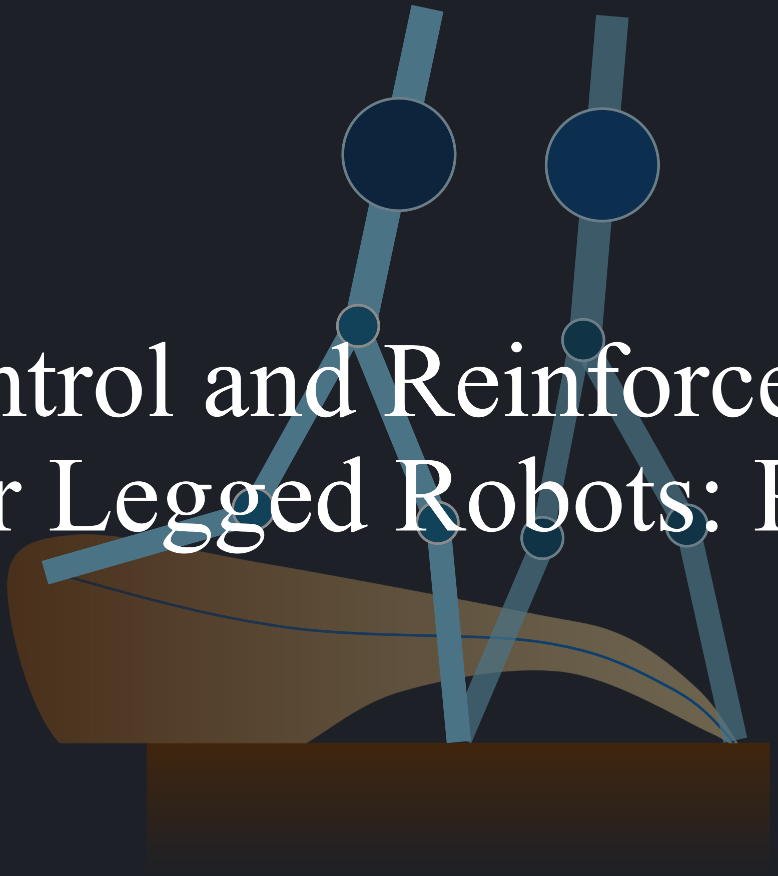
Optimal Control and Reinforcement Learning for Legged Robots: Part 1



Jason Jangho Choi

Berkeley
Mechanical Engineering

Apr. 17, 2023
EECS206B Guest Lecture

Model-based Optimal Control

Deep Reinforcement Learning

Deep Reinforcement Learning

Agility Robotics

Your first design decision:

Model-based (Optimal) Control? or Deep Reinforcement Learning?

Review of last lecture

	Optimal Control	Reinforcement Learning
System	$x_{k+1} = f(x_k, u_k)$ (x_k: state, u_k: control)	$\mathcal{T}_{ijk} = p(s_{t+1} = i \mid s_t = j, a_t = k)$ (s_t: state, a_t: action)
Objective	$V(x_0) = \min_{u(\cdot)} \sum_{k=0}^{\infty} \gamma^k c(x_k, u_k)$ (γ^k: discount, $c(x_k, u_k)$: stage cost)	$V(s_0) = \max_{\theta} E_{\tau \sim p_\theta(\tau)} \left[\sum_t \gamma^t r(s_t, a_t) \right]$ $p_\theta(\tau) = p_\theta(s_0, a_0, s_1, a_1, \ldots) = p(s_0) \prod_{t=0}^{\infty} \pi_\theta(a_t \mid s_t) p(s_{t+1} \mid s_t, a_t)$
Dynamic Programming (DP) Principle	$V(x_0) = \min_{u_0} [c(x_0, u_0) + \gamma V(x_1)]$	$V(s_0) = \max_{\theta} \left[E_{a_0 \sim \pi_\theta(a_0 \mid s_0)} [r(s_0, a_0)] + \gamma E_{s_1 \sim p(s_0)\pi_\theta(a_0 \mid s_0) p(s_1 \mid s_0, a_0)} [V(s_1)] \right]$

In the end, if both methods are solving the same problem, why should the choice of method matter?

Contents

- Why should the choice of method matter?
- Dynamics of legged robots
- Model-based optimal control

In the end, if both methods are solving the same problem, why should the choice of method matter?

My take:
A lot of model-based optimal control methods aim at obtaining solutions as close as possible to the "global and deterministic" optimal solution, within their computational limits.

