Legged Robots Locomotion Control and Motion Planning

Maren Bennewitz, Shahram Khorshidi
Humanoid Robots Lab, University of Bonn

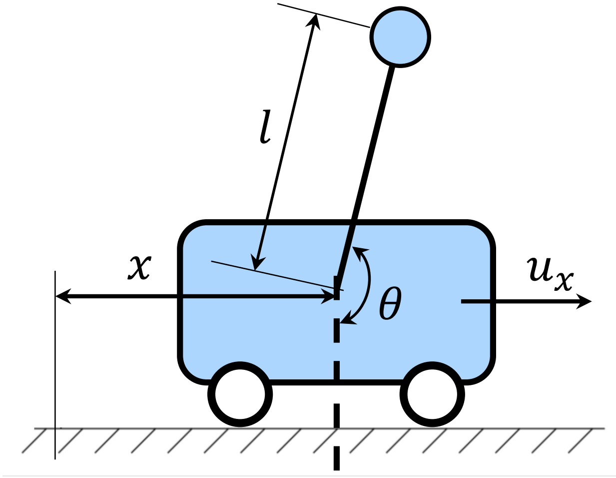
Goals of This Lecture

- Learn how to **linearize** the robot’s dynamics
- Understand the principle of **optimality** in the linear case
- Apply linear control techniques to underactuated robots
- Understand a simple **model** of **humanoids** for walking
- Learn **dynamically stable** control for humanoid walking

Control of Underactuated Robots

Cart-Pole Model

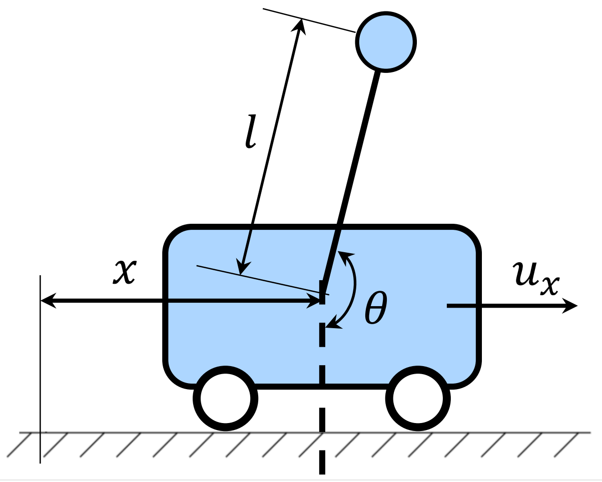
- Cart-pole system is a classic example in control theory and robotics to study and develop control strategies for **underactuated** systems
- It consists of:
  1. A cart that can move horizontally on a track
  2. A rigid pole (pendulum) that can rotate freely in the vertical plane is attached to the cart



- **Goal:** Keep the **pole upright** by moving the cart left and right (**2 DoF** and **only one control input**)

Cart-Pole Model

- Cart-pole has two **equilibriums** (fixed points):
  1. The pole is in the upright position (unstable)
  2. The pole hangs straight down (stable)

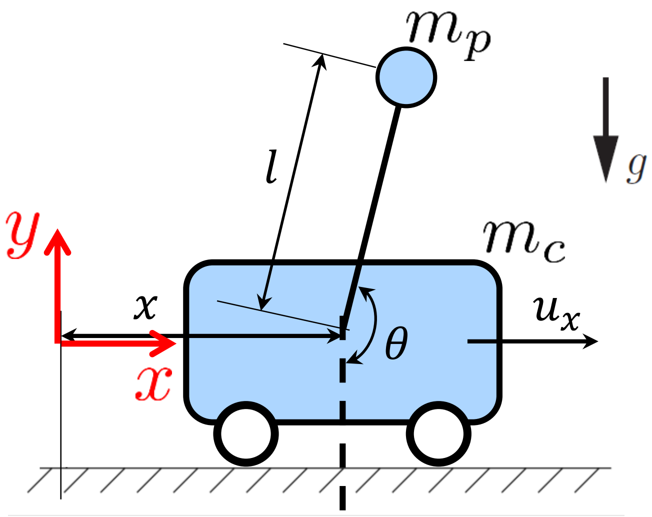


- **Fixed point** definition:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}^*,\, \boldsymbol{u}^*) = 0$$

- **Goal**: Keep the pendulum in the upright position, $\theta = \pi$, by applying force to the cart u_x

Cart-Pole Model



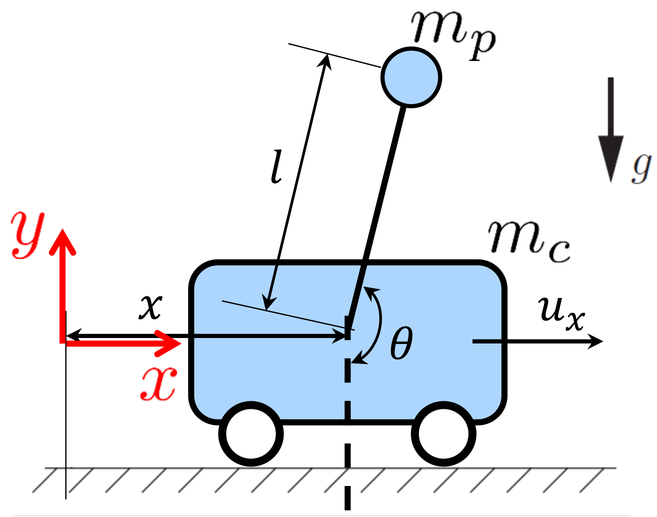
- Generalized coordinates

$$\boldsymbol{q} = \begin{pmatrix} x \\ \theta \end{pmatrix}, \qquad \dot{\boldsymbol{q}} = \begin{pmatrix} \dot{x} \\ \dot{\theta} \end{pmatrix}, \qquad \ddot{\boldsymbol{q}} = \begin{pmatrix} \ddot{x} \\ \ddot{\theta} \end{pmatrix}$$

- Control input: force on the cart

$$\boldsymbol{u} = \begin{pmatrix} u_x \\ 0 \end{pmatrix}$$

Cart-Pole Model



- **Kinematics** of the system in the Cartesian coordinate system

- Position of the cart

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} x \\ 0 \end{bmatrix}$$

- Position of the pole

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} x + lsin(\theta) \\ -lcos(\theta) \end{bmatrix}$$

Cart-Pole Model

- **Kinetic** and **potential** energy of the system [*]

$$
\begin{aligned}
T =& \frac{1}{2}(m_c + m_p)\dot{x}^2 + m_p \dot{x}\dot{\theta} l \cos\theta + \frac{1}{2} m_p l^2 \dot{\theta}^2 \\
U =& - m_p g l \cos\theta.
\end{aligned}
$$

- Use the **Lagrangian**, write the equation in general form

$$\mathcal{L}(\theta, \dot{\theta}) = \sum_{i=1}^{2} (\mathcal{K}_i - \mathcal{P}_i) \longrightarrow \tau_i = \frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{\theta}_i} - \frac{\partial \mathcal{L}}{\partial \theta_i}, \qquad i = 1, 2$$

[*] Herbert Goldstein, “Classical mechanics,” Addison-Wesley Press, 1950.

Cart-Pole Model

- **Underactuated** system with **nonlinear** dynamics [*]

$$\boxed{M(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \boldsymbol{u}}$$

- With:

$$M(\boldsymbol{q}) = \begin{bmatrix} m_c + m_p & m_p l cos(\theta) \\ m_p l cos(\theta) & m_p l^2 \end{bmatrix}$$

forces due to inertia

$$\boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \begin{bmatrix} -m_p l \dot{\theta}^2 sin(\theta) \\ 0 \end{bmatrix}$$

forces due to rotation

$$\boldsymbol{g}(\boldsymbol{q}) = \begin{bmatrix} 0 \\ -m_p g l sin(\theta) \end{bmatrix}$$

