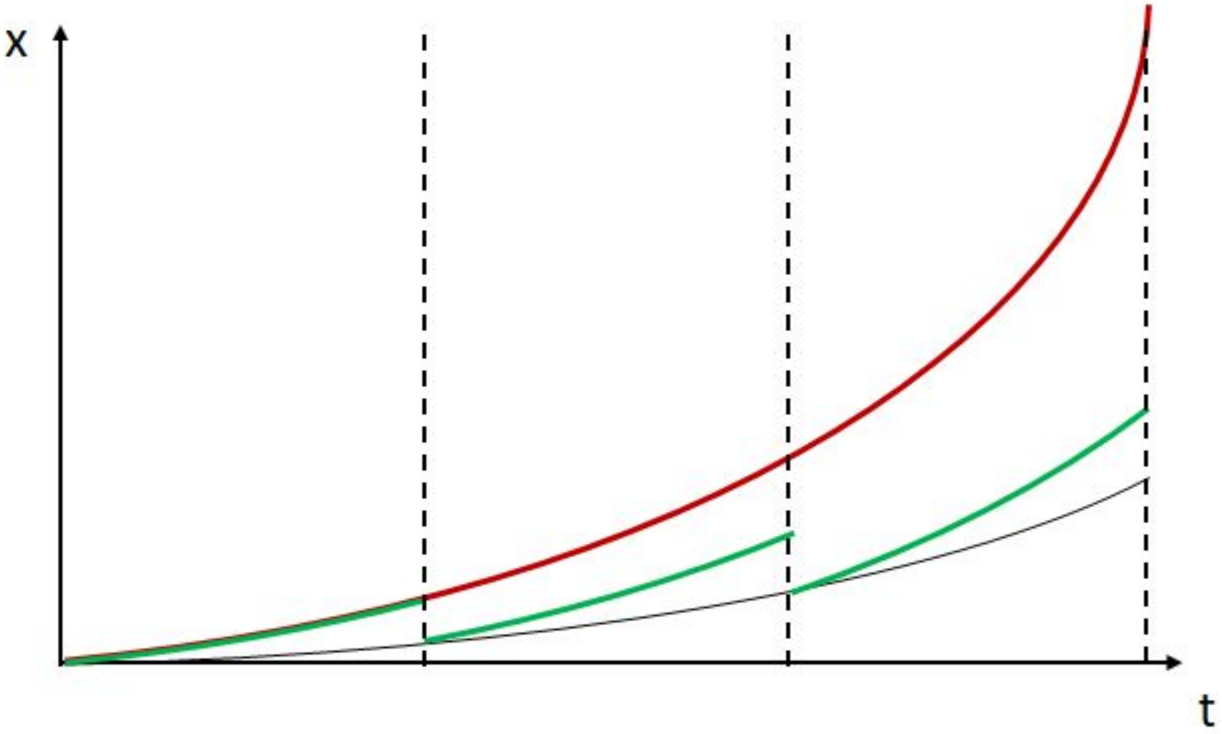
Direct shooting versus direct collocation: a cartoon



