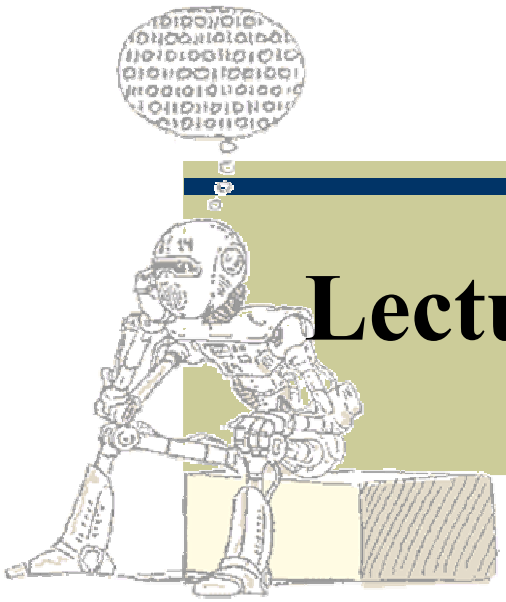


Lecture XII– Trajectory Planning (I)

Contents:

- Imitation based
  - Open vs closed loop
  - Hand tuned trajectories
  - ZMP control
  - Virtual Model Control
- Optimization based trajectory plans
  - Minimum Torque Change
  - Minimum Jerk
  - Minimum End-Point Variance

Trajectory Planning: Simple methods

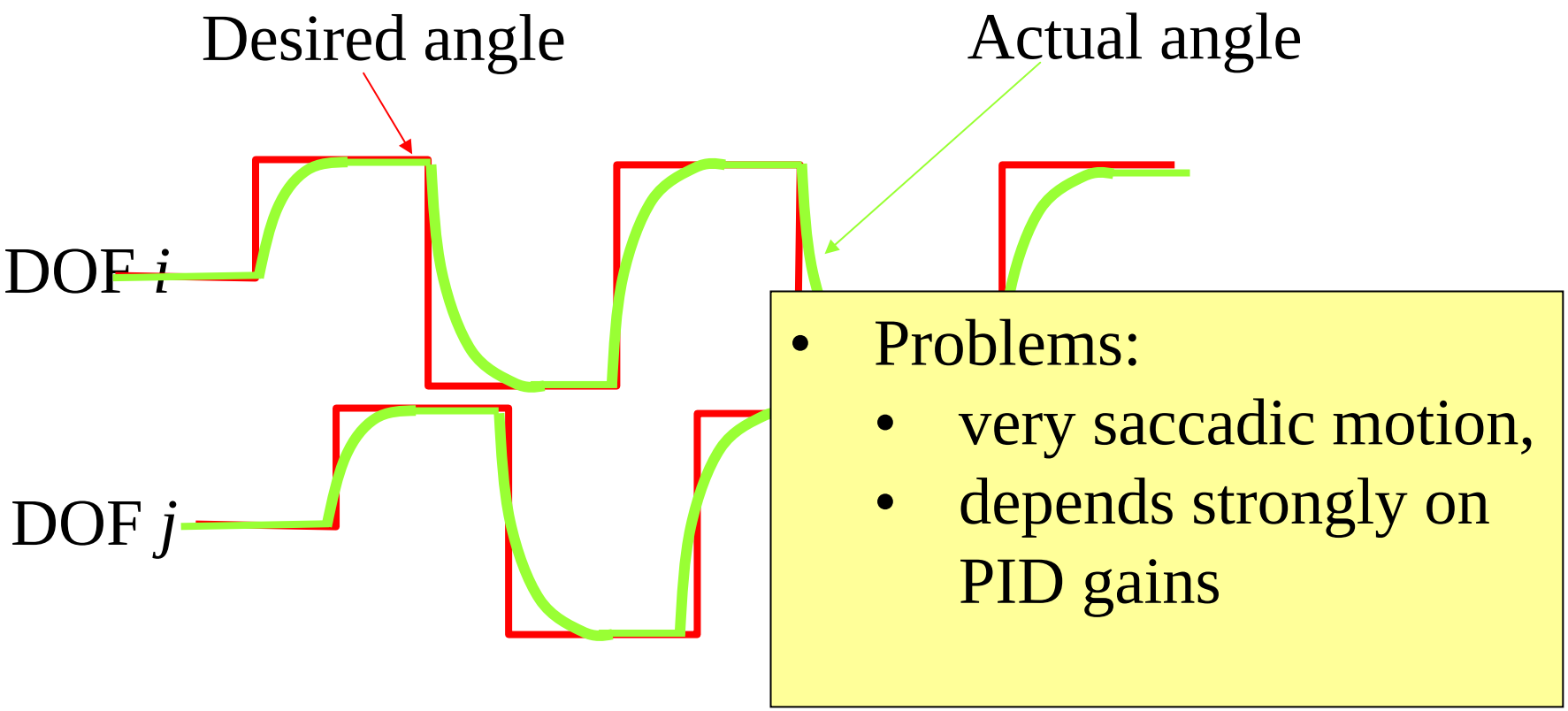
- Question: how to generate good trajectories?