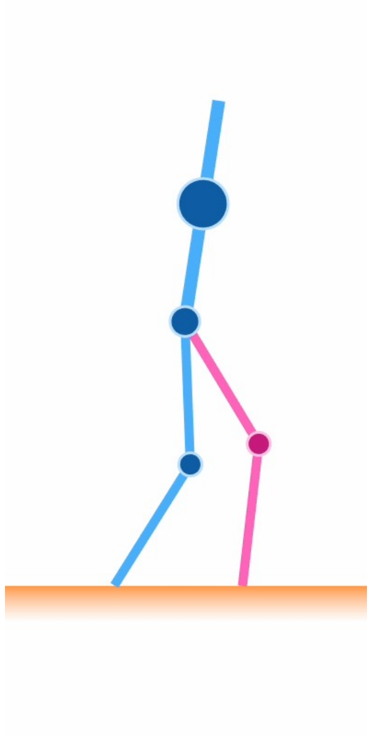
vs

A lot of deep reinforcement learning methods aim at obtaining generalizable, albeit suboptimal, solutions.

Dynamics of Legged Robots

(Simplified) Dynamic model of legged robots

- **Simple hybrid automaton:** One continuous dynamics mode with one reset map.

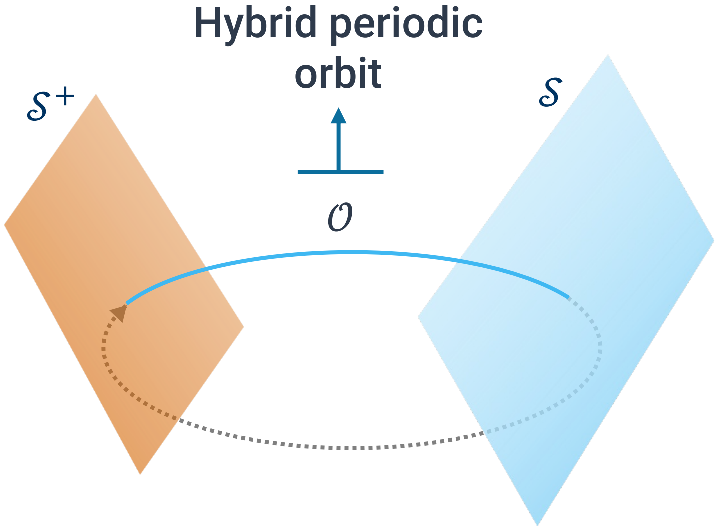


$$\dot{x} = f(x, u), \quad x \notin S$$

$$x^{+} = \Delta\left(x^{-}\right), \quad x^{-} \in S$$

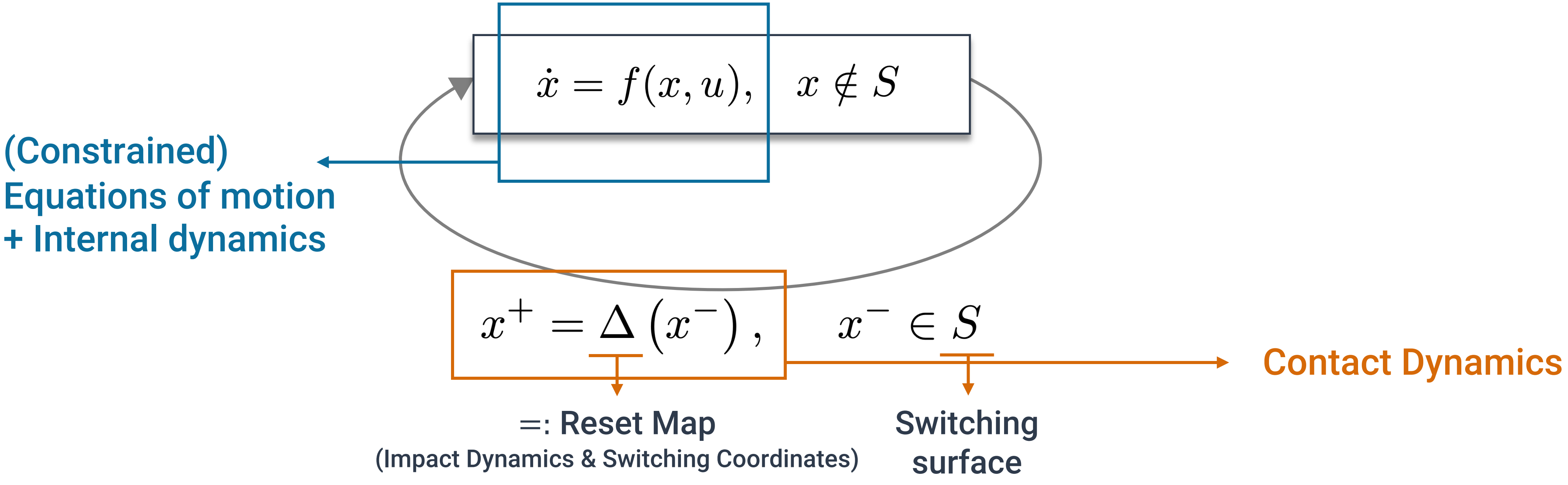
=: Reset Map
(Impact Dynamics & Switching Coordinates)

Switching surface



(Simplified) Dynamic model of legged robots

- **Simple hybrid automaton:** One continuous dynamics mode with one reset map.