forces due to gravity

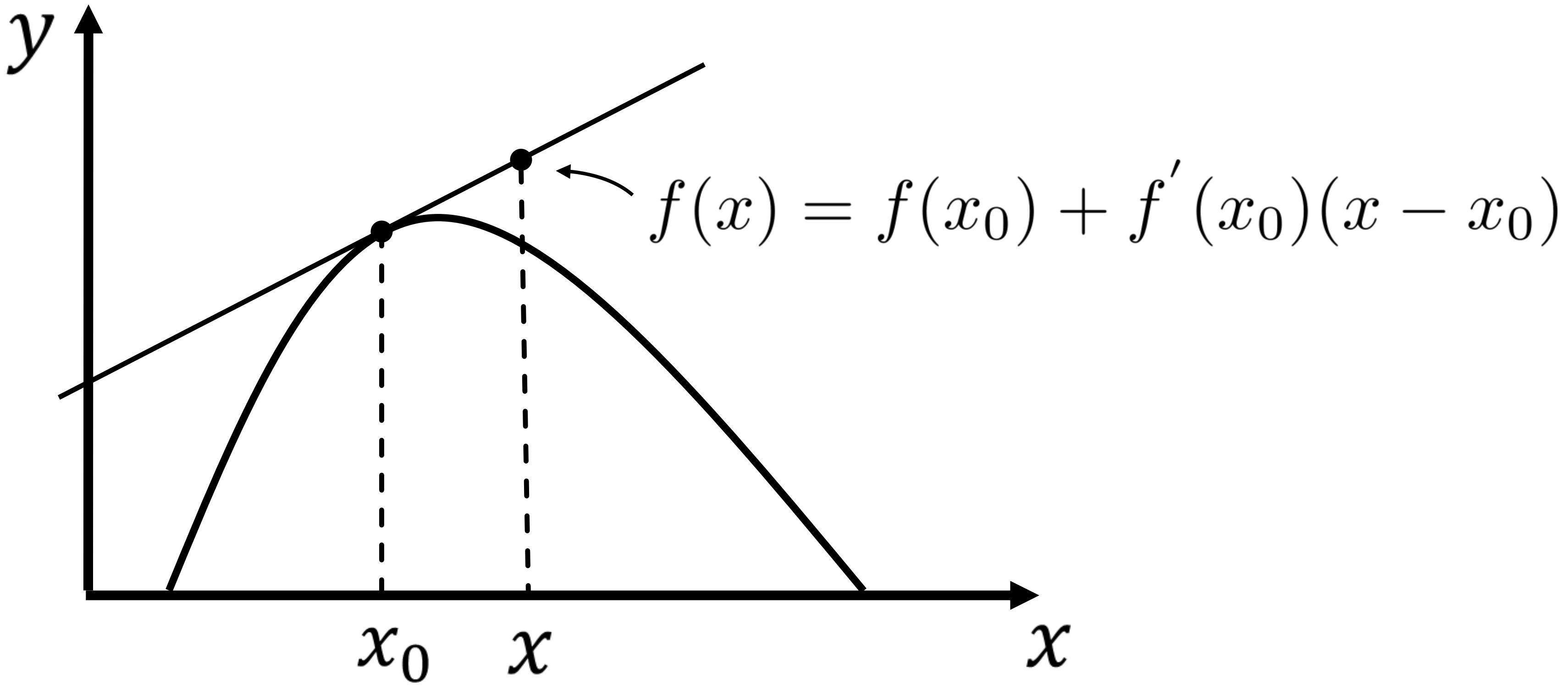
[*] Russ Tedrake, "Underactuated robotics," Course notes for MIT 6.832, 2023.

Cart-Pole Model

- Actual dynamics are **nonlinear**
- **Near** an **equilibrium point** (e.g., upright position), the system behaves almost linearly
- This linear model captures the **local behavior** of the system accurately enough for control purposes
- Linear control techniques (e.g., **LQR**) are **well-studied, efficient,** and **easier to implement** than nonlinear controllers

Taylor Expansion

- The first order Taylor series of a function $f(x)$ in x_0 is a **linear approximation** that is tangential in x_0



Linearization of Dynamics - Balancing

- **Linearize** the nonlinear equations about a **fixed point**

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \approx \mathbf{f}(\mathbf{x}^*, \mathbf{u}^*) + \left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right]_{\mathbf{x}=\mathbf{x}^*,\, \mathbf{u}=\mathbf{u}^*} (\mathbf{x} - \mathbf{x}^*) + \left[\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right]_{\mathbf{x}=\mathbf{x}^*,\, \mathbf{u}=\mathbf{u}^*} (\mathbf{u} - \mathbf{u}^*)$$

- By defining a new coordinate system

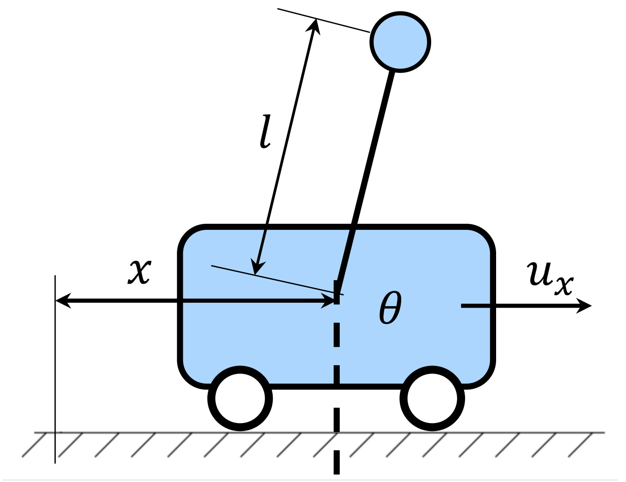
$$\bar{\mathbf{x}} = \mathbf{x} - \mathbf{x}^*,\ \bar{\mathbf{u}} = \mathbf{u} - \mathbf{u}^*$$

- We get **linear dynamical** system

$$\dot{\bar{\mathbf{x}}} = A\bar{\mathbf{x}} + B\bar{\mathbf{u}}$$

Linear Quadratic Regulator (LQR)

- **Motivation**: the goal is to keep the pole upright by moving the cart left and right



- **Challenges**:
  - Underactuated dynamics
  - Trade-off in control
    1. **Accuracy**: how well we balance the pole (minimizing deviation from the upright position)
    2. **Control effort**: how much force we apply to the cart

LQR as an Optimal Solution

- LQR systematically finds the **optimal** control policy

- What do we mean by optimal?

- **Trade-off** between:
  1. Minimizing the deviation of the pole from the upright position
  2. Minimizing the amount of force applied to the cart

LQR Formulation

- **Linear time-invariant** system in state-space form

$$\boxed{\dot{x}(t) = Ax(t) + Bu(t)}$$

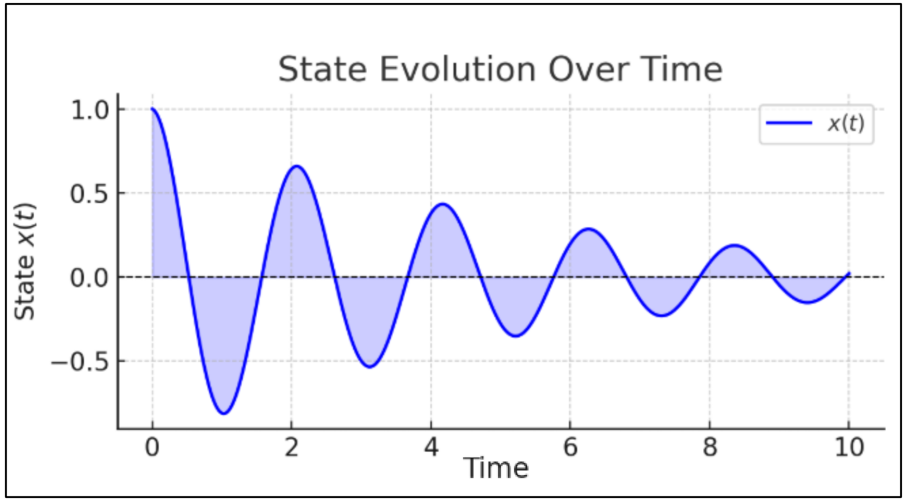
- **Minimize** a quadratic cost function

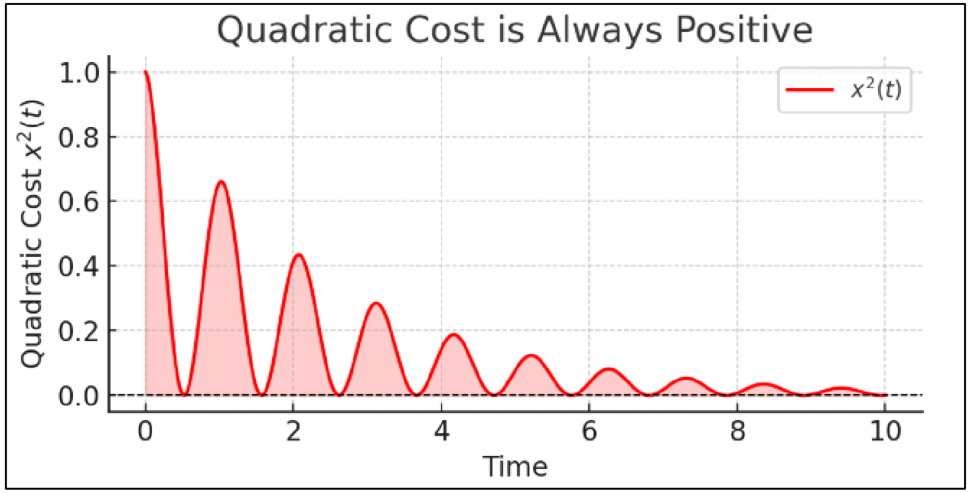
$$\boxed{J = \int_0^\infty \left(x^T Q x + u^T R u\right) dt} \qquad Q = Q^T,\ Q \succeq 0 \qquad R = R^T,\ R \succ 0$$

- Q and R are **cost matrices**

LQR Formulation

- Why quadratic terms?
- Q penalizes the pole's deviation from vertical and cart displacement





- R penalizes the force applied (control effort)

LQR Formulation

- How to **choose** Q and R, and how do they affect the LQR **performance**?