- (Very) simple options:
  - Provide a few desired positions (angles) over time (controller with step functions)
  - Provide smooth hand-tuned trajectories (e.g. with spline fitting)
  - Use a sinusoidal controller

- These are ***open-loop*** solutions, i.e. no feedback to the trajectory planner

Controller with step function

- Simply provide a new desired angle every so often:



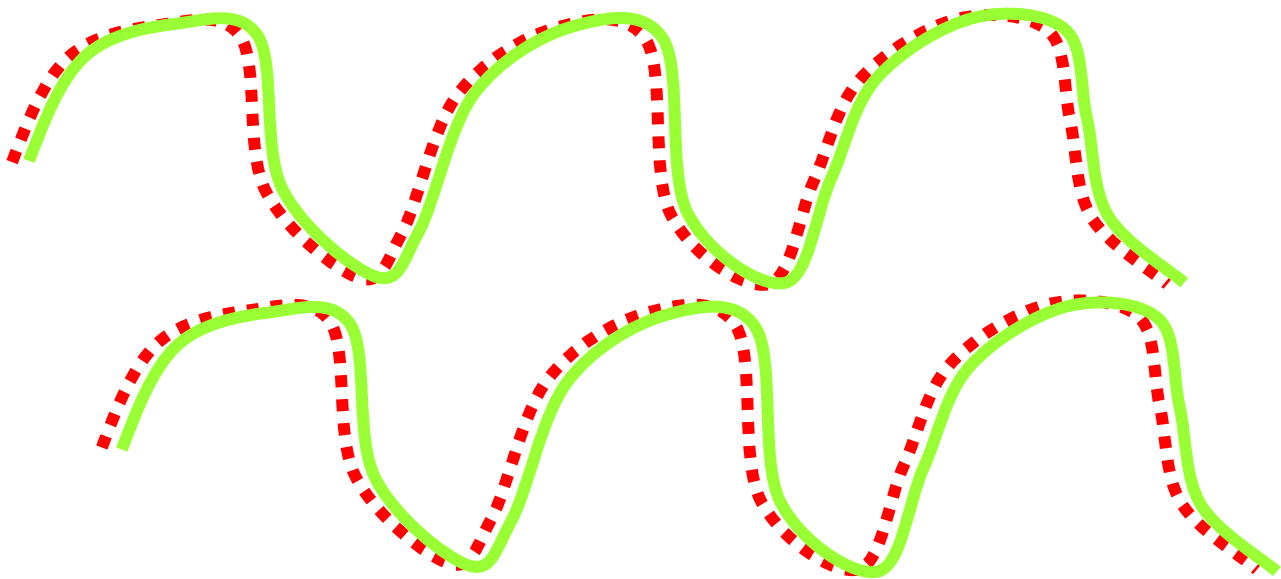
- Problems:
  - very saccadic motion,
  - depends strongly on PID gains

Hand Tuned Trajectories

- Provide a trajectory as a vector:

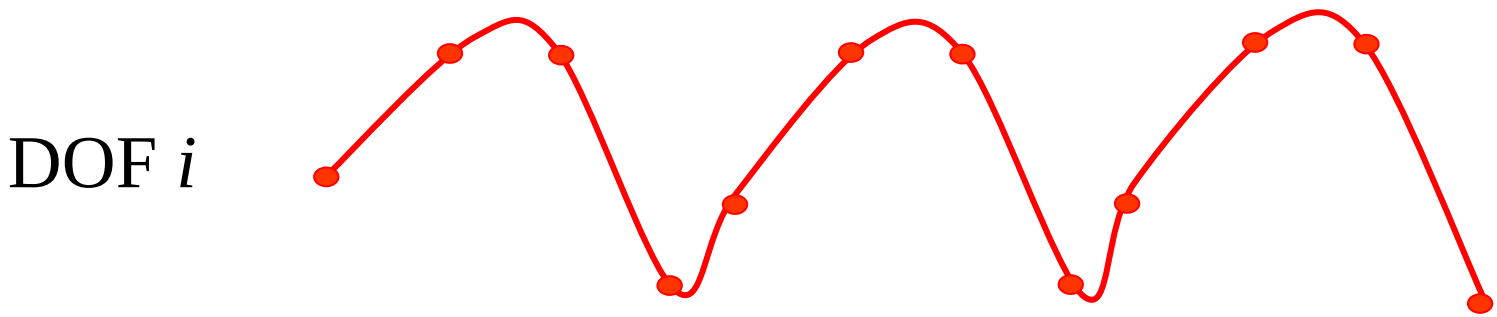
DOF *i*

DOF *j*



Controller with hand tuned trajectories

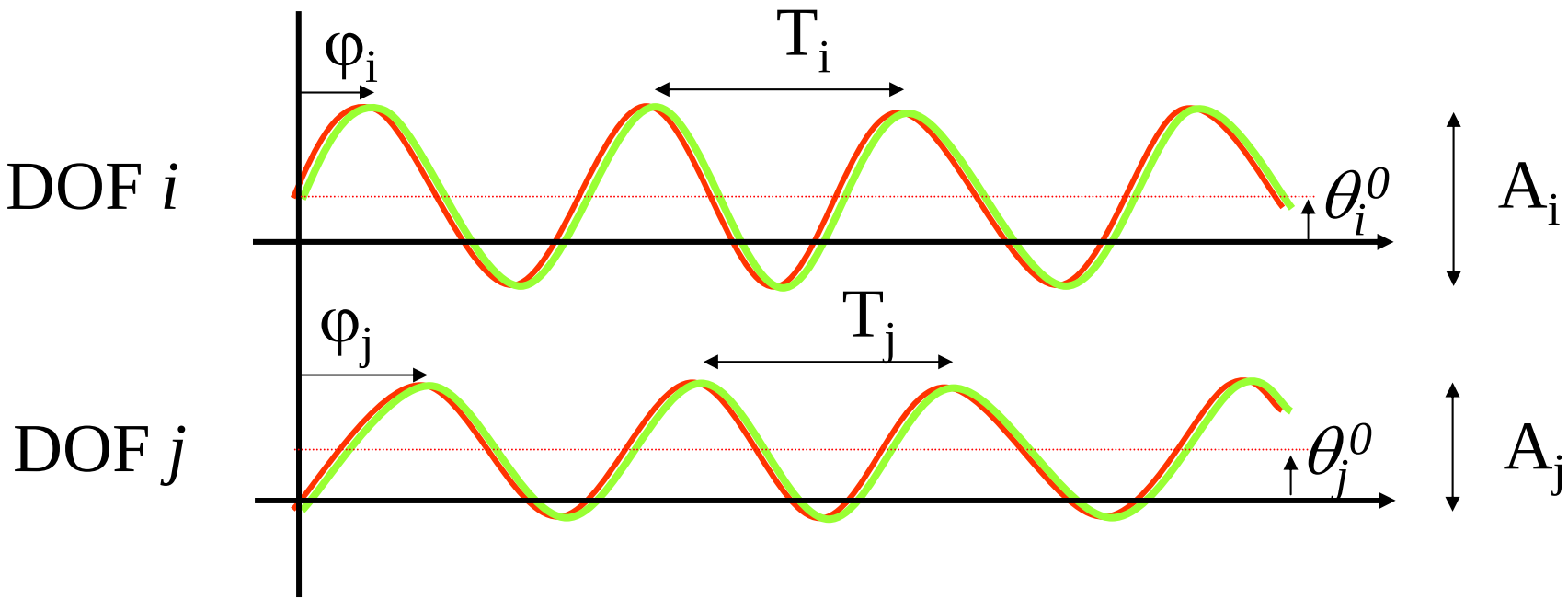
- Hand-tuned trajectory: usually only give a few via points
- Use of spline-fitting to interpolate



- Problems:
  - How to be sure that the locomotion is stable?
  - Need for mathematical tools to prove stability, cf ZMP control in a few slides

Open-loop sinus-based controller

- Desired angle for each DOF: $\theta_i = \theta_i^0 + A_i \sin(\upsilon_i t + \varphi_i)$
- Problem: finding, for each DOF *i* suitable θ_i^0, A_i, υ_i and φ_i



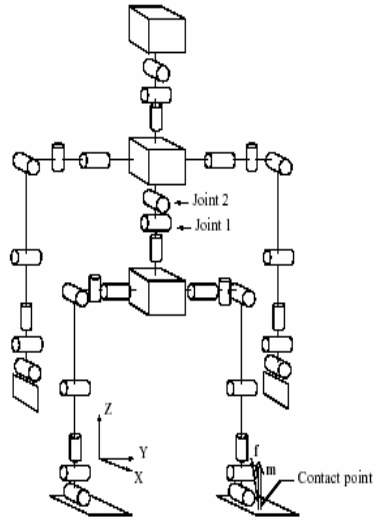
Walking based on Trajectory Methods

- **Main idea:** design walking kinematic trajectories, and use the dynamic equations to test and prove that locomotion is stable
- Trajectories are designed by trial-and-error, or from human recordings
- Most successful approach: **Zero Moment Point (ZMP)** method (Vukobratovic 1990)