Constrained Equations of Motion

EoM:

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = Bu + J(q)^\top \lambda$$

- $D(q)\ddot{q}$: Mass inertia
- $C(q,\dot{q})\dot{q}$: Coriollis
- $G(q)$: Gravity
- Bu: Input mapping
- $J(q)^\top \lambda$: Ground reaction (constraining) force

Constraint:

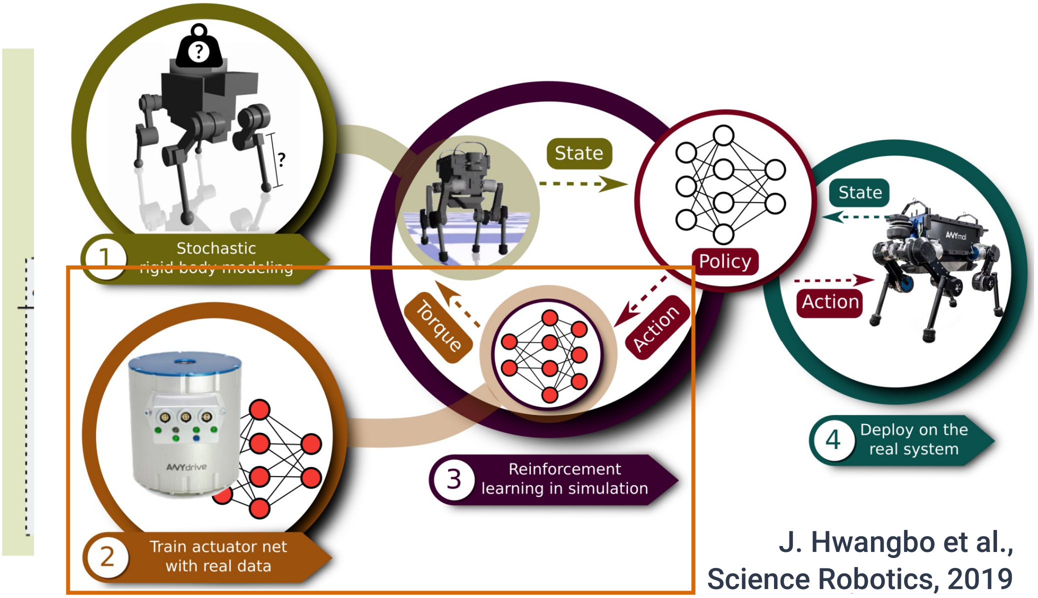
$$J(q)\dot{q} \equiv 0$$

Constrained EoM:

$$\begin{bmatrix} D(q) & -J^T \\ J & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{q} \\ \lambda \end{bmatrix} = \begin{bmatrix} -C(q,\dot{q})\dot{q} - G(q)Bu \\ -\dot{J}(q)\dot{q} \end{bmatrix}$$

Internal Dynamics

- Dynamics between commands and actual motor torques
  - Motor dynamics
  - Delays in control signals introduced by multiple hardware and software layers
  - Low-level controller dynamics, etc.