Shooting
⇒ state trajectory sensitive
to controls

Multiple shooting
⇒ integration over shorter
time horizon reduces
sensitivity

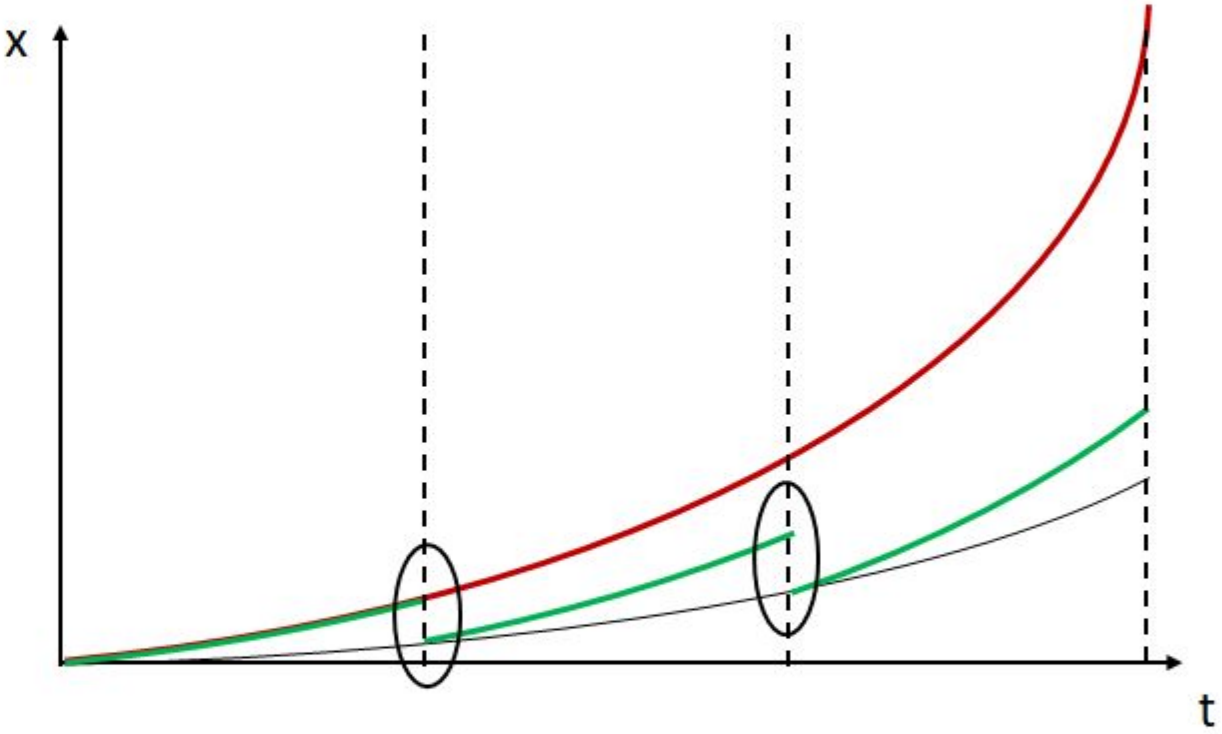
Direct shooting versus direct collocation: a cartoon



Shooting
⇒ state trajectory sensitive
to controls

Multiple shooting
⇒ integration over shorter
time horizon reduces
sensitivity

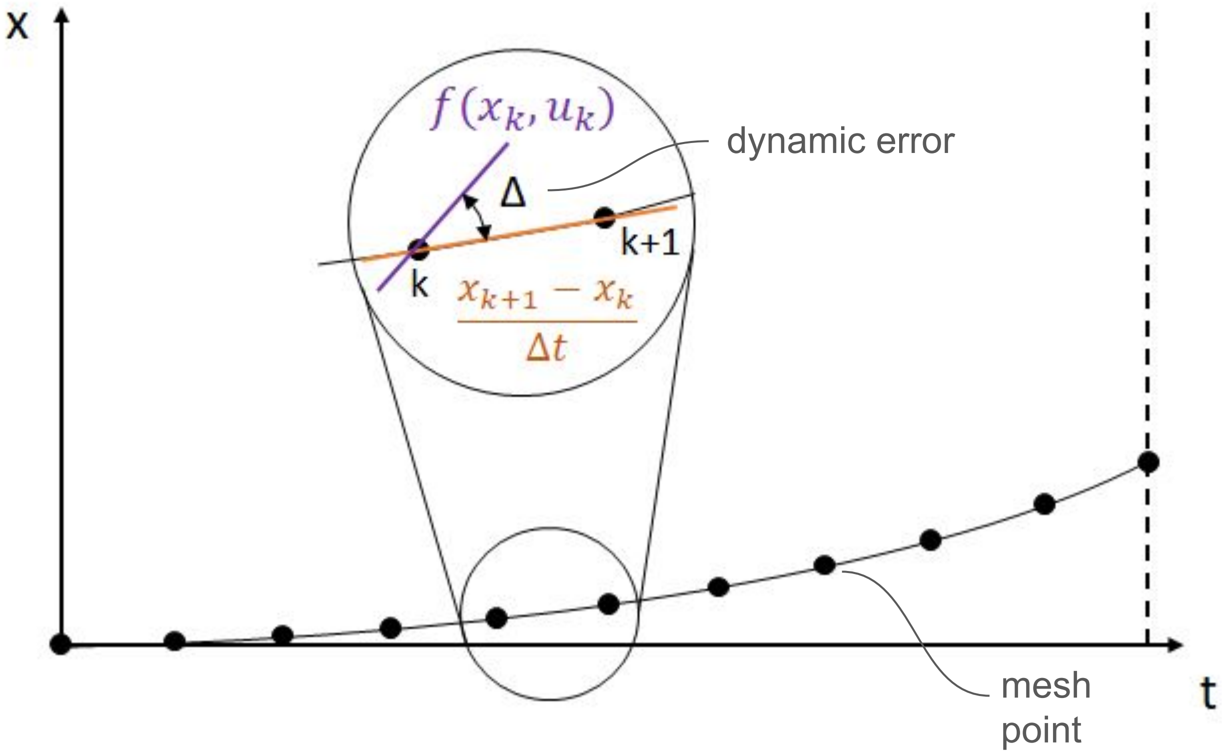
Direct shooting versus direct collocation: a cartoon



Shooting
⇒ state trajectory sensitive
to controls

Multiple shooting
⇒ integration over shorter
time horizon reduces
sensitivity

Direct shooting versus direct collocation: a cartoon



Shooting
⇒ state trajectory sensitive to controls

Multiple shooting
⇒ integration over shorter time horizon reduces sensitivity

Collocation
- polynomial approximation of state trajectory
- discretized dynamic equations imposed as constraints

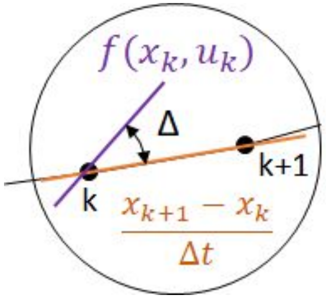
⇒ large optimization problem

Direct collocation

minimize $\sum_{k=0}^{N} u_k^2$

subject to $\frac{x_{k+1} - x_k}{\Delta t} = f(x_k, u_k)$

$$c(x_k, u_k) = 0$$

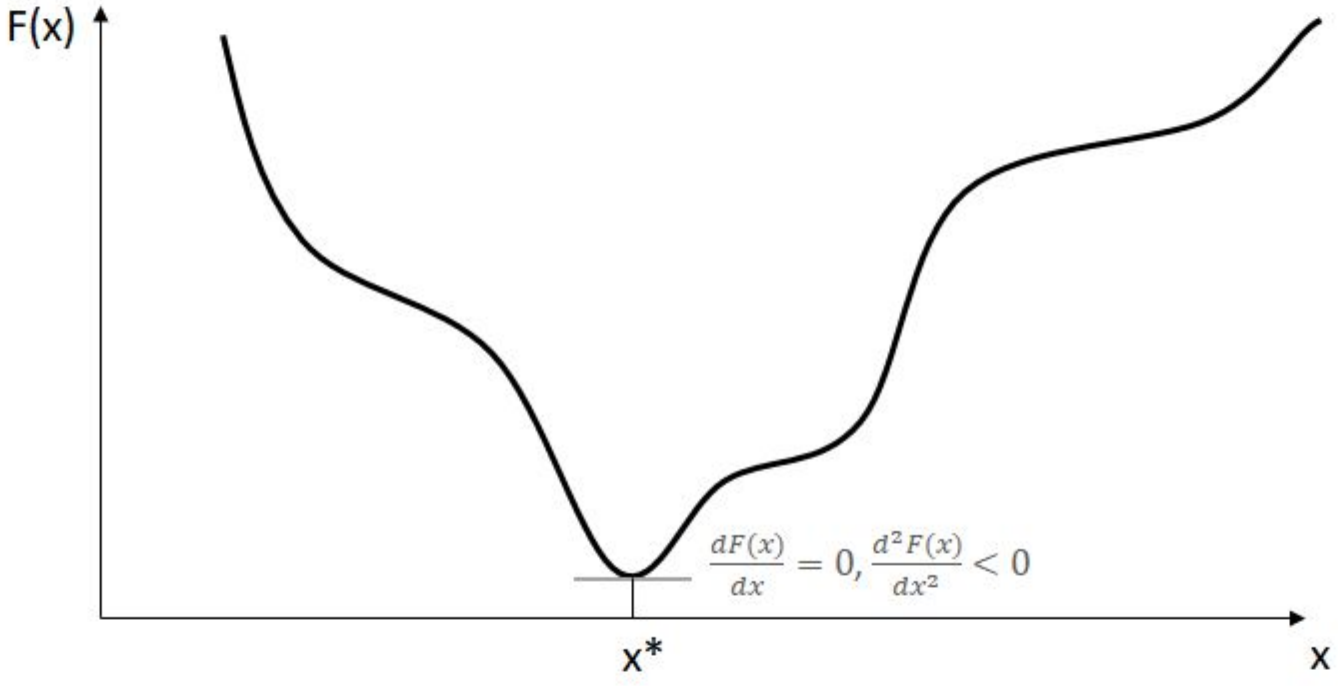


sparse problem
depends only on x_k, u_k, x_{k+1}

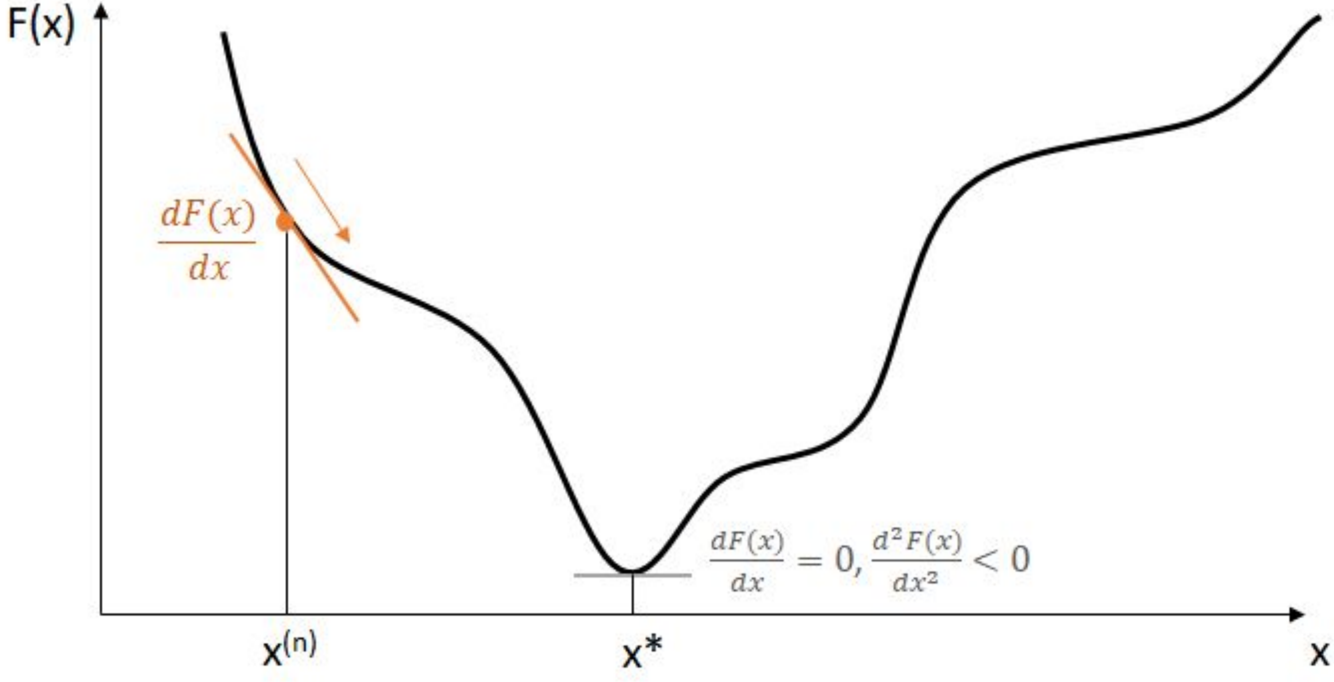
with respect to x_k, u_k

many optimization variables!