Zero Moment Point Approach

- Method for proving that a trajectory is stable
- *Zero Moment Point (ZMP)*: point on the ground about which the net moment of the inertial forces and the gravity forces has no component along the horizontal planes (a.k.a. center of pressure, CoP).
- ZMP is different from the projection of the *center of mass (CoM)* on the ground
- ZMP ~ projection on the ground of the point around which the robot is rotating

ZMP Approach



- $\mathbf{P}_{ZMP}$ is such that:

$$\mathbf{r}_i = \mathbf{p}_i - \mathbf{p}_{ZMP}$$

Moment due to ext. acc.

M. due to angular acceleration

$$\sum_i^N (\mathbf{r}_i \times m_i \mathbf{a}_i + \mathbf{I}_i \boldsymbol{\alpha}_i + \boldsymbol{\omega}_i \times \mathbf{I}_i \boldsymbol{\omega}_i - \mathbf{r}_i \times m_i \mathbf{g}) = (0,0,*)^T$$

M. due to Coriolis forces

M. due to gravity

N : number of links

p_i : position of link i

m_i : mass of link i

$\mathbf{a}_i$: external acceleration

$\mathbf{I}_i$: moment of inertia (matrix)

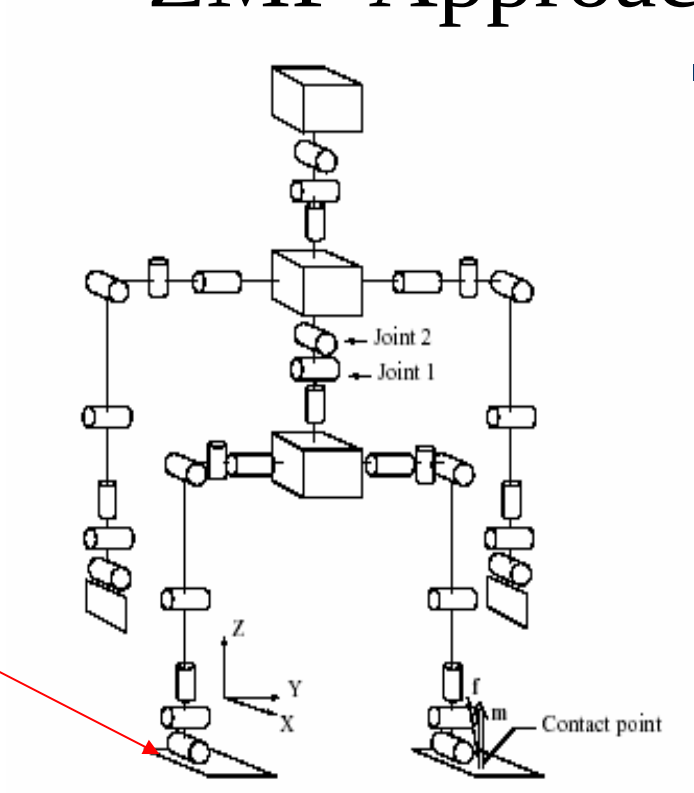
$\boldsymbol{\alpha}_i$: angular acceleration

$\boldsymbol{\omega}_i$: angular velocity

$\mathbf{g}$: gravity

Two independent equations,
Two unknowns: $p_{ZMP,X}$ $p_{ZMP,Y}$

ZMP Approach



Foot-print polygon

Locomotion is **stable** if the ZMP remains within the ***foot-print polygons***

Honda & SONY uses ZMP control

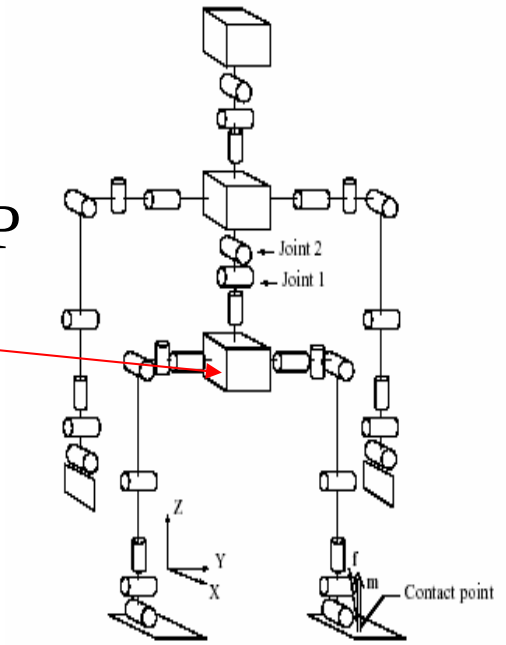
ZMP Approach

Most used method:

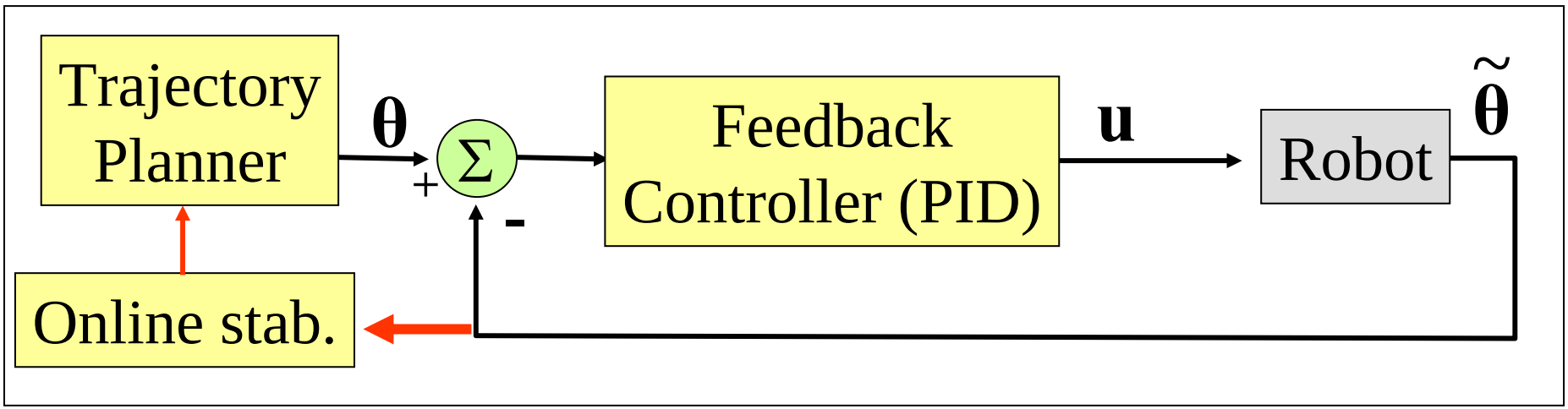
1. Human motion capture for getting trajectories,
2. Modify trajectories such that locomotion is stable according to the ZMP criterion
3. Add online stabilization to deal with perturbations.

Example of online stabilization:

- Use of hip actuators to manipulate the ZMP



Control with ZMP Approach



$\boldsymbol{\theta}$ Desired robot posture

$\tilde{\boldsymbol{\theta}}$ Actual robot posture

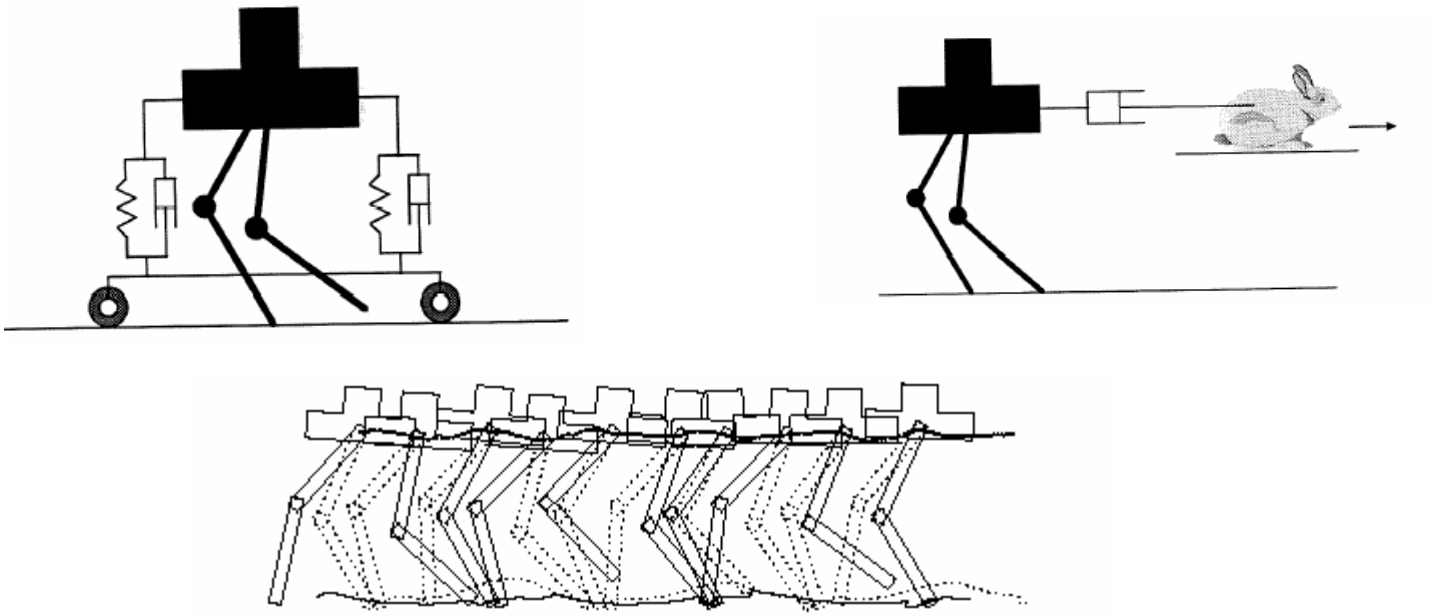
$\mathbf{u}$ Command (torque)