- $\boldsymbol{Q}$ is usually **positive semi-definite** and $\boldsymbol{R}$ must be **positive definite**

- Most common form is positive diagonal matrices

- Q_{ii} penalize the **relative errors** in state variable x_i

- R_{ii} penalize **control effort** u_i

LQR Formulation

- LQR finds an **optimal feedback** control law [*]

$$\boxed{u = -Kx}$$

- *K* is the feedback gain matrix, and is calculated by solving the **Riccati equation**

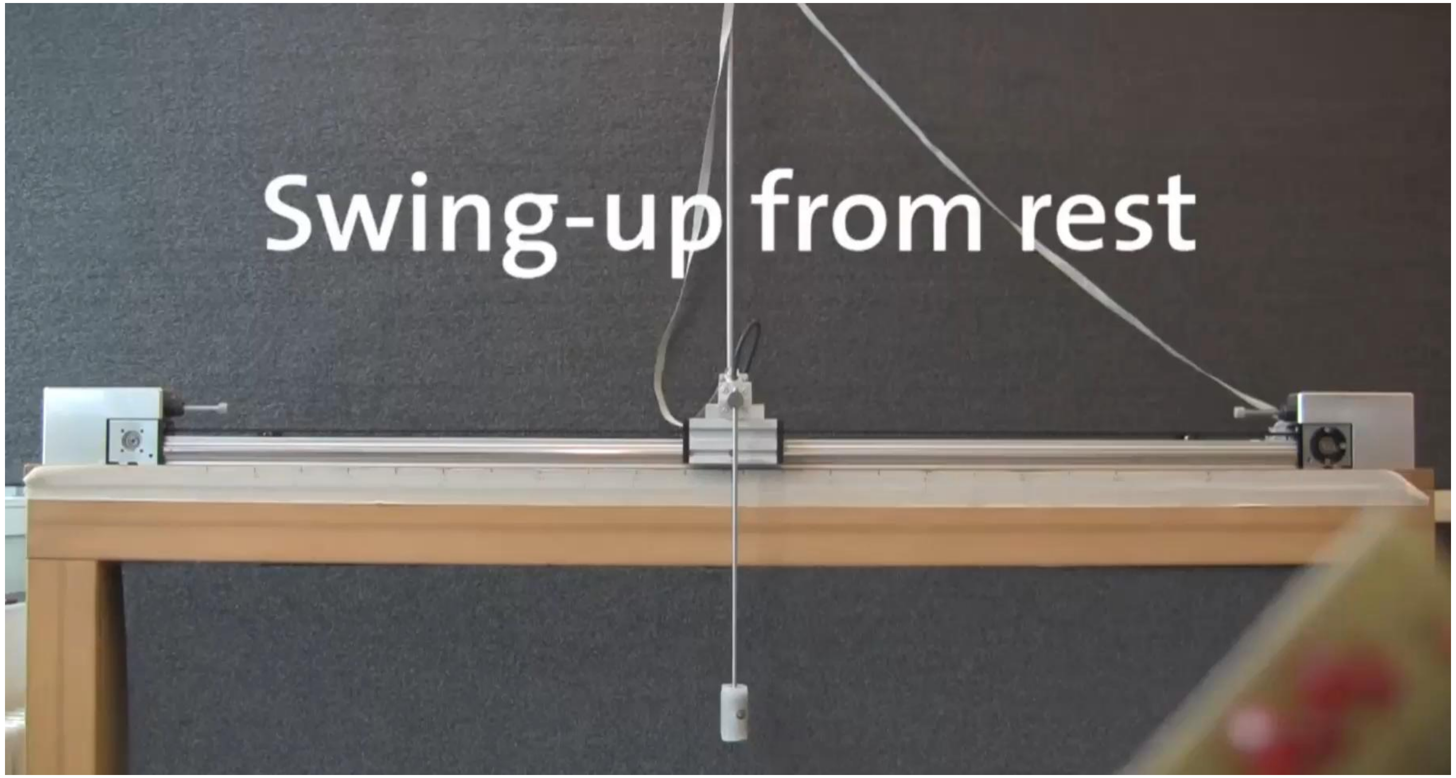
- LQR is a **feedback controller**, it naturally reacts to disturbances and model uncertainties

[*] Anderson, Brian D.O., John B. Moore. "Optimal control: linear quadratic methods," Prentice Hall, 1990.

LQR Limitations

- LQR assumes **linear** dynamics
- Requires **full-state feedback**
- Assumes an **infinite time** horizon
- Basic LQR does **not** explicitly handle state or input **constraints**

LQR on Cart-Pole



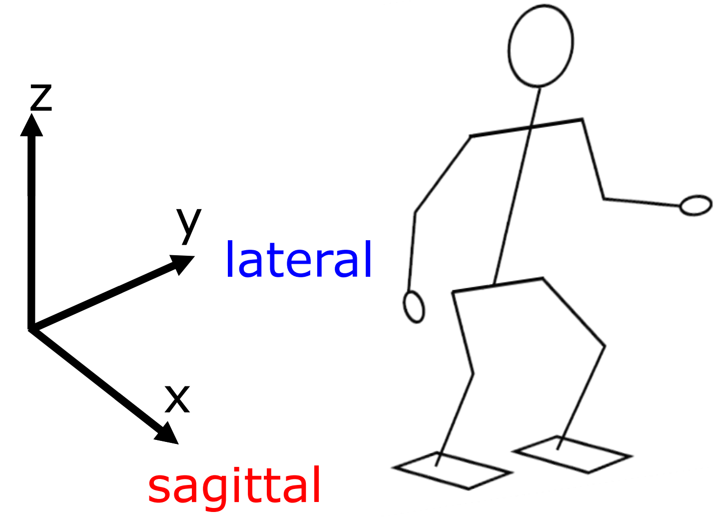
YouTube: Cart-Pole Swing-Up Experiments using
Simulation-Based LQR-Trees

Legged Robots Modeling

Legged Robot Motion

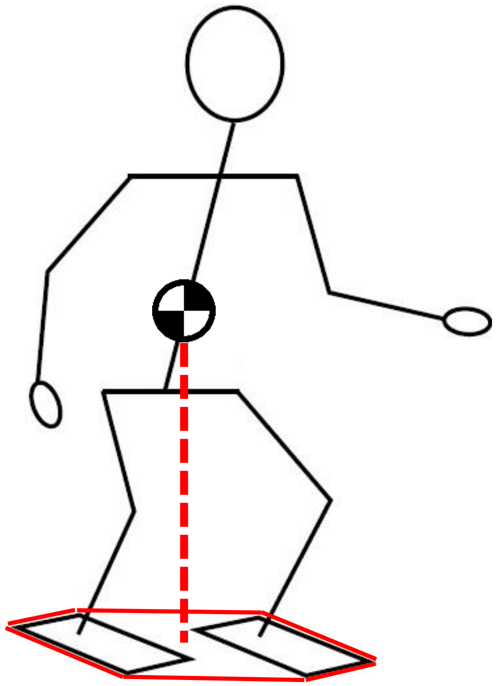
- To control complex systems like humanoids, first derive a simplified model that captures essential balance dynamics and are amenable to linear control methods

- Walking motion can be decomposed into orthogonal projections in the **sagittal** (forward) and **lateral** (sideways) directions



Posture Stability

- A pose is **statically stable** if the **vertical projection** of the robot's Center of Mass (CoM) lies within the support polygon on the floor
- **Support polygon**: convex hull of all contact points of the robot on the floor