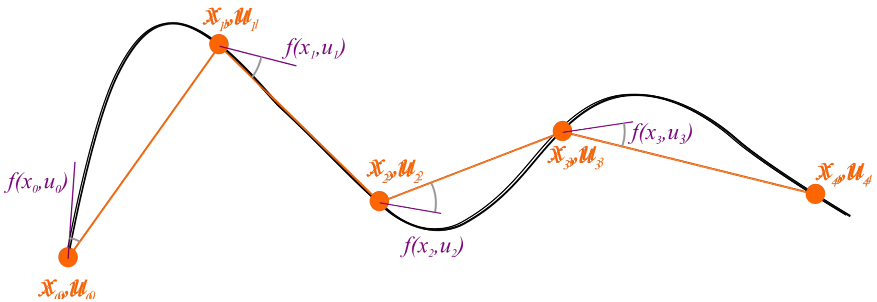
Gradient-based optimization



Gradient-based optimization

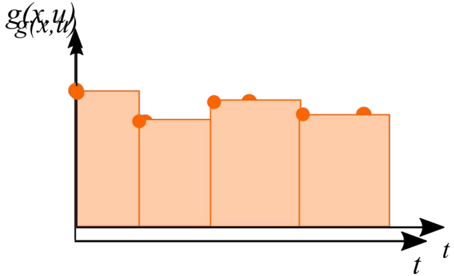


Collocation scheme



Dynamic constraints $(x_{k+1}-x_k)-f(x_k,u_k)\Delta t=0$

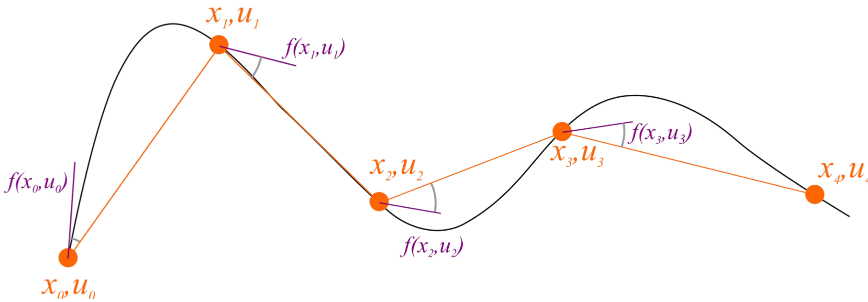
Cost function



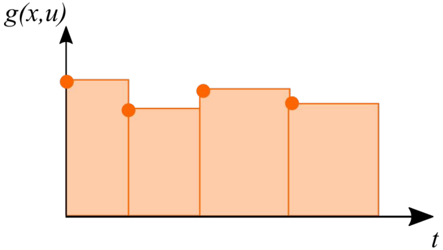
N mesh intervals

Collocation scheme

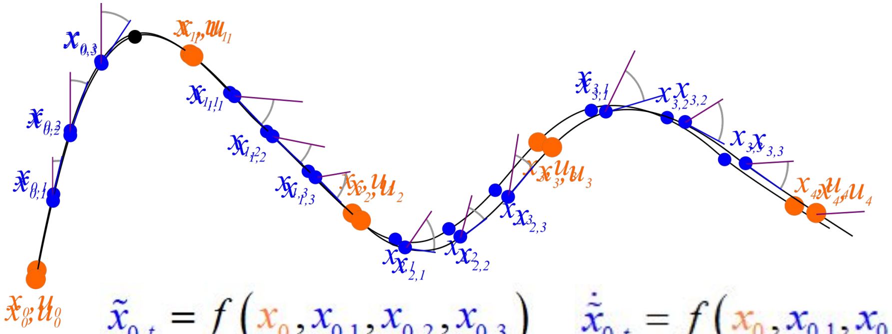
Dynamic constraints $(x_{k+1}-x_k)-f(x_k,u_k)\Delta t=0$



Cost function

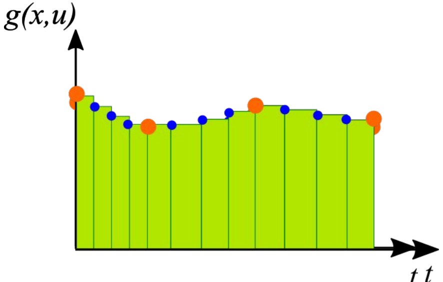


N mesh intervals



Dynamic constraints $\dot{\tilde{x}}_{k,r} - f\left(x_{k,r}, u_k\right) = 0$

Cost function



$$\tilde{x}_{0,t} = f\left(x_0, x_{0,1}, x_{0,2}, x_{0,3}\right) \quad \dot{\tilde{x}}_{0,t} = f\left(x_0, x_{0,1}, x_{0,2}, x_{0,3}\right)$$

States parameterized as Lagrange polynomials

Overview

- Background
    - Simulating movement: system dynamics and optimal control
    - Solving optimal control problems
    - Improving computational efficiency

Overview

- Background
    - Simulating movement: system dynamics and optimal control
    - Solving optimal control problems
    - Improving computational efficiency
        - Direct collocation
        - Explicit versus implicit formulation of the dynamics
        - Algorithmic differentiation (AD)

Explicit versus implicit formulation of the dynamics

Equations of motion

$$[M(q)]\ddot{q} + C(q,\dot{q}) = \mathcal{F}(q,\dot{q},u)$$ → state: $x = [q,\dot{q}]$

Explicit

dynamic constraints:

$$\dot{\hat{x}}_k - \left[\dot{q}_k, [M(q_k)]^{-1}\left(\mathcal{F}(q_k,\dot{q}_k,u_k) - C(q_k,\dot{q}_k)\right)\right] = 0$$

Potential convergence issues due to mass matrix inversion

Explicit versus implicit formulation of the dynamics

Equations of motion

$$\left[M(q)\right]\ddot{q}+C(q,\dot{q})=\mathcal{F}(q,\dot{q},u)$$

$\rightarrow$ state: $x=\left[q,\dot{q}\right]$

Explicit

dynamic constraints:

$$\dot{\tilde{x}}_k-\left[\dot{q}_k,\left[M(q_k)\right]^{-1}\left(\mathcal{F}(q_k,\dot{q}_k,u_k)-C(q_k,\dot{q}_k)\right)\right]=0$$

Potential convergence issues due to mass matrix inversion

Implicit

Introduce slack control u_a

dynamic constraints: $\dot{\tilde{x}}_k-\left[\dot{q}_k\, u_{a,k}\right]=0$

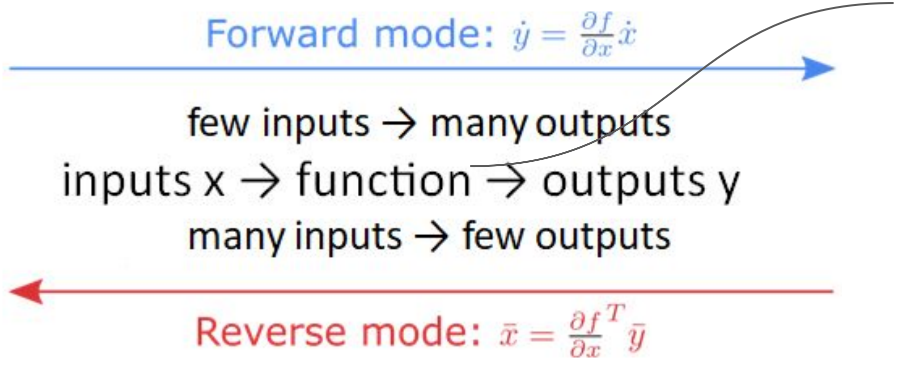
path constraints:

$$\left[M(q_k)\right]u_{a,k}+C(q_k,\dot{q}_k)-\mathcal{F}(q_k,\dot{q}_k,u_k)=0$$

More variables but results often in better convergence.

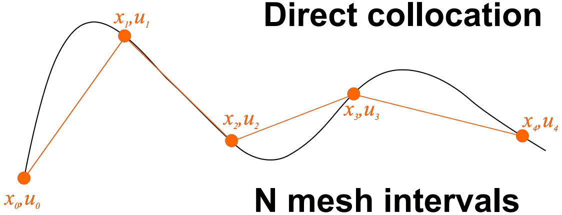
Algorithmic differentiation (AD)

AD generates derivatives based on code
by applying the chain rule to the underlying elementary operations



cost function, dynamics, path constraints, ...

Falisse et al. (2019) *Plos one*



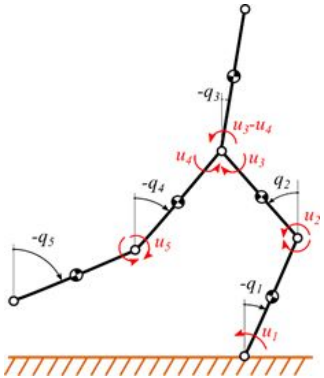
Example: system dynamics

5-dof torque driven model

Cost function

$$J = \int_{t_0}^{t_f} \sum_{i}^{nq} w_i u_i^2(t)dt$$

Weighted sum of squared joint torques



States and controls

States x: segments angles q and angular velocities $\dot{q}$

Controls u: joint torques

Bounds

$$x_{lb} \leq x \leq x_{ub}$$

$$u_{lb} \leq u \leq u_{ub}$$

Constraints

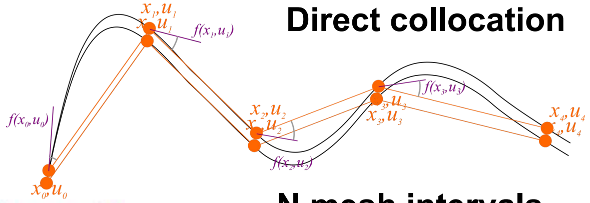
Dynamic constraints: equations of motion

Path constraints: swing foot height (h > 0)

Boundary constraints:

- Impulsive collision
- Stride length $(\dot{y}_{foot} < 0)$ $(\dot{y}_{foot} > 0)$
- Swing foot velocities at heel-strike and toe-off

Summary



Direct collocation

N mesh intervals

Design variables

- States x: segment angles q and angular velocities $\dot{q}$ ⟶ $x_k = [q_k, \dot{q}_k]$
- Controls u: joint torques and accelerations ⟶ $u_k = [u_{T,k}, u_{a,k}]$

Bounds ⟶ $x_{lb} \le x_k \le x_{ub} \quad u_{lb} \le u_k \le u_{ub}$

Constraints:

- Dynamic constraints ⟶ $c_{1,k} = q_{k+1} - q_k - \dot{q}_{k+1}\Delta t = 0 \quad c_{2,k} = \dot{q}_{k+1} - \dot{q}_{k+1} - u_{a,k+1}\Delta t = 0$
- Path constraints: equations of motion ⟶ $c_3(x_k, u_k) = 0$

  swing foot height ⟶ $c_4(x_k) \le 0$
- Boundary constraints:

  impulsive collision ⟶ $c_5(x_0, u_0, x_N, u_N) = 0$

  stride length and velocities at “heel strike” $(\dot{y}_{foot} < 0)$ and “toe off” $(\dot{y}_{foot} > 0)$ ⟶ $c_6(x_0, x_N) \le 0$

Cost function

$$J = \int_{t_0}^{t_f} \sum_i^{nq} w_i u_i^2(t)\,dt$$

Weighted sum of squared joint torques ⟶ $J = \sum_k^N h \sum_i^{nq} w_i \left(u_{i,k+1}\right)^2$

Tools

- Python or MATLAB - define problem
- CasADi - interface to NLP solver and automatic differentiation.
- Ipopt - NLP solver

Overview

- Background
- Hands-on example
- Examples of optimal control simulations of movement in research
- Resources and guidelines for setting up your own simulations

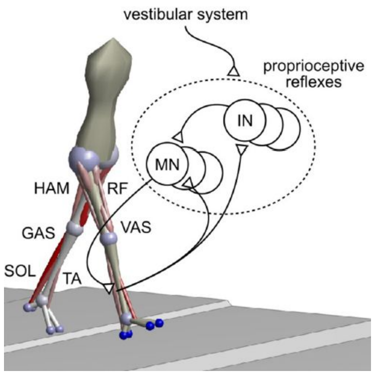
https://github.com/antoinefalisse/ISB21-workshop

Generate your own simulations

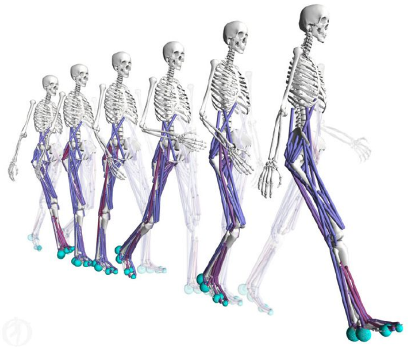
- Change the model parameters
  - e.g. explore sensitivity to segment masses and lengths.
- Change the bounds and constraints
  - e.g. constrain joint torques to simulate weakness.
  - e.g. simulate a knee orthosis that allows small flexion.
- Change the cost function
  - e.g. minimize joint accelerations more strongly.
- Change the problem settings
  - e.g. change the stride time or length.
- Change the numerical settings
  - e.g. decrease/increase the mesh size.

Overview

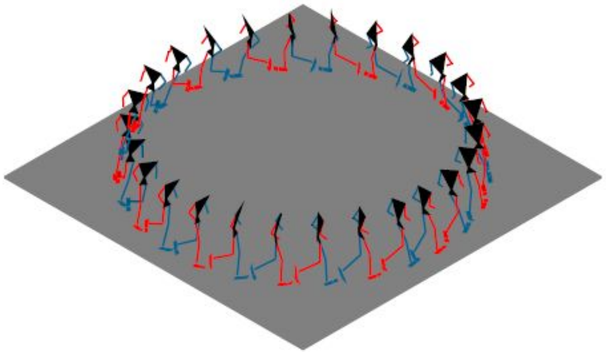
- Background
- Hands-on example
- Examples of optimal control simulations of movement in research
- Resources and guidelines for setting up your own simulations



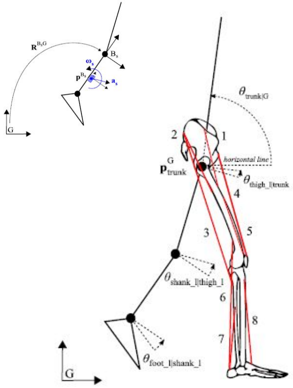
Song & Geyer (2015)
3D gait prediction



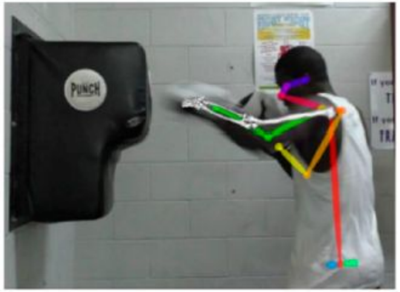
Falisse et al. (2019)
3D walking prediction



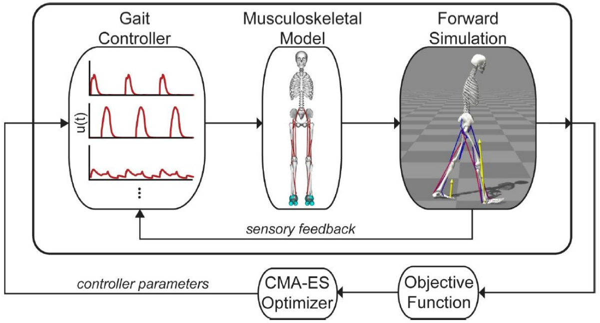
Nitsche et al. (2020)
3D curved running prediction



Dorschky et al. (2019)
2D walking tracking (IMU)



Haralabidis et al. (2020)
2D boxing tracking (IMU and video)



Ong et al. (2019)
2D impaired walking prediction

Overview

- Background
- Hands-on example
- Examples of optimal control simulations of movement in research
- Resources and guidelines for setting up your own simulations

Resources and guidelines

- OpenSim Moco
  - Direct collocation tool integrated with OpenSim
  - https://opensim-org.github.io/opensim-moco-site/
- Scone
  - Single shooting tool integrated with OpenSim
  - https://scone.software
- Muscle redundancy solver
  - Direct collocation tool to solve inverse problems
  - https://github.com/KULeuvenNeuromechanics/MuscleRedundancySolver
- Other available repositories:
  - Fast 3D predictive simulations with direct collocation and AD: https://github. … efalisse/3dpredictsim
  - Bioptim (Biomechanical Optimal Control): https://github.com/pyomeca/biopt…
  - Other resources: https://github.com/modenaxe/awesome-biomechanics#optimal-control-and-trajectory-optimization-rocket
- OpenSim webinar: Which simulation pipeline should I use? An overview of common workflows
  - https://www.youtube.com/watch?v=e0gfkm5_Rps

Thanks