ZMP Approach: Conclusions

- Pros:
  - Well-defined methodology for proving stability
  - Well-suited for expensive robots that should never fall
- Cons:
  - Requires a perfect knowledge of the robot's dynamics and of the environment
  - Requires additional online control to deal with perturbations
  - Transitions from online control back to desired trajectories can be tricky
  - Define good trajectories can be time-consuming

Virtual Model Control

- Most successful example: Virtual Model Control (G.Pratt)
- Idea: create virtual elements to keep the robot upright and have it move forward
- Then compute the necessary torques such that the robot motors replicate the effect of those virtual elements

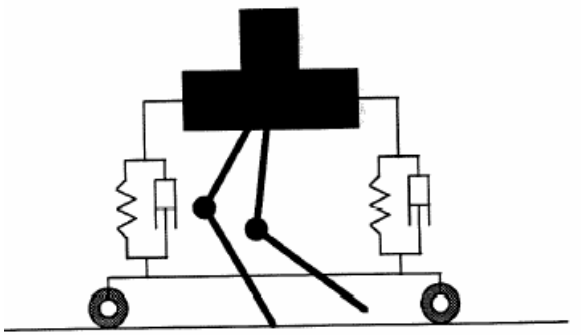


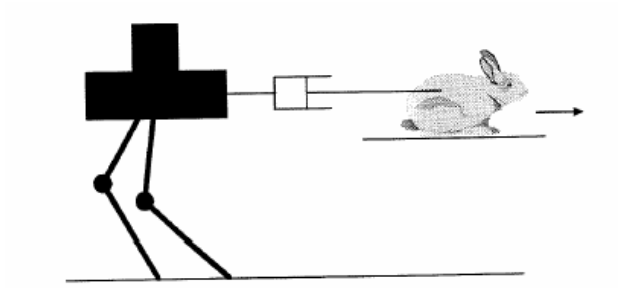
Virtual Model Control (cont'd)

- For each virtual element producing a force F, the joint torque needed to produce that virtual force can be computed with:

$$\vec{T} = \mathbf{J}^T \vec{F}$$

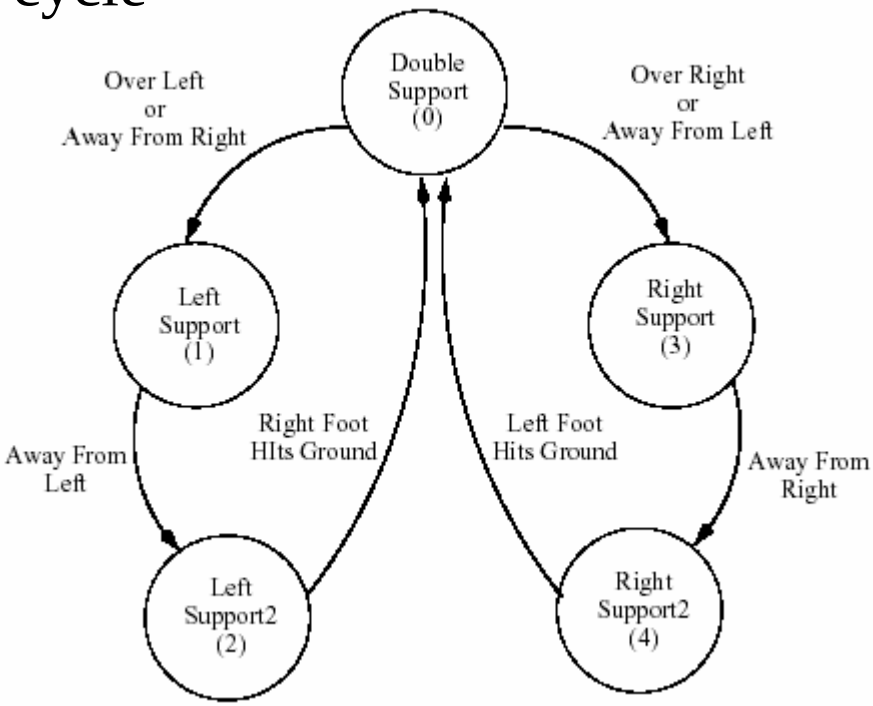
- **J** is the *Jacobian* relating the reference frame of the virtual element to the robot