J. Hwangbo et al., Science Robotics, 2019

- Frictions, damping, compliance in mechanical components

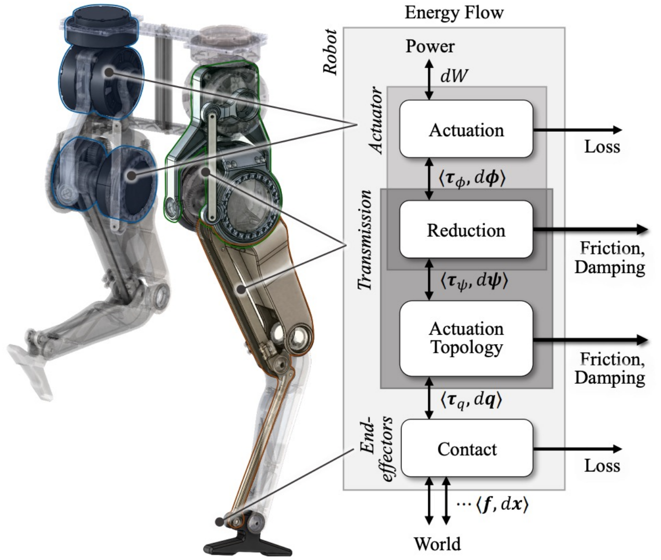
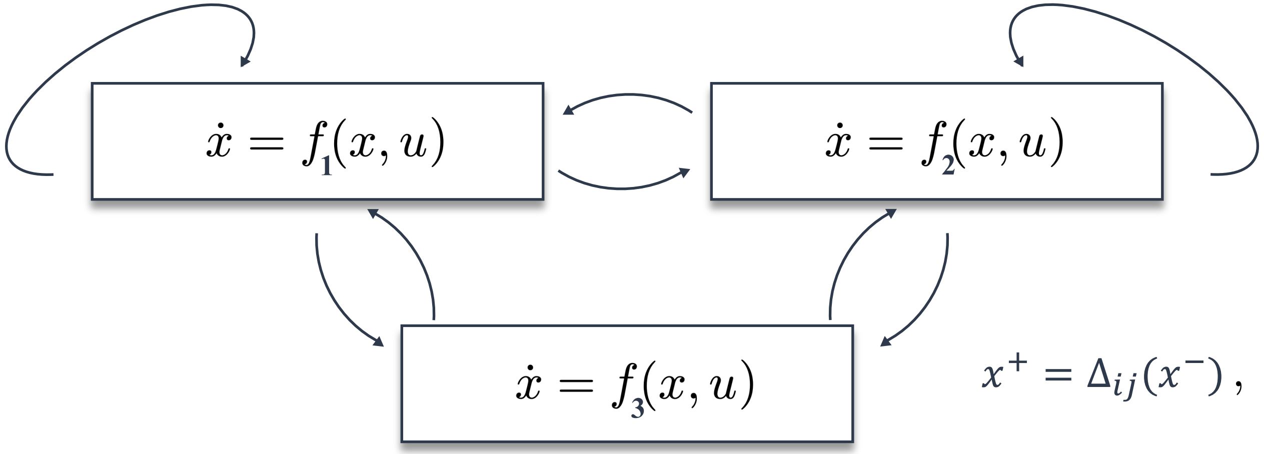


Fig. 1. Energy flow diagram of a robotic system showing the dissipation of energy in actuators and transmissions. The energy conversions are always accompanied by energy losses such as Joule heating or friction.

Y. Sim & J. Ramos, ICRA 2021

Contact Dynamics – Impact and friction

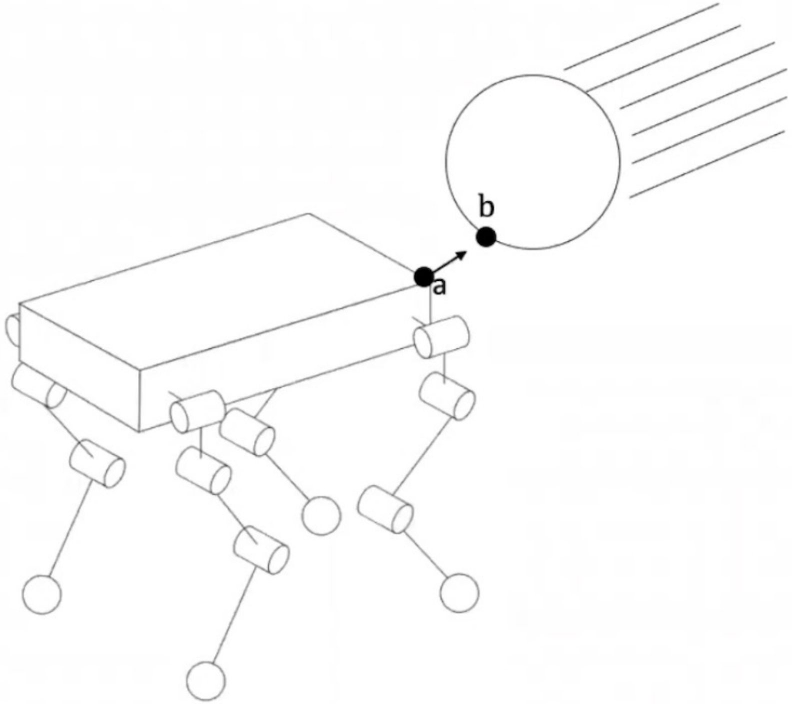
- Various impact dynamics
  - Rigid Impact
  - Elastic Impact
  - Compliant ground impact
- Various modes of contact
  - Stick
  - Slip
  - Open contact
- Various locations of contact



$$x^+ = \Delta_{ij}(x^-), \qquad x^- \in S_{ij}$$

Contact Dynamics – Impact and friction

- **Single Contact:**



Contact point dynamics

$$\lambda_{imp} = M_{imp}\Delta v_{imp} + c$$

What else do we know?

Complementarity condition:

$$(\Delta v_{imp} + v_{imp})_z \geq 0$$
$$\lambda(\Delta v_{imp} + v_{imp})_z = 0$$

Friction cone:

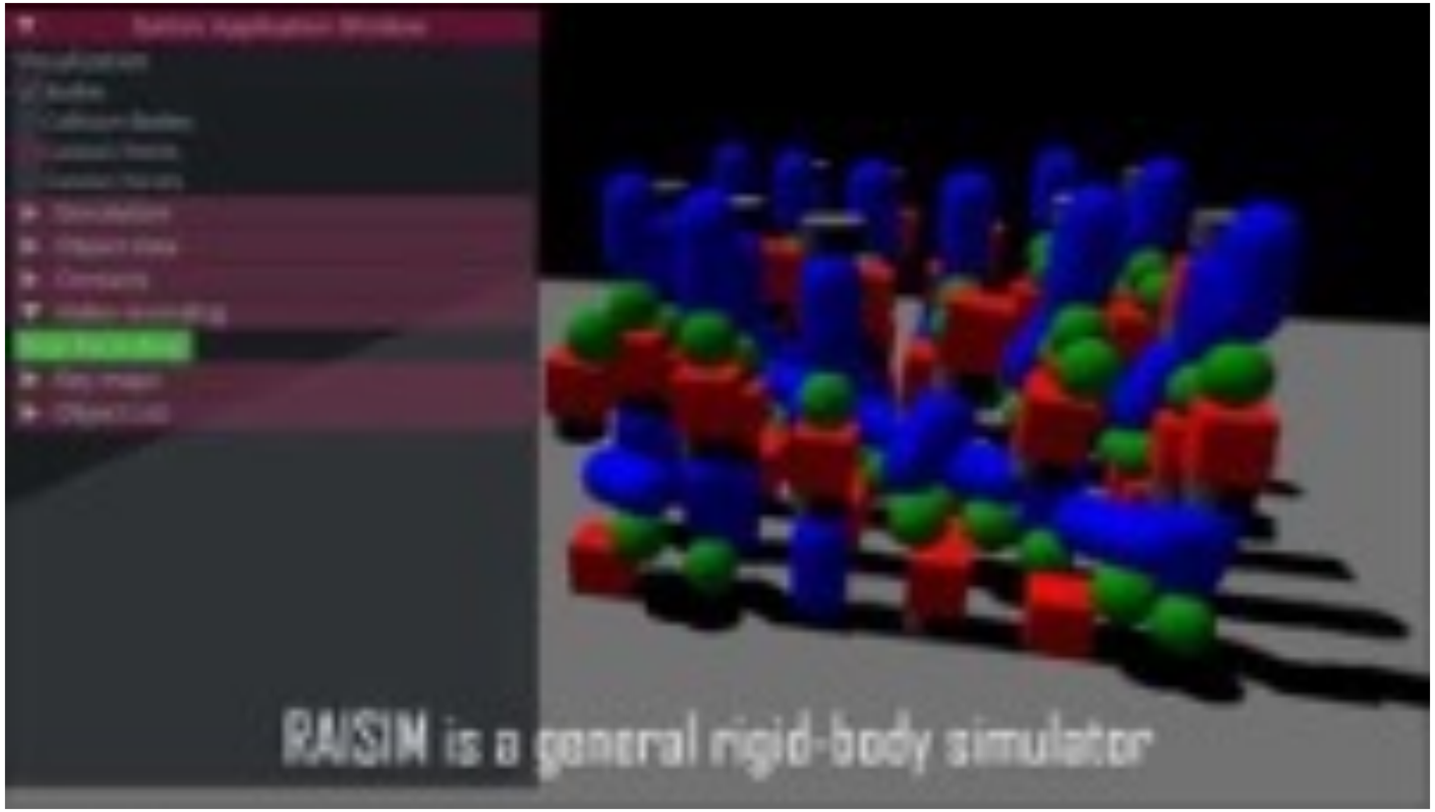
$$\lambda_{imp,z} \geq \mu\sqrt{\lambda_{imp,x}^2 + \lambda_{imp,y}^2}$$

Contact energy minimization:

$$\lambda^* = \underset{\lambda}{\operatorname{argmin}}\frac{1}{2}(\Delta v_{imp} + v_{imp})^T M_{imp}(\Delta v_{imp} + v_{imp})$$

Contact Dynamics – Impact and friction

- **Multi Contact:**



Accuracy vs Computational Efficiency

Constrained EoM → Nonlinear dynamics

Internal Dynamics → Model uncertainty

Contact Dynamics → Hybrid/Combinatorial dynamics

Model-based Optimal Control

Optimal control as an optimization problem.

$$\min_{u(\cdot)} \sum_{k=0}^{T} \gamma^k c(x_k, u_k)$$

s.t. $x_{k+1} = f(x_k, u_k)$ ← Dynamics constraint

$x_k \in X$

$u_k \in U$ for k $= 1, \cdots, T$

$x_0 \in X_0, x_T \in X_T$

Optimization-based approaches

$$\min_{u(\cdot)} \sum_{k=0}^{T} \gamma^k c(x_k, u_k)$$

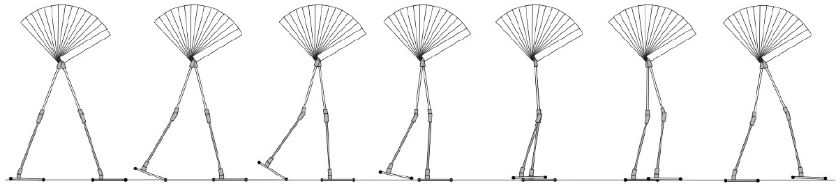
$$\text{s.t.} \quad x_{k+1} = f(x_k, u_k)$$
$$x_k \in X$$
$$u_k \in U \ \text{ for k} = 1, \cdots, T$$
$$x_0 \in X_0, x_T \in X_T$$

- Main benefit
  - Can employ rich set of numerical optimization algorithms to solve optimal control.
- Main caveat
  - The resulting problem is in general nonconvex, and there might be no guarantee of finding globally optimal solution.
  - In general, the optimization problem in its primary form is not computationally tractable.

How to approach finding good solutions?
1. How to deal with hybrid dynamics (mode switches)?

$$\min_{u(\cdot)} \sum_{k=0}^{T} \gamma^k c(x_k, u_k)$$

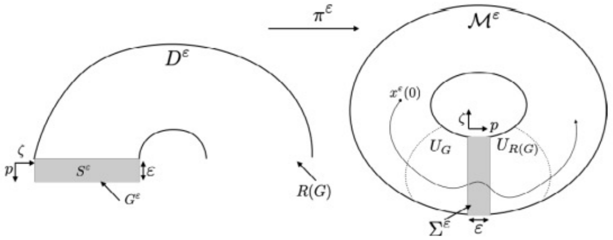
$$\text{s.t.} \quad x_{k+1} = f(x_k, u_k)$$
$$x_k \in X$$
$$u_k \in U \quad \text{for k} = 1, \cdots, T$$
$$x_0 \in X_0, x_T \in X_T$$



1. Separate out mode sequence decision.
2. Use different dynamics representation that captures mode switches implicitly.
   - Complementarity-based formulation[1]

$$\phi(q) \geq 0$$
$$\lambda \geq 0$$
$$\phi(q)^{\mathrm{T}} \lambda = 0.$$

   - Gluing the dynamics in a new topology[2]



1. M. Posa et al., IJRR 2014
2. Westenbroek et al., IFAC 2021

How to approach finding good solutions?
2. Use simplified dynamics

$$\min_{u(\cdot)} \sum_{k=0}^{T} \gamma^k c(x_k, u_k)$$

s.t.

$$x_{k+1} = f(x_k, u_k)$$

$$x_k \in X$$

$$u_k \in U \quad \text{for k} = 1, \cdots, T$$

$$x_0 \in X_0, x_T \in X_T$$

- Linear dynamics:
  - The dynamics constraint becomes linear.
  - Linearization error
  - Fits nicely to Model Predictive Control

How to approach finding good solutions?
2. Use simplified dynamics

$$\min_{u(\cdot)} \sum_{k=0}^{T} \gamma^k c(x_k, u_k)$$

$$\text{s.t.} \quad \boxed{x_{k+1} = f(x_k, u_k)}$$

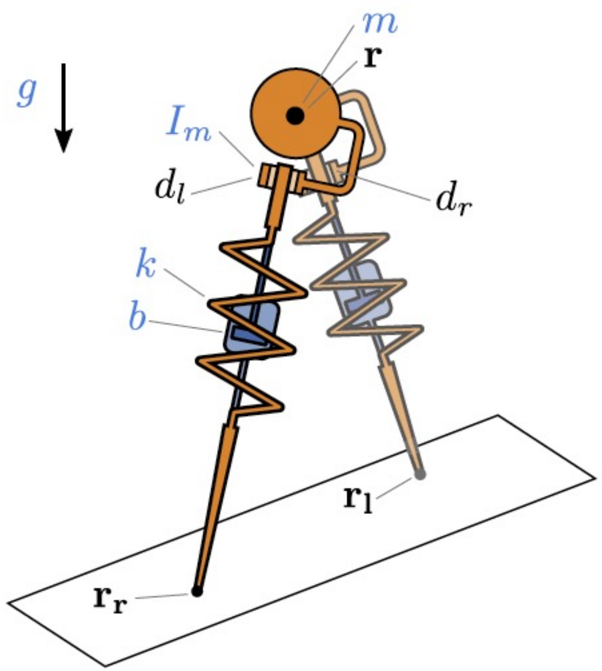
$$x_k \in X$$

$$u_k \in U \quad \text{for k} = 1, \cdots, T$$

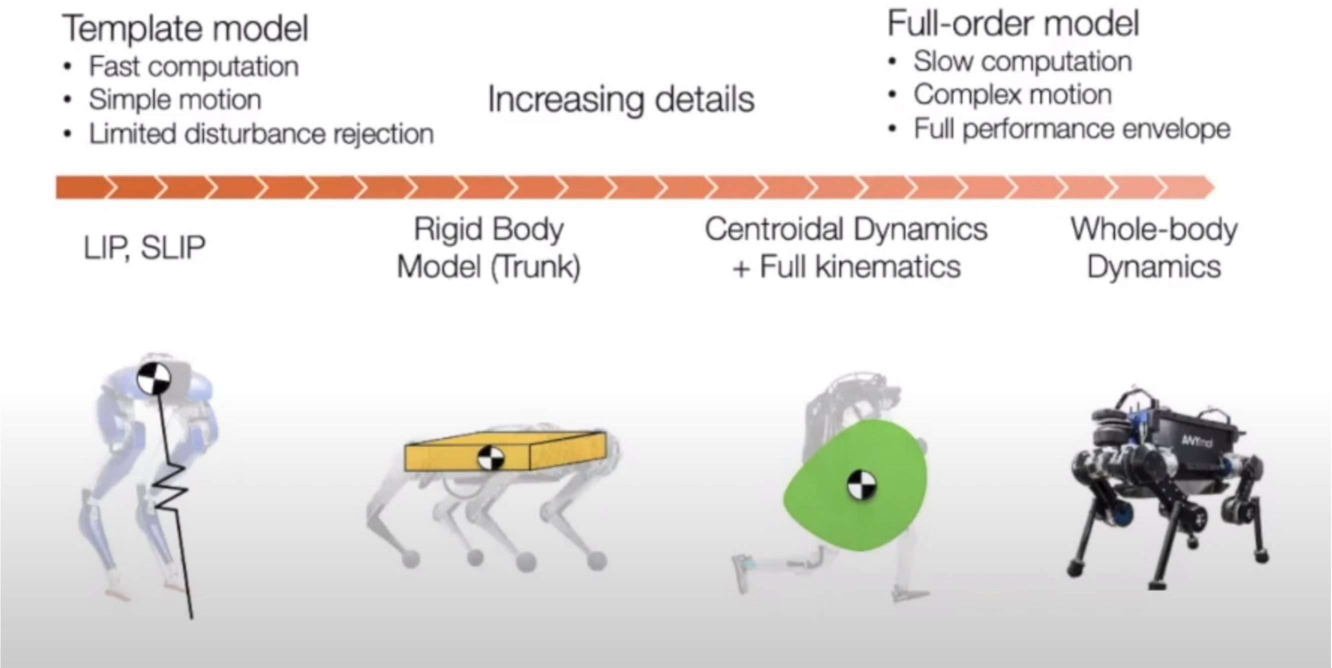
$$x_0 \in X_0, x_T \in X_T$$

- **Use reduced order model:**
  - Reducing the complexity of the problem by simplifying the dynamics and reducing the state variables.
  - The resulting solution might actually not be dynamically feasible.
  - Separate low-level controllers might be necessary to track the solution.

Using reduced order model



K. Green et al., RA-L 2021



P.Wensing, IROS 2022 Workshop

How to approach finding good solutions?
3. Convexify the problem.

$$\min_{u(\cdot)} \sum_{k=0}^{T} \gamma^k c(x_k, u_k)$$

$$\text{s.t.} \quad x_{k+1} = f(x_k, u_k)$$
$$x_k \in X$$
$$u_k \in U \quad \text{for k} = 1, \cdots, T$$
$$x_0 \in X_0, x_T \in X_T$$

- **Lossless convexification**
  - Pontryagin's maximum principle[1]
  - Hopf-lax formula[2]
  - Works only for special cases.
- **Sequential convexification**
  - Differential Dynamic Programming[3]
  - Trust-region-based algorithms[1]
  - Can get stuck at locally optimal solutions.

1. D. Malyuta et al., CSM 2022
2. D. Lee et al., TAC 2022
3. H. Li, P. Wensing, RAL 2021

How to approach finding good solutions?
4. Start with good initial guess.

- A good initial guess can be used to warm-start many numerical optimization algorithms.
- It is helpful to converge to better, if not global, optimal solutions.

Summary of Model-based Optimal Control

$$\min_{u(\cdot)} \sum_{k=0}^{T} \gamma^k c(x_k, u_k)$$

$$\begin{aligned} \text{s.t.} \quad & x_{k+1} = f(x_k, u_k) \\ & x_k \in X \\ & u_k \in U \ \text{ for k} = 1, \cdots, T \\ & x_0 \in X_0, x_T \in X_T \end{aligned}$$

- Explicit dynamics (models) and constraints
- Explicit solutions
  - state trajectory
  - optimal control signal / policy
- Most methods are about how to balance computational tractability & good approximation.
  - Simplified models / good representation of dynamics
  - Convexification
  - Warm-start

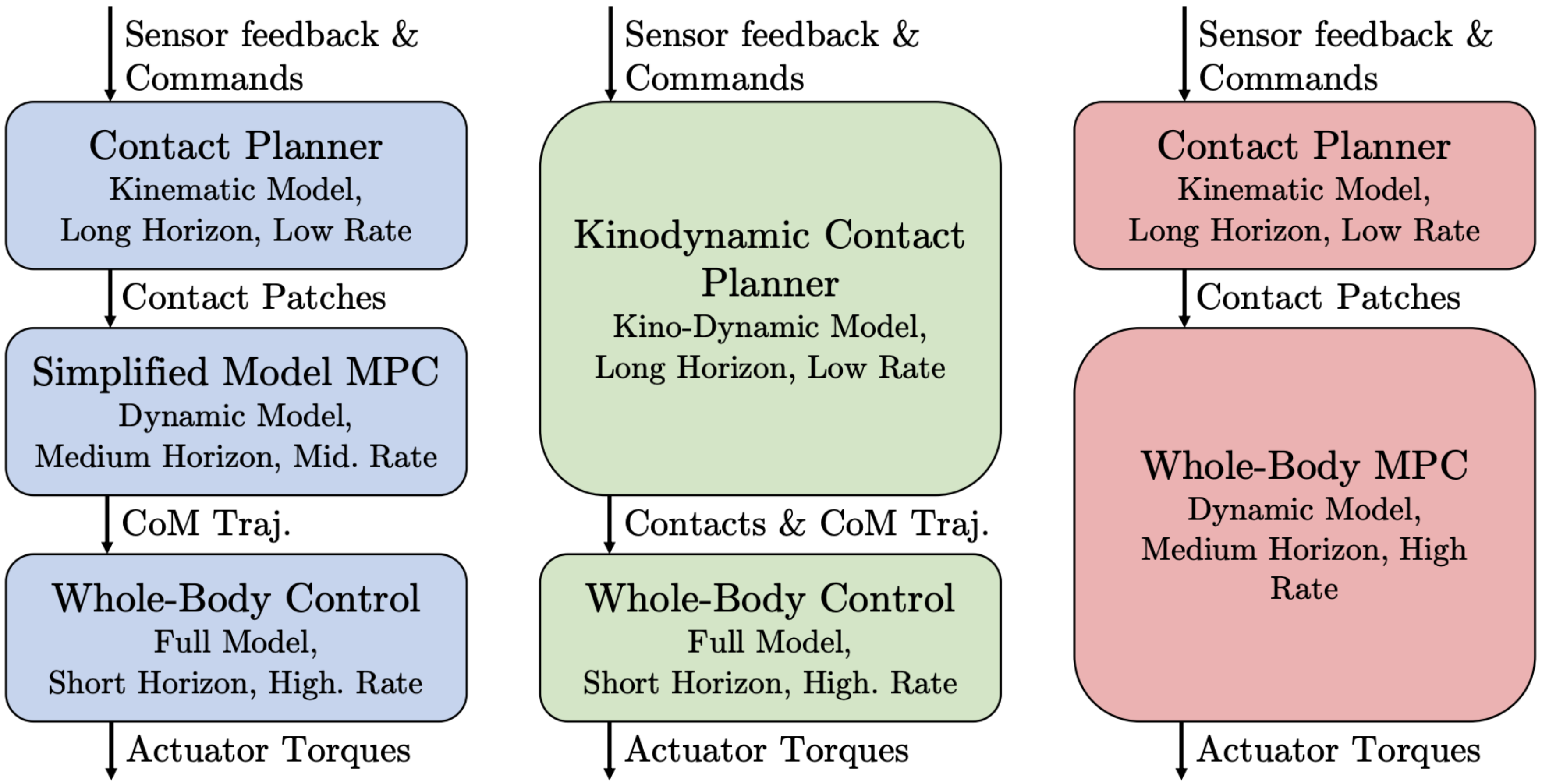
Drawbacks

1. Lack of generalizability: Each system has its own optimization problem. Thus, hard to find generalizable approaches.

2. Lack of robustness: Resulting optimal solutions are often not robust enough. They are good solutions only when the models are good enough.

3. Lack of computational efficiency: Solving the optimization might not be fast enough for the online execution.

Survey paper – P. Wensing et al., Optimization-Based Control for Dynamic Legged Robots



Part 2:
Deep Reinforcement Learning &
Combining Model-based optimal control with Deep RL
(Bike Zhang)