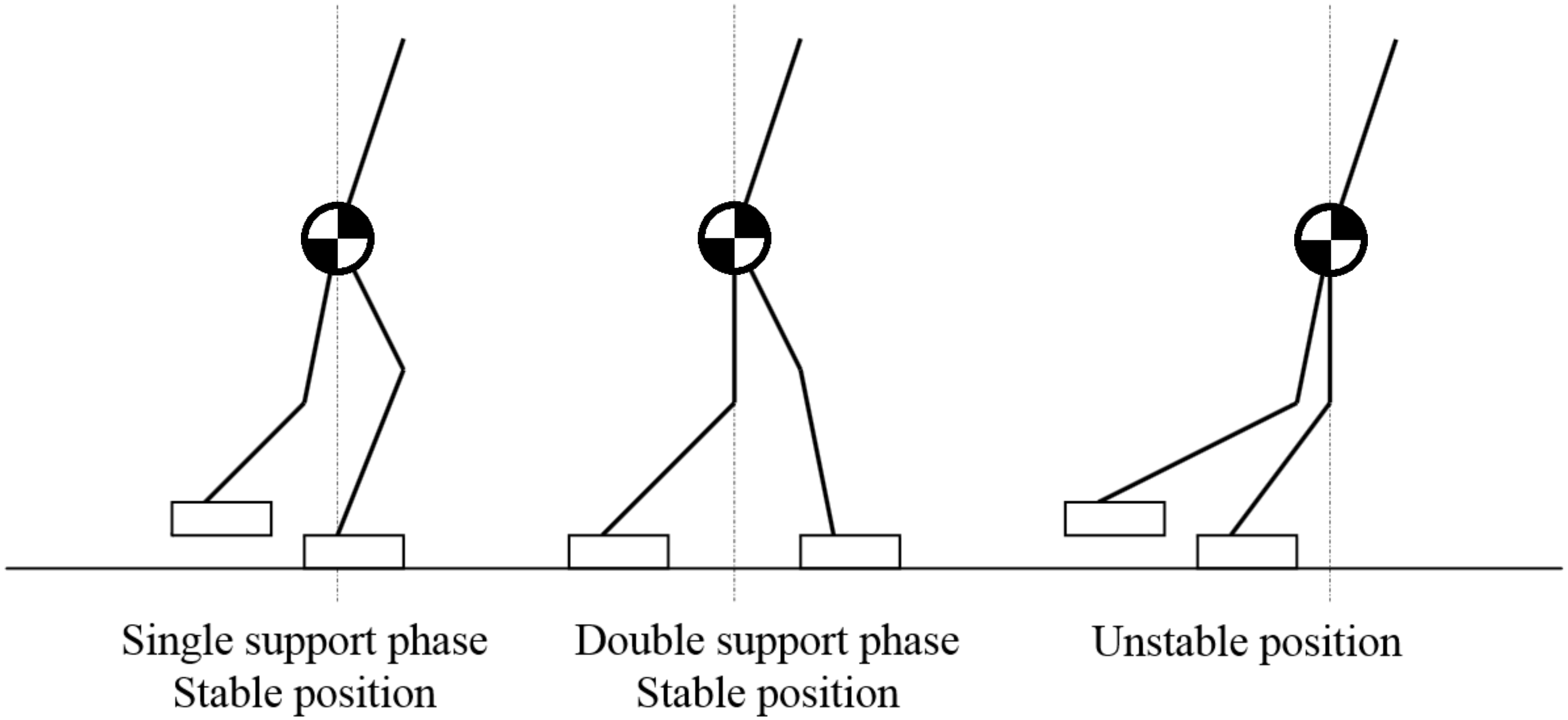


Statically Stable Walking

- The robot will stay in a stable pose whenever the motion is stopped
- **At any time**, the projection of the robot's COM on the ground must be contained within the support polygon
- Support polygon:
  1. Either the foot surface in case of one supporting leg, or
  2. The minimum convex area containing both foot surfaces when both feet are on the ground

Statically Stable Walking

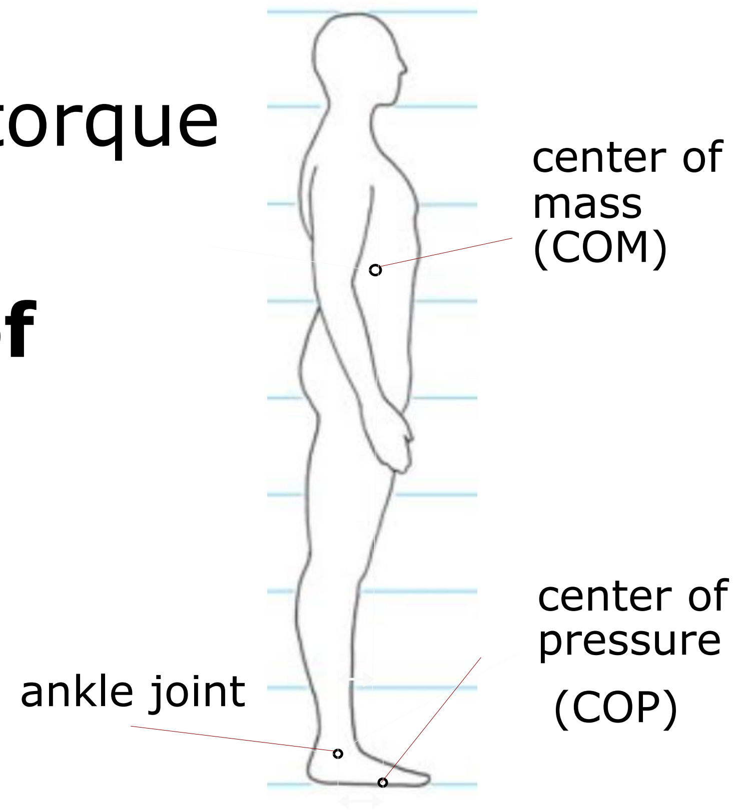
- Leads to robust but **slow** walking performance



source: T. Asfour

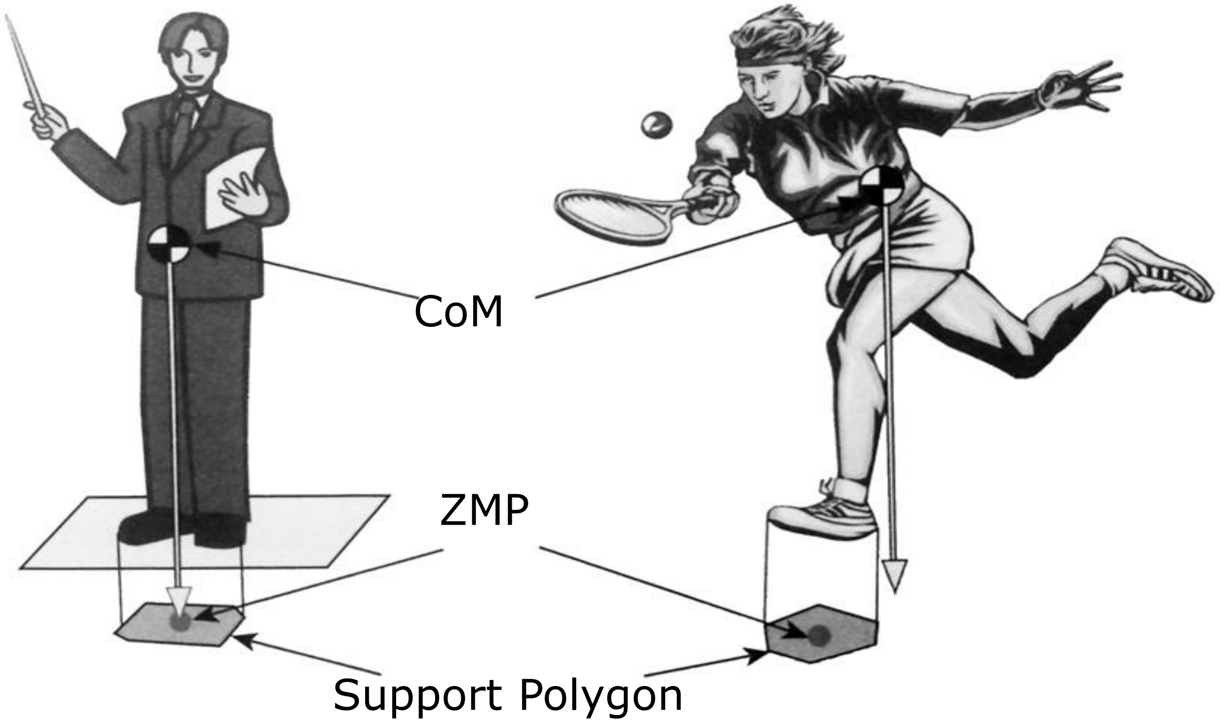
Human Walking

- Body as an inverted pendulum pivoting around the ankle joint
- Represented as a point mass
- Body weight around at the COM creates a torque about the ankle, leading to motion
- Ground reaction force acts at the **Center of Pressure (CoP)**



source: T. Asfour

Dynamically Stable Walking

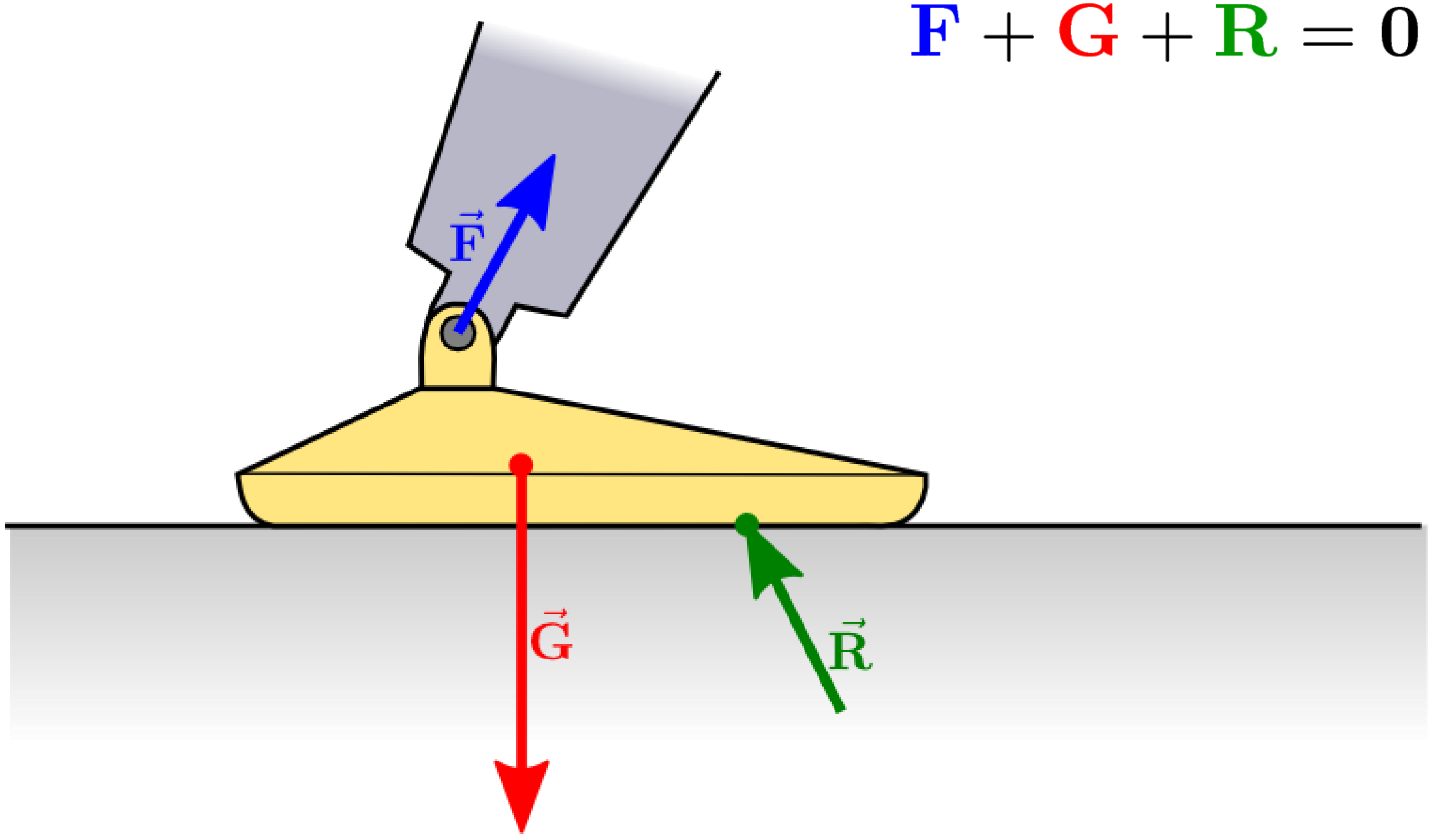


In dynamic walking, stopping the motion may result in falling

source: S. Kajita

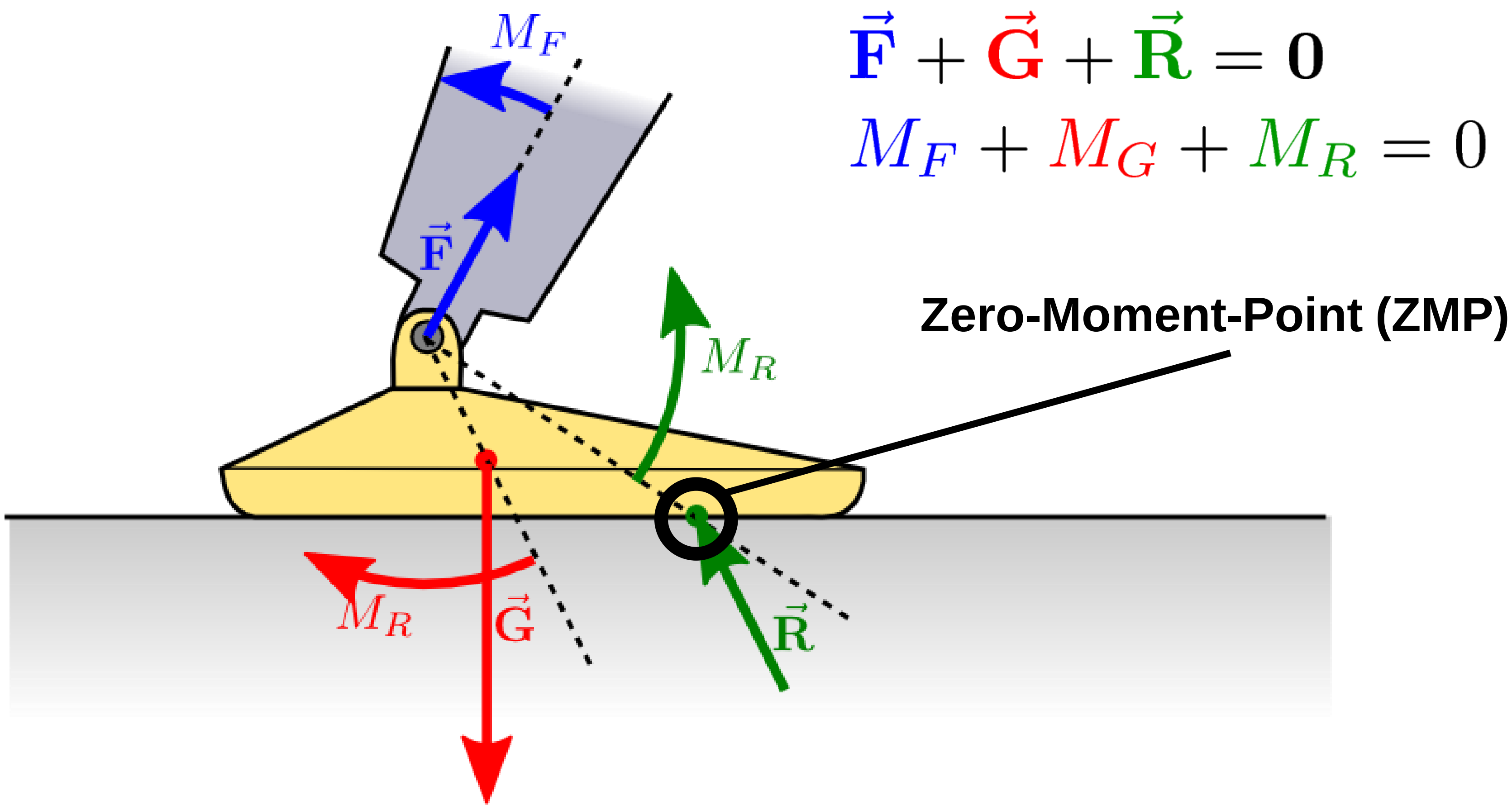
Zero Moment Point

$$\vec{\mathbf{F}} + \vec{\mathbf{G}} + \vec{\mathbf{R}} = \mathbf{0}$$



Zero Moment Point

$$\vec{\mathbf{F}} + \vec{\mathbf{G}} + \vec{\mathbf{R}} = \mathbf{0}$$
$$M_F + M_G + M_R = 0$$



Ground Reaction Force

- The ground acts on the whole contact area, but it can be substituted by a single force acting at the CoP

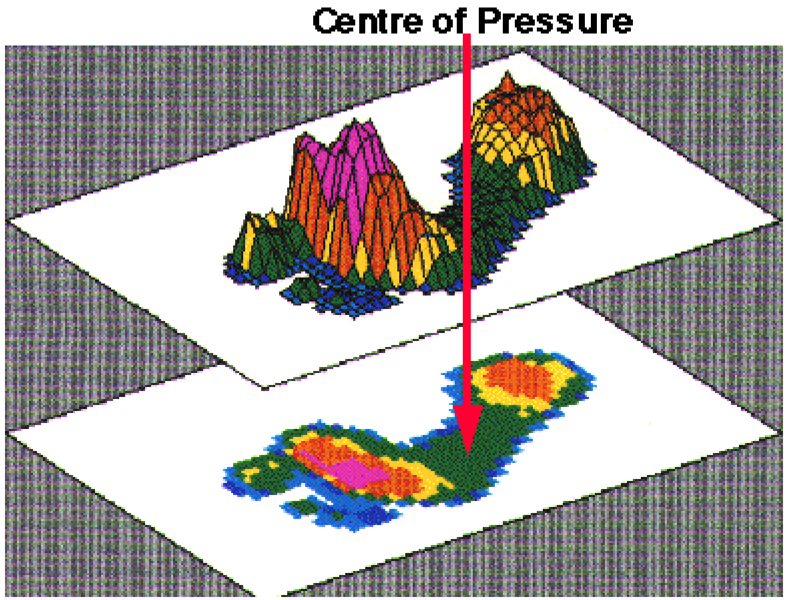


image source: clinical gait analysis

Zero Moment Point

- Stability is achieved if the **Zero Moment Point (ZMP)** is in the support area
- A robot standing on the ground applies a force and moment to the ground
- At the same time, the ground applies a force and a moment to the robot (ground reaction force)
- ZMP is the **point on the ground** where the **total moment generated due to gravity and inertia equals zero**

Zero Moment Point

- For stable walking, the support foot must rest on the ground
- Forces and torques acting on support foot must sum up to 0
- ZMP must remain **inside the footprint** of the support foot
- Then, the **ZMP coincides with the CoP**
- ZMP should not approach the edge of the support polygon, as this increases the risk of instability
- **During the movement**, the projection of the **CoM can leave the support polygon**

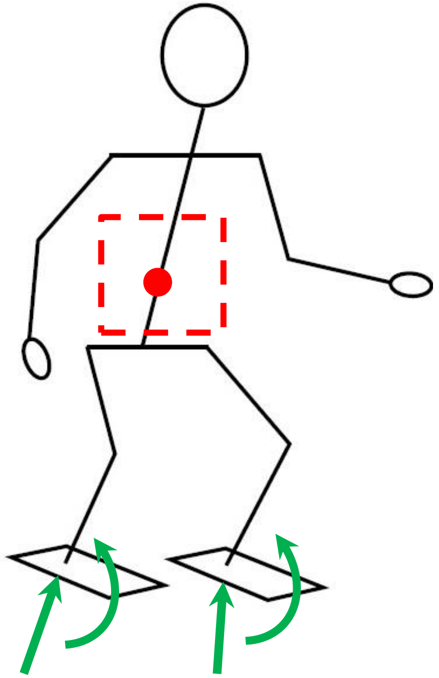
Legged Robots Model

- General form of EoM

$$M(\boldsymbol{q})\dot{\boldsymbol{v}} + \boldsymbol{c}(\boldsymbol{q}, \boldsymbol{v}) + \boldsymbol{g}(\boldsymbol{q}) = \boxed{S^\top} \boldsymbol{\tau} + \sum_i J_i^T \boxed{\boldsymbol{f}_i}$$

- The base does not have any actuation

$$S = \begin{bmatrix} 0_{n_j \times 6} & I_{n_j \times n_j} \end{bmatrix}$$



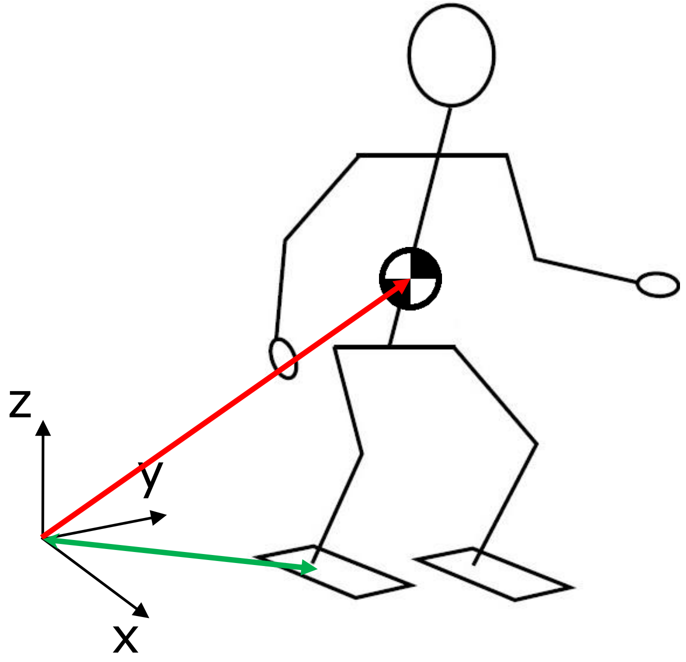
Robot's Center of Mass (CoM)

- **Newton** equation for CoM

$$\boxed{m\left(\ddot{\mathbf{c}}+\mathbf{g}\right)=\sum_i \boldsymbol{f}_i}$$

- **Euler** equation for the angular momentum

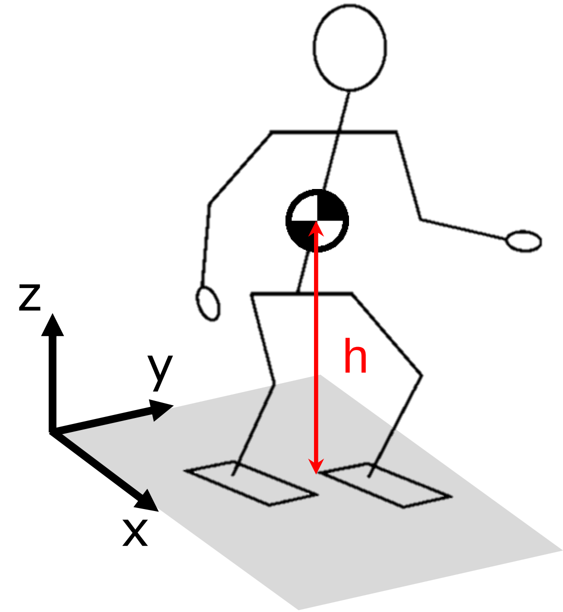
$$\boxed{\dot{\mathbf{L}}=\sum_i \left(\boxed{\boldsymbol{p}_i}-\boxed{\boldsymbol{c}}\right)\times \boldsymbol{f}_i}$$



- Where $\boldsymbol{p}_i$ is the contact location

In Contact with a Flat Ground

- Consider a reference frame oriented along the ground, with the z-axis orthogonal to it
- **Assume** the robot base is not tilted
  $g_x = 0,\ g_y = 0,\ g_z = g$
- **Assume** that the height of COM stays constant during walking $\ddot{c}_z = 0$
- **Suppose** that for $\boldsymbol{p}_i$, points of contact with the ground, $p_i^z = 0$ and $\mathrm{f} = [0,\ 0,\ f_i^z]^T$
- With the **assumption** that the angular momentum is constant for walking, $\dot{L} = 0$



How ZMP/CoP Appear

- With the assumptions of in contact with a flat ground

$$c_x - \frac{c_z}{g}\ddot{c}_x = z_x$$

- c_x: CoM position in the x-direction
- $\ddot{c}_x$: CoM acceleration in the x-direction

- With CoP definition as

$$z_x = \frac{\sum_i f_i^z p_i^x}{\sum_i f_i^z}$$

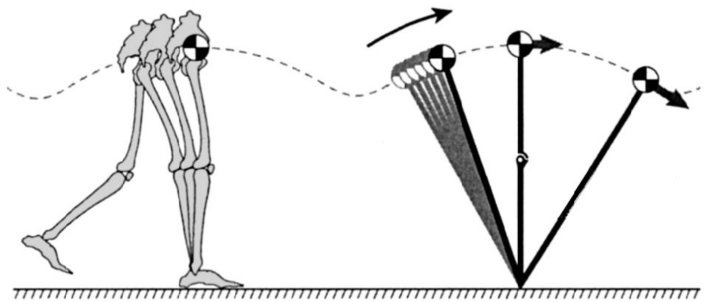
- f_i^z: the vertical force at contact point
- p_i^x: the x-coordinate of contact point

- For full derivation look at [*]

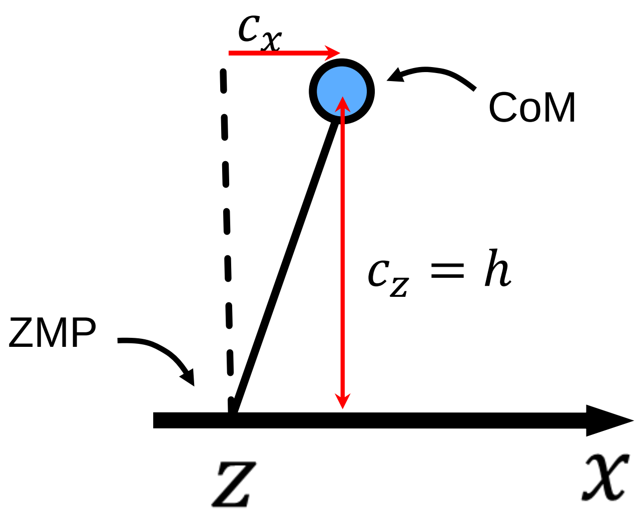
[*] Pierre-Brice Wieber, Russ Tedrake, Scott Kuindersma, "Modeling and control of legged robots," Springer Handbook of Robotics, 2016.

Dynamics of Linear Inverted Pendulum

- Finally, we get simplified dynamics for the robot motion, known as Linear Inverted Pendulum Model (**LIPM**)



Source: Omar et.al.: "Study of Bipedal Robot Walking Motion in Low Gravity: Investigation and Analysis," International Journal of Advanced Robotic Systems, 2014.



- CoM equation:

$$\ddot{c}_x = \frac{g}{h}(c_x - z_x)$$

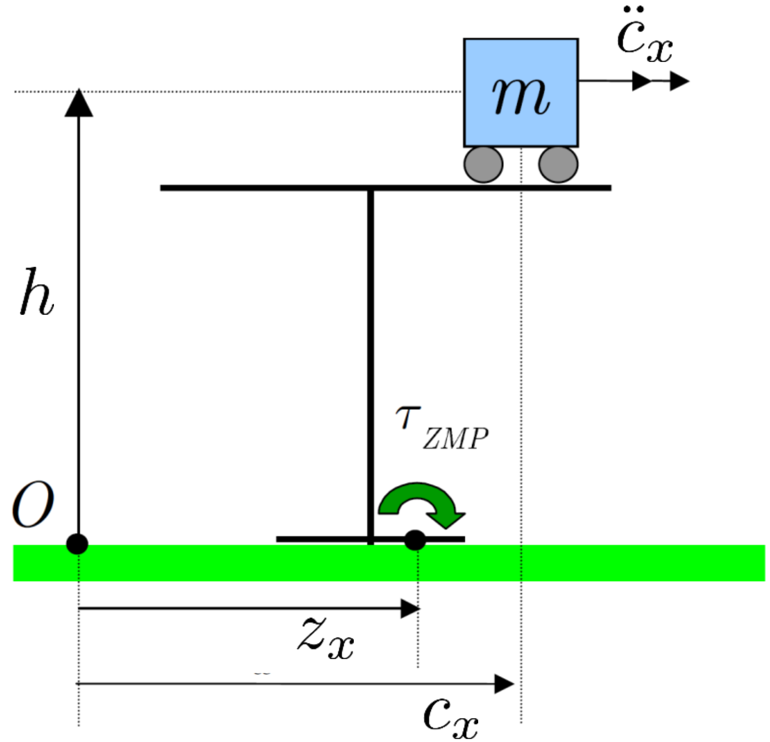
- ZMP equation:

$$z_x = \frac{\sum_i f_i^z p_i^x}{\sum_i f_i^z}$$

A Cart-on-Pedestal Model for ZMP Intuition

- The foot of the table is **too small** to let the cart stay in balance
- If the cart **accelerates** with a proper rate, the table can **keep upright for a while**

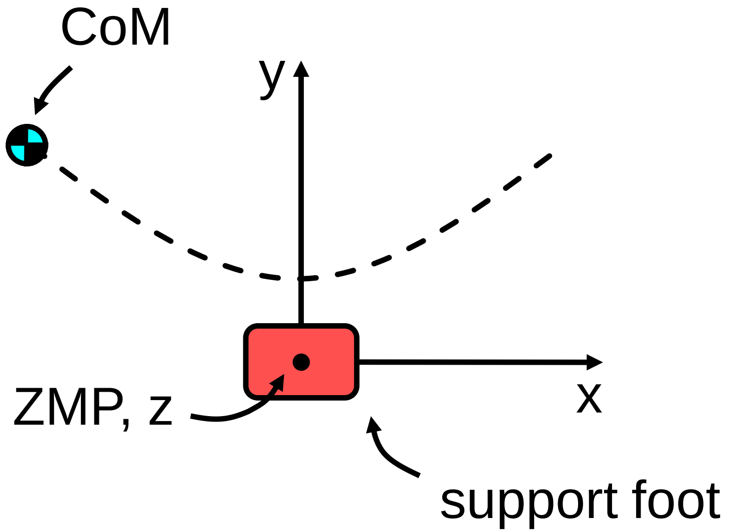
$$mg(c_x - z_x) - m\ddot{c}_x h = 0$$



Source: Kajita et al., "Biped walking pattern generation by using preview control of zero-moment point," ICRA, 2003.

2D Linear Inverted Pendulum Model

$$\boldsymbol{c} = \begin{bmatrix} c_x \\ \dot{c}_x \\ c_y \\ \dot{c}_y \end{bmatrix}$$



- x-direction:

$$\ddot{c}_x = \frac{g}{h}(c_x - z_x)$$

- y-direction :

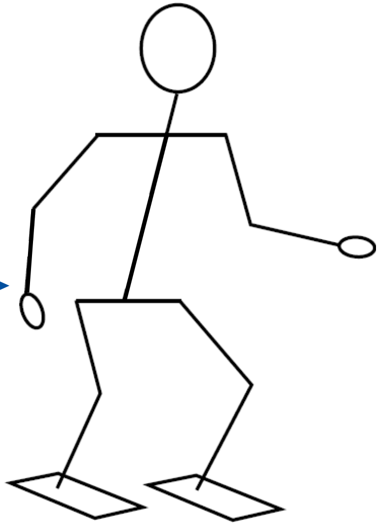
$$\ddot{c}_y = \frac{g}{h}(c_y - z_y)$$

Legged Robots Motion Planning

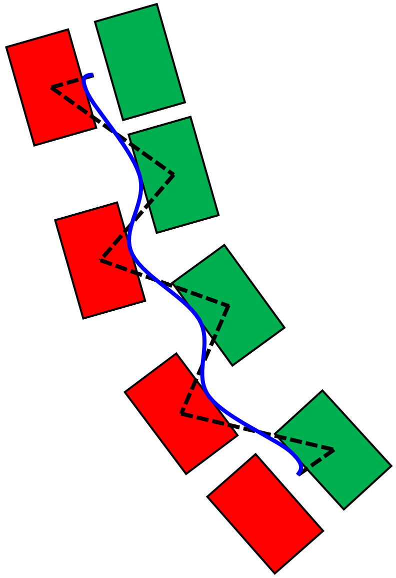
Overview

- **Decomposition** approach

Footsteps → Desired ZMP Trajectory → CoM Trajectory → Joint Commands →



This Lecture



ZMP-Based Walking Pattern Generator

- Stable walking requires **contact forces**, which are strictly constrained by physics
- On flat ground, the **ZMP** (**CoP**) must stay within the **convex hull** of the foot contact points
- Key walking parameters:
  1. COM height
  2. Step duration (single/double support)
  3. Step speed
- **Foot trajectories** are often predefined using **polynomial curves** with zero velocity and acceleration at the start and end of each step

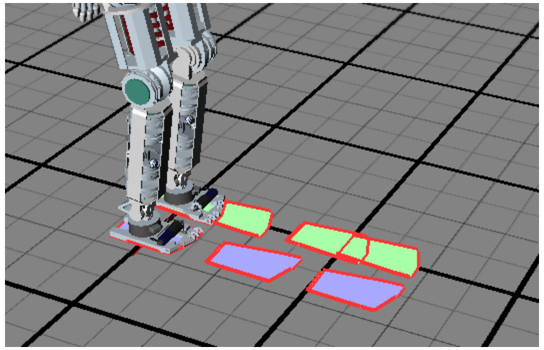
ZMP Preview Control – Key Idea

- **ZMP preview control** computes a **CoM trajectory** given:
  1. **Fixed sequence of footsteps**
  2. **Reference ZMP trajectory**
- Assumptions:
  1. Footsteps are fixed and cannot be changed during execution
  2. COM height remains constant throughout the motion
- Main constraint: **Resulting ZMP trajectory** must always stay inside the support polygon

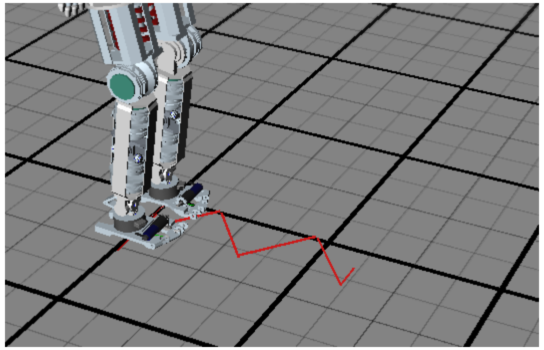
Reference ZMP, CoM Trajectory Generation

- Reference ZMP is defined based on **support phase**:
  1. In single support: located at the center of the foot
  2. In double support: quick transition from the previous foot to the next
- Using the LIPM, compute the CoM trajectory that follows the reference ZMP
- Once CoM and feet positions are known, compute joint angles via inverse kinematics

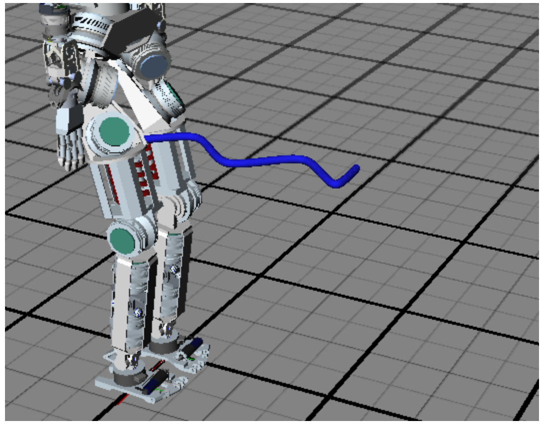
ZMP-Based Walking Pattern Generator



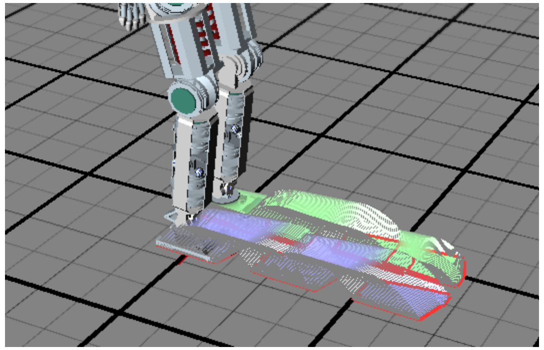
footstep positions



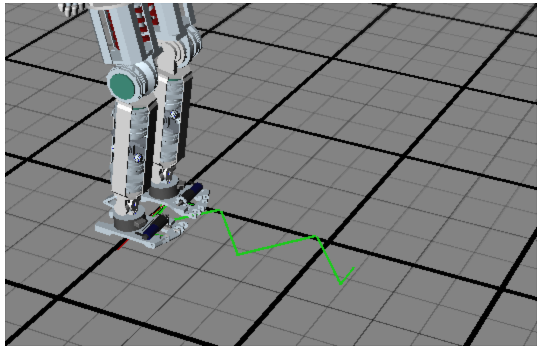
reference ZMP



CoM trajectory



feet trajectory



resulting ZMP

source: T. Asfour

ZMP Preview Control

- Model Predictive Control (**MPC,** aka receding horizon control)
- Solves a sequence of optimal control problems **online** to generate motion for **constrained** dynamical systems
- Enables **real-time adaptation** to changing conditions

ZMP Preview Control

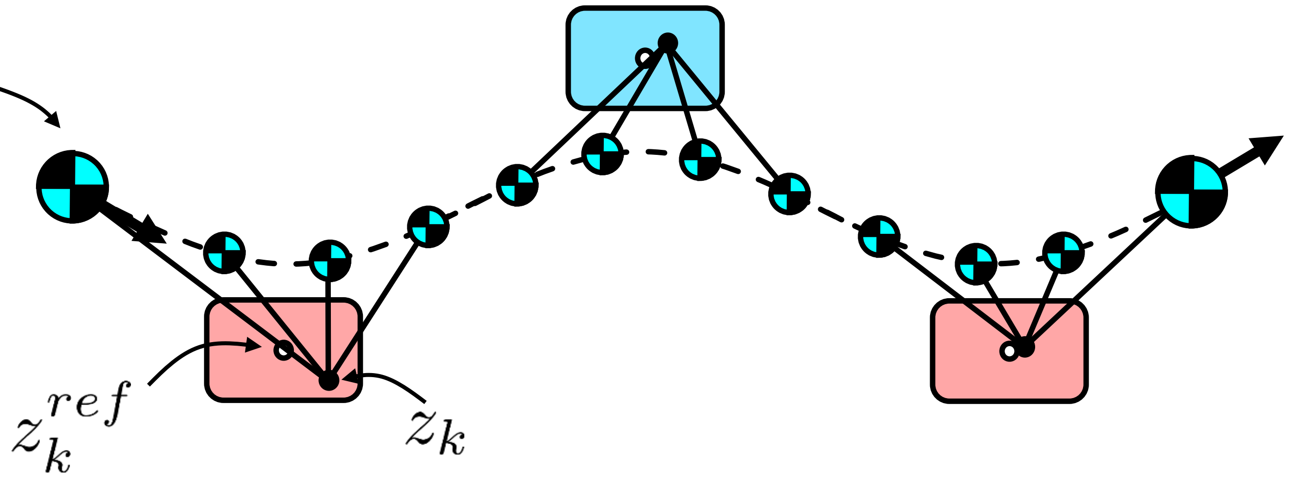
- ZMP reference is given (from fixed footsteps)
- Use **LIPM** dynamics to predict CoM motion
- **Goal:** Compute **CoM trajectory** such that the resulting ZMP stays inside the support polygon
- At each time step, MPC uses Quadratic Programming (**QP**) to compute optimal CoM accelerations (or jerks) that track the reference ZMP [*]

[*] Pierre-Brice Wieber, "Trajectory Free Linear Model Predictive Control for Stable Walking in the Presence of Strong Perturbations," Humanoids, 2006

ZMP Preview Control

- The robot's state (CoM position, velocity, acceleration)
- The reference ZMP trajectory z_k^{ref}
- The actual (predicted) ZMP z_k based on current/future CoM motion
- **Goal**: Track z_k^{ref} while ensuring dynamic feasibility (LIPM)

$$\boldsymbol{c} = \begin{bmatrix} c_x \\ \dot{c}_x \\ \ddot{c}_x \\ c_y \\ \dot{c}_y \\ \ddot{c}_y \end{bmatrix}$$



ZMP Preview Control

- Let's only consider the motion along the **x-direction** (sampled with time step *T*)

- State: $\boldsymbol{x}_k = \begin{bmatrix} c_k \\ \dot{c}_k \\ \ddot{c}_k \end{bmatrix}$, where $\boldsymbol{x}_k = \boldsymbol{x}(kT)$

- Input: jerk $u_k = \dddot{c}_k$

- Discrete-time dynamics: $\boldsymbol{x}_{k+1} = A\boldsymbol{x}_k + Bu_k$

ZMP Preview Control

- ZMP **reference** trajectory: from footsteps (fixed in advance), a piecewise reference ZMP trajectory is computed:
  1. In single support: center of foot
  2. In double support: fast linear transition from one foot to the next

- ZMP **output** equation

$$z_k = c_k - \frac{h}{g}\ddot{c}_k = D\boldsymbol{x}_k$$

ZMP Preview Control

- At every control step, solve a **QP** to **minimize** the cost over a time horizon *N*

- The MPC **cost**: $J = \sum_{i=0}^{N-1} \|z_{k+i} - z_{ref,k+i}\|_Q^2 + \|u_{k+i}\|_R^2$

- **ZMP tracking** is penalized
- **Smoothness** of motion via jerk minimization

- **Constraints**: optional bounds on ZMP (stability)

Limitations of ZMP

- ZMP works effectively only on **flat surfaces**
- CoM has to move on a **fixed plane**, not possible to run, jump, and climb stairs, without modification
- Not capable of dealing with external forces (e.g., leaning on a wall)
- ZMP has to remain in the support polygon for all time instances

Summary

- Control of humanoid walking
- **Linearize** nonlinear dynamics with Taylor expansion
- Principle of **optimality** in linear case
- Linear Quadratic Regulator (**LQR**) for underactuated robots
- Modeling a **humanoid** with simple **Linear Inverted Pendulum Model** (LIPM)
- **ZMP-based** control for LIPM
- **Motion planning** for ZMP and CoM trajectory for humanoid walking

Literature

- M. W. Spong, "Underactuated mechanical systems," Control Problems in Robotics and Automation, 1997.
- Russ Tedrake, "Underactuated robotics (algorithms for walking, running, swimming, flying, and manipulation)," Course notes for MIT 6.832, 2023.
- J.P. Hespanha, “Linear system theory,” Princeton University Press, 2018.
- Anderson, Brian D.O., John B. Moore. “Optimal control: linear quadratic methods,” Prentice Hall, 1990.
- Pierre-Brice Wieber, Russ Tedrake, Scott Kuindersma, “Modeling and control of legged robots,” Springer Handbook of Robotics, 2016.
- Kajita et al., “Biped walking pattern generation by using preview control of zero-moment point,” IEEE International Conference on Robotics and Automation, 2003.
- Pierre-Brice Wieber, “Trajectory Free Linear Model Predictive Control for Stable Walking in the Presence of Strong Perturbations,” Humanoids, 2006.
- Herdt et al., “Online Walking Motion Generation with Automatic Foot Step Placement,” Advanced Robotics, 2010.