Virtual Model Control (cont'd)

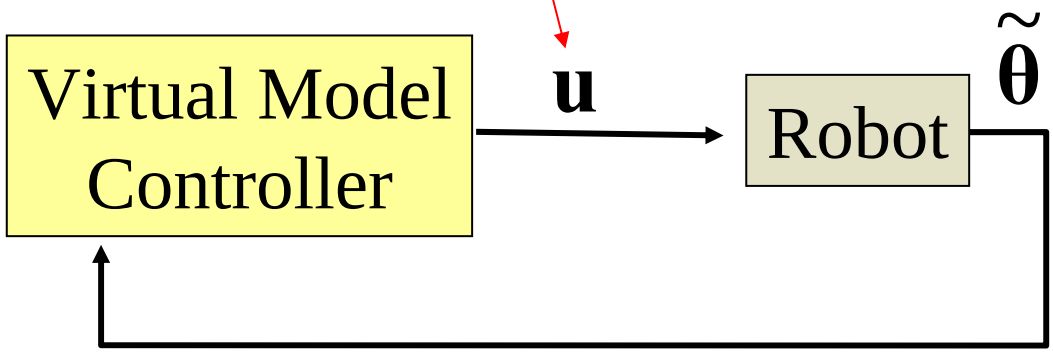
Only some motors should be activated at particular phases in the locomotor cycle



Finite state machine (set of if-then rules) for cycling through different actuation phases

Control Diagram: Virtual Model Control

Directly produces torques,
no tracking of a desired trajectory

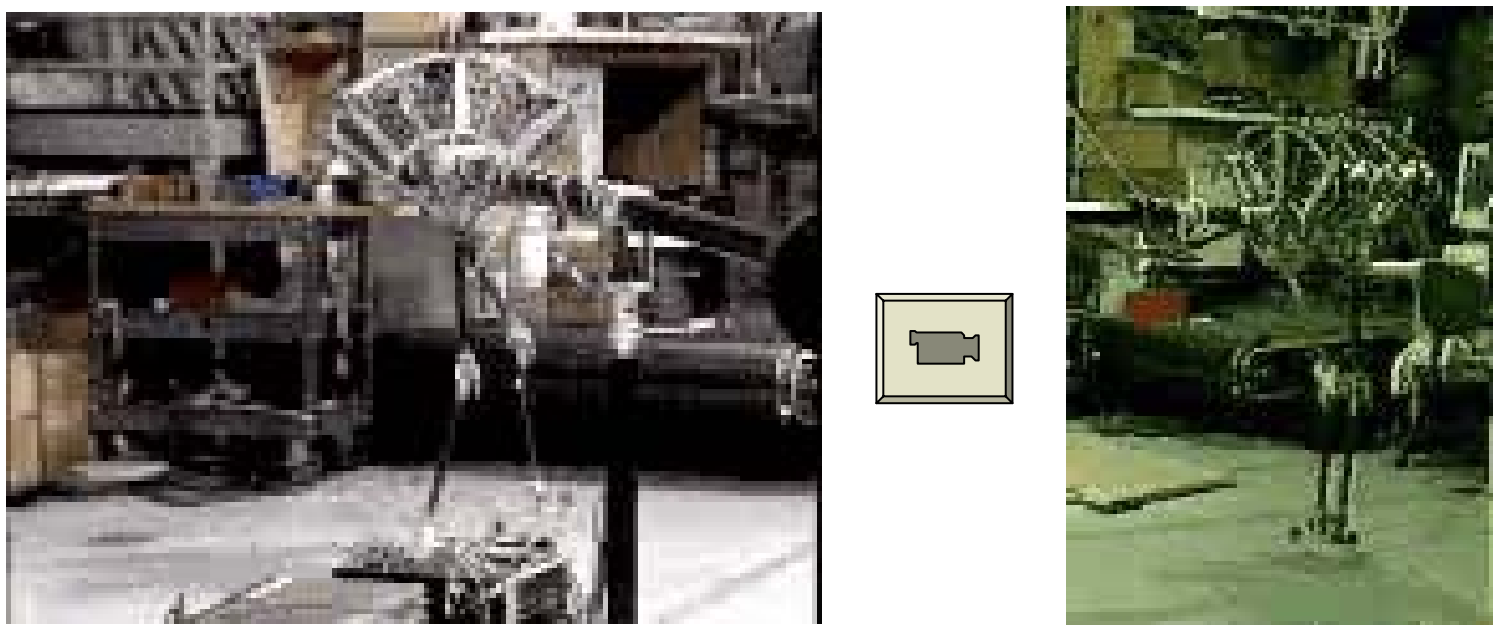


$\tilde{\boldsymbol{\theta}}$ Actual robot posture

$\mathbf{u}$ Command (torque)

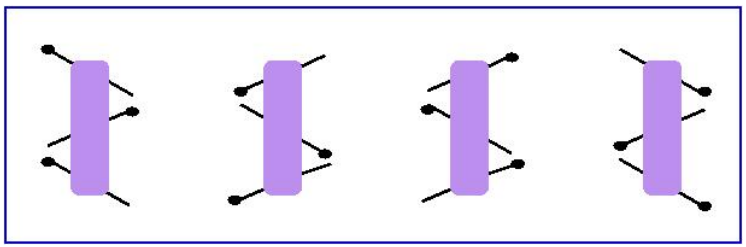
Virtual Model Control (cont'd)

- Example: Flamingo robot at MIT Leg LAB

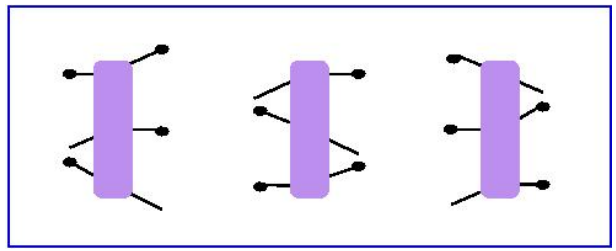


Walking: Different Types of Gait

- **Hexapod locomotion:** tripod gait, metachronal wave

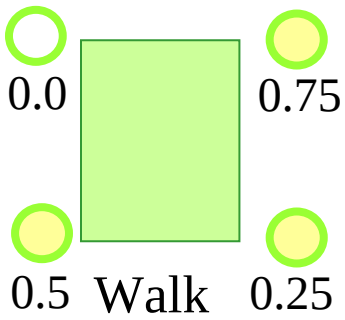


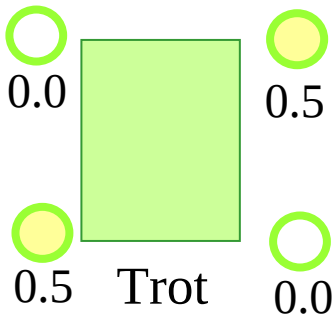
Tripod

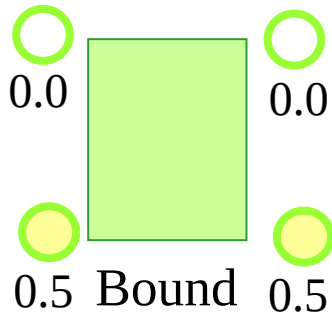


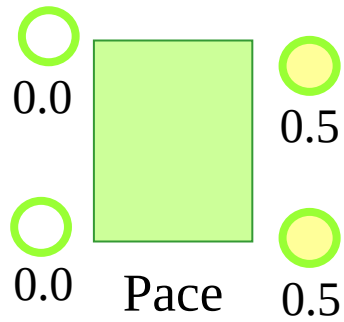
Metachronal wave

- **Quadruped locomotion:** walk, trot, gallop/bound, pace







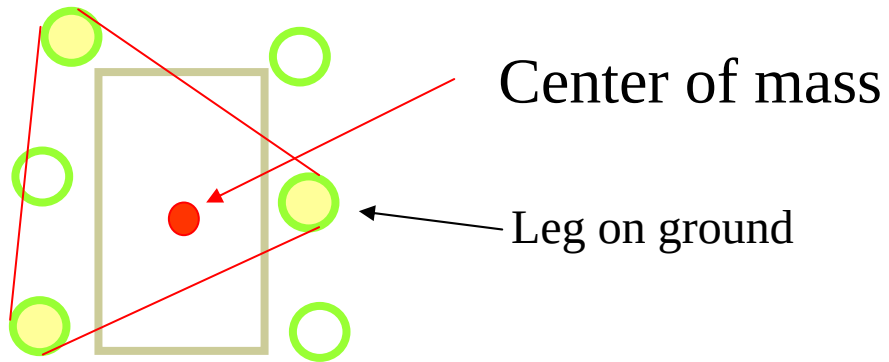


- **Biped locomotion:**
  walk (at least one leg on the ground at all times), running

Statistically vs Dynamically Stable Gaits

- ***Statically stable gait***: the center of mass is maintained at all times above the *support polygon* formed by the contacts between the limbs and the ground

Tripod gait in a hexapod robot:



- ***Dynamically stable gait***: the center of mass is maintained over the support polygon only in average

Trotting gait in a quadruped robot: