Humanoid Robotics

Bipedal Walking

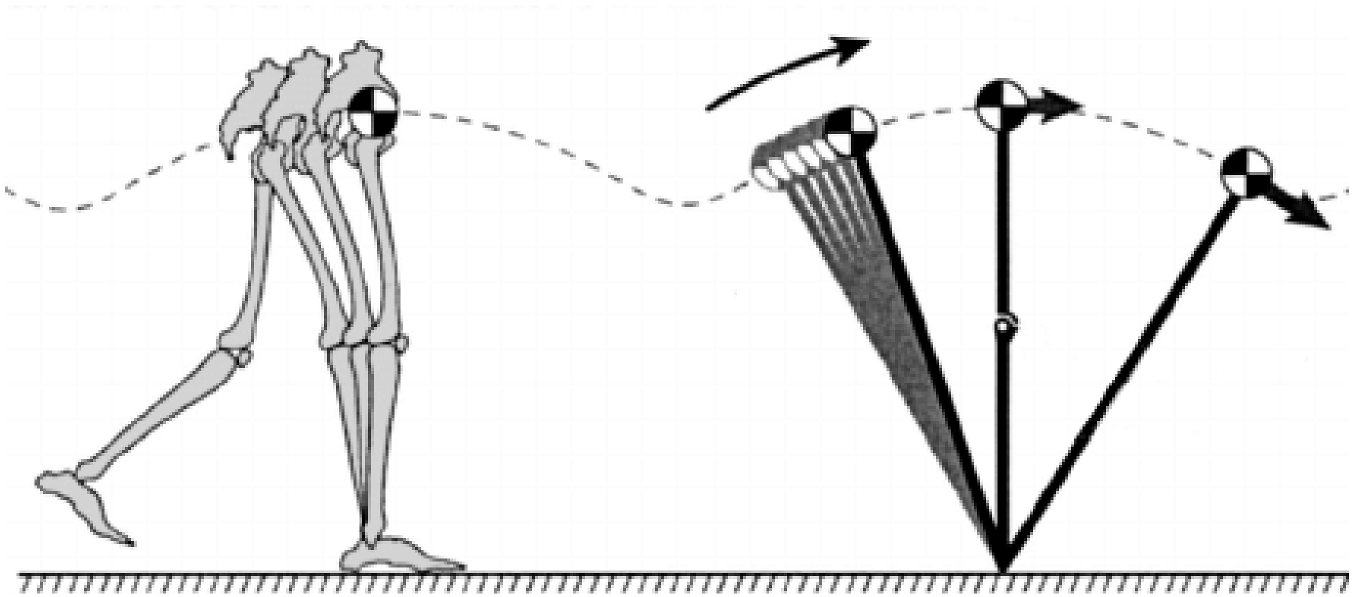
Marcell Missura

Motivation

- Bipedal walking is the most versatile form of locomotion
- Legs can go anywhere: walk, jump, run, climb, and swim
- Legs can also be used to drive cars and to ride bicycles

Motivation

- Walking on stretched legs is energy-efficient

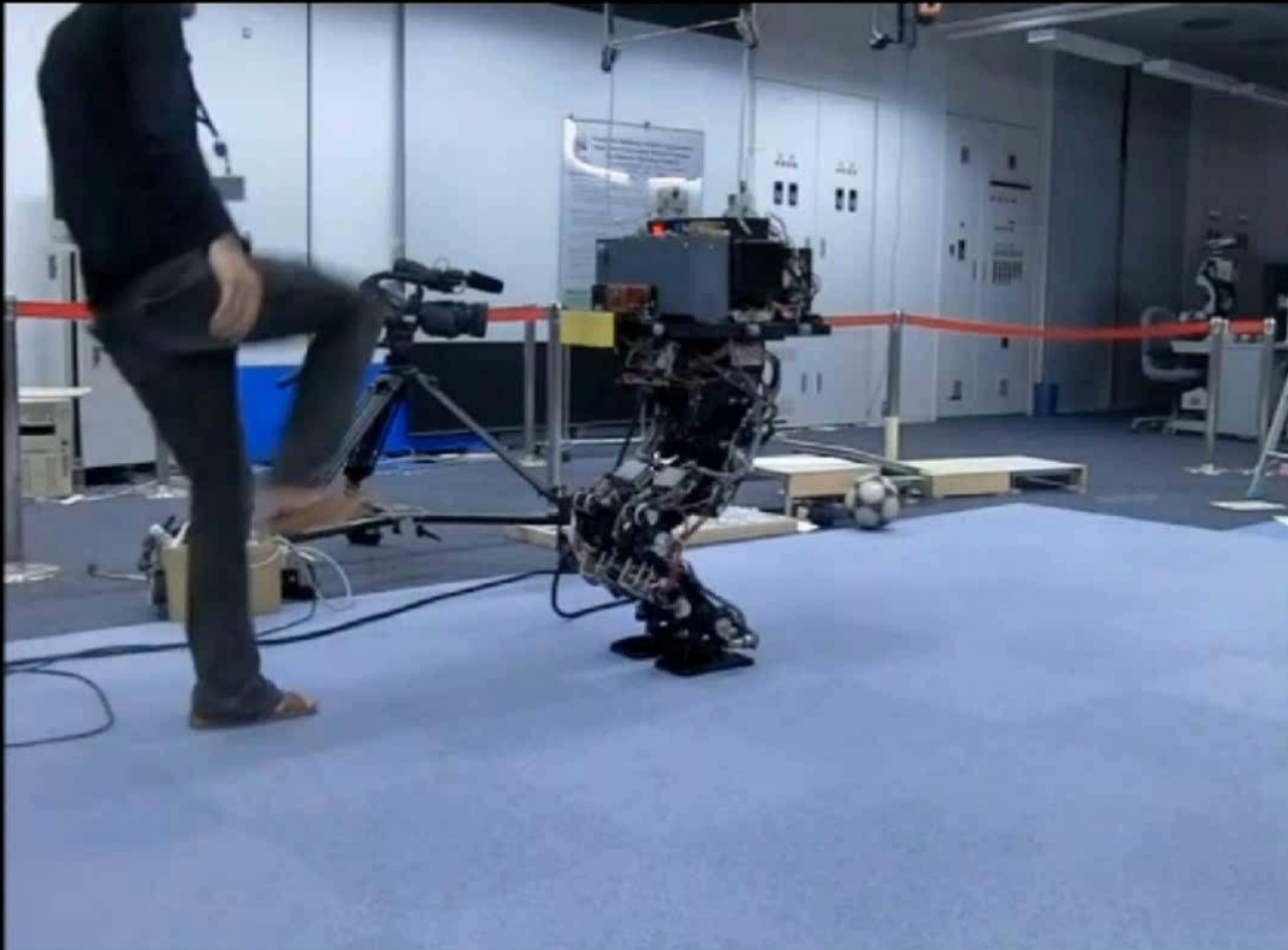


Source: Omar et.al.: Study of Bipedal Robot Walking Motion in Low Gravity: Investigation and Analysis

Motivation

- Bipedal walking is the most versatile form of locomotion
- Legs can go anywhere: walk, jump, run, climb, and swim
- Walking on stretched legs is energy-efficient
- However: **balancing is difficult**


AIST




The robot attempts to maintain the
reference velocity after a disturbance.

Passive Dynamic Walking

- Walking only with a mechanical construction
- No control and no actuation apart from a shallow slope
- Proves that walking is strongly based on the natural dynamics of legs


Nagoya Inst. Tech. June 2005. About 4000 steps (about 35 minutes).
Yoshito Ikemata, Akihito Sano & Hideo Fujimoto


TUDelft

Central Pattern Generated Walking

- We add actuation and control and obtain open-loop walking robots
- Parameterized periodic motion patterns are used to generate targeted stepping motions
- Control of the walking velocity and direction
- Still **no control of balance**!

Kinematics

Kinematics

- Kinematics describes the geometry and the **motion of physical bodies.**
- Position, velocity, and acceleration are considered, but no forces (that would be dynamics).
- The motion of kinematic chains is described by **rigid body transformations.**

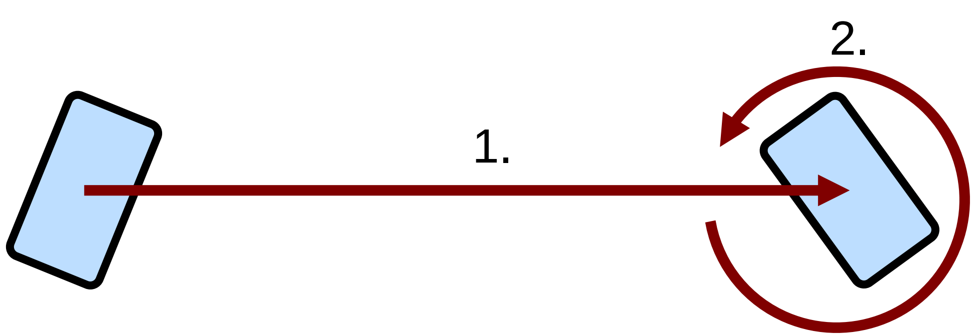
Kinematics

- **Forward kinematics** computes the pose of a robot using the measured joint angles.
- **Inverse kinematics** computes the joint angles needed to reach a specific pose or end effector position.
- **Abstract kinematics** computes joint angles from abstract leg pose parameters. This has certain advantages over inverse kinematics.

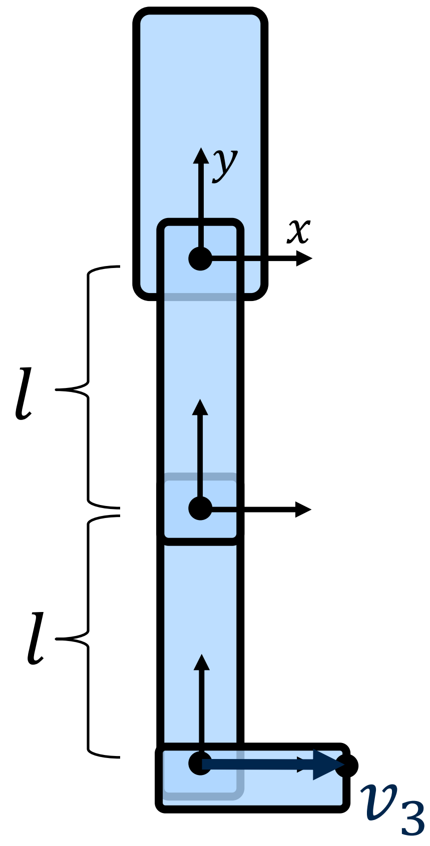
Rigid Body Transform

- A rigid body transformation consisting of a translation by a vector (d_x, d_y) and then a rotation by an angle α can be represented by a multiplication with a homogeneous transformation matrix.

$$\begin{bmatrix} {v_x}' \\ {v_y}' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha & d_x \\ \sin\alpha & \cos\alpha & d_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ 1 \end{bmatrix}$$



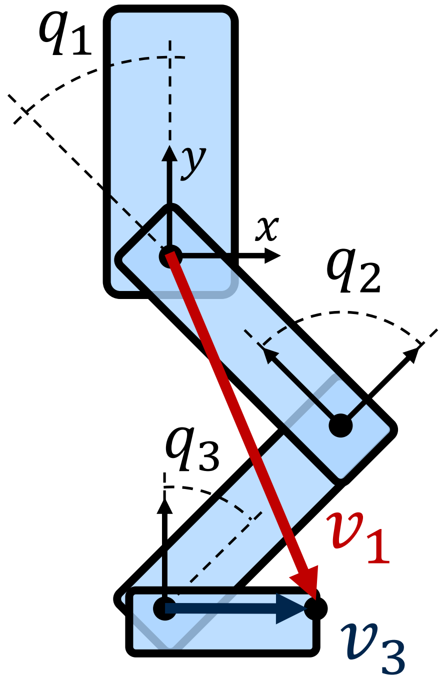
Forward Kinematics



- We begin with a known **kinematic model**, i.e., the lengths of the links of the kinematic chain and the position of the end effector v_3.
- The **zero pose** is a special pose calibrated such that the transformation between the links is known and the joint angles q_1, q_2, and q_3 are zero.

Forward Kinematics (2)

- By chaining multiple rigid body transforms, we can compute the position of the end effector in the base frame.

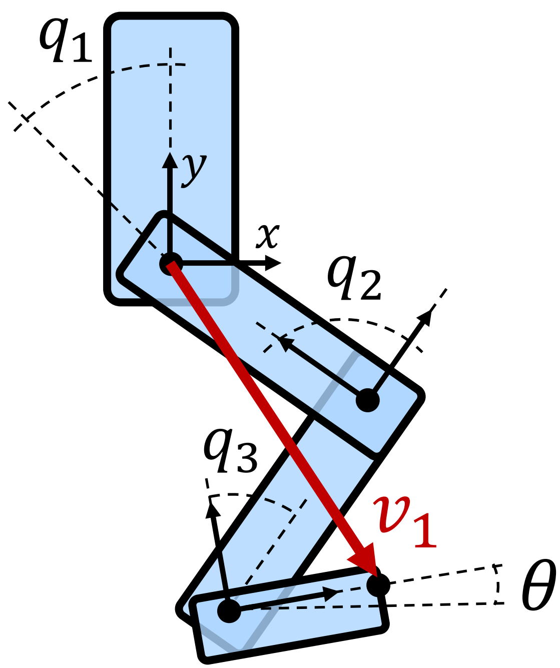


$$T_1(q_1) = \begin{bmatrix} \cos q_1 & -\sin q_1 & l \sin q_1 \\ \sin q_1 & \cos q_1 & -l \cos q_1 \\ 0 & 0 & 1 \end{bmatrix}$$

$$v_1 = T_1(q_1) \cdot T_2(q_2) \cdot T_3(q_3) \cdot v_3$$

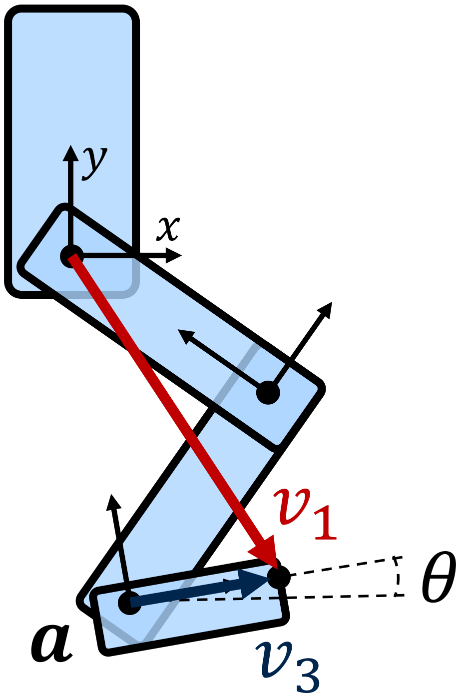
Inverse Kinematics

- Inverse kinematics is the opposite of forward kinematics
- Input: v_1 (end effector), θ (foot angle)
- Output: q_1, q_2, q_3 (joint angles)



Inverse Kinematics (2)

- First, we compute the location of the ankle joint $\boldsymbol{a}$ using the end effector vectors $\boldsymbol{v_1}$ and $\boldsymbol{v_3}$ and the foot angle θ.



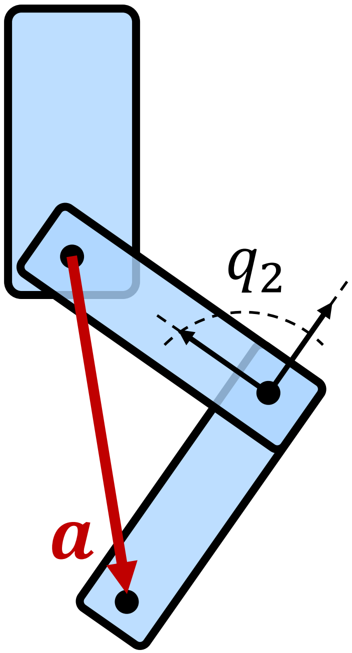
$$\boldsymbol{a} = (v_{1_x} - |\boldsymbol{v_3}| \cos\theta \, , \, v_{1_y} - |\boldsymbol{v_3}| \sin\theta)$$

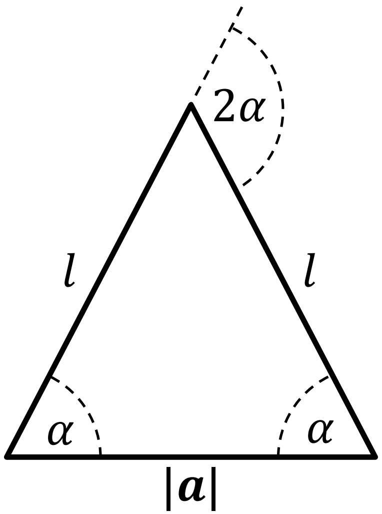
Inverse Kinematics (3)

- Then, we compute the angle q_2 of the knee joint using a triangle principle.

$$\boldsymbol{a} = (v_{1_x} - |\boldsymbol{v_3}|\cos\theta,\; v_{1_y} - |\boldsymbol{v_3}|\sin\theta)$$

$$q_2 = -2\cos^{-1}\frac{|\boldsymbol{a}|}{2l}$$



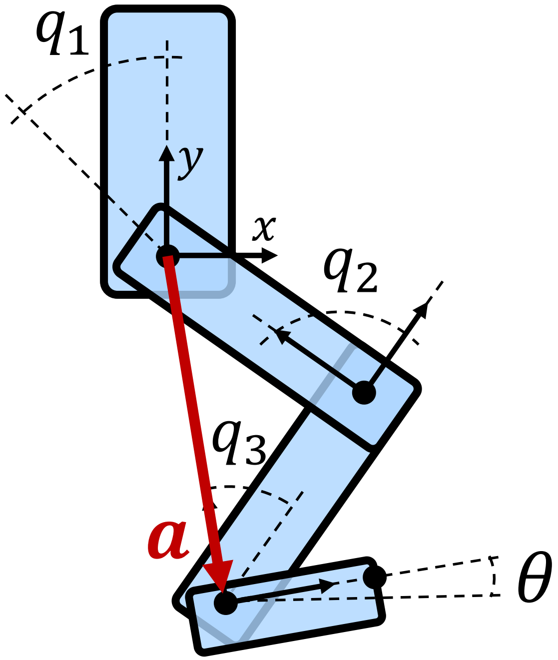


$$\alpha = \cos^{-1}\frac{|\boldsymbol{a}|}{2l}$$

Inverse Kinematics (4)

- Now we can compute the remaining joint angles.

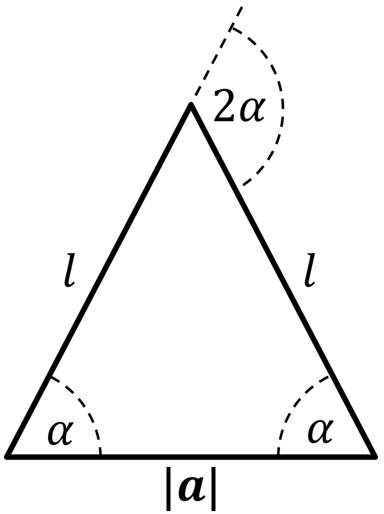
$$\boldsymbol{a} = (v_{1_x} - |\boldsymbol{v_3}| \cos\theta,\ v_{1_y} - |\boldsymbol{v_3}| \sin\theta)$$



$$q_2 = -2\cos^{-1}\frac{|\boldsymbol{a}|}{2l}$$

$$q_1 = \sin^{-1}\frac{a_x}{|\boldsymbol{a}|} - \frac{q_2}{2}$$

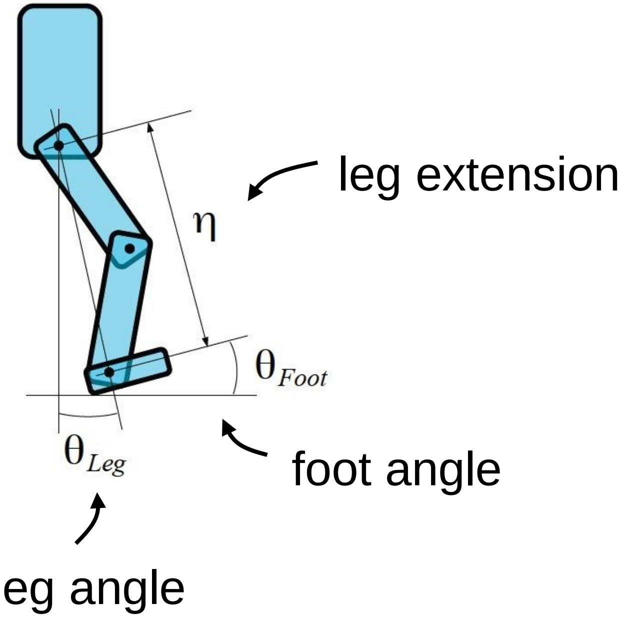
$$q_3 = \theta - q_1 - \frac{q_2}{2}$$



$$\alpha = \cos^{-1}\frac{|\boldsymbol{a}|}{2l}$$

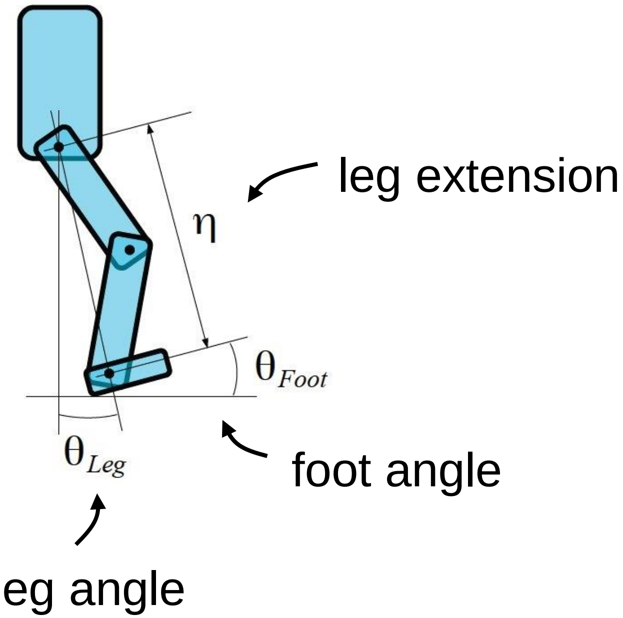
Abstract Kinematics

- In Abstract Kinematics, the leg pose is parameterized in a different way as opposed to an end effector position. The computations are simpler and **no kinematic model is needed**.



Abstract Kinematics (2)

- Input: $\eta \in [0,1]$ (leg extension), θ_{Leg} (leg angle), θ_{Foot} (foot angle)
- Output: q_1, q_2, q_3 (joint angles)



$$q_2 = 2\cos^{-1}\eta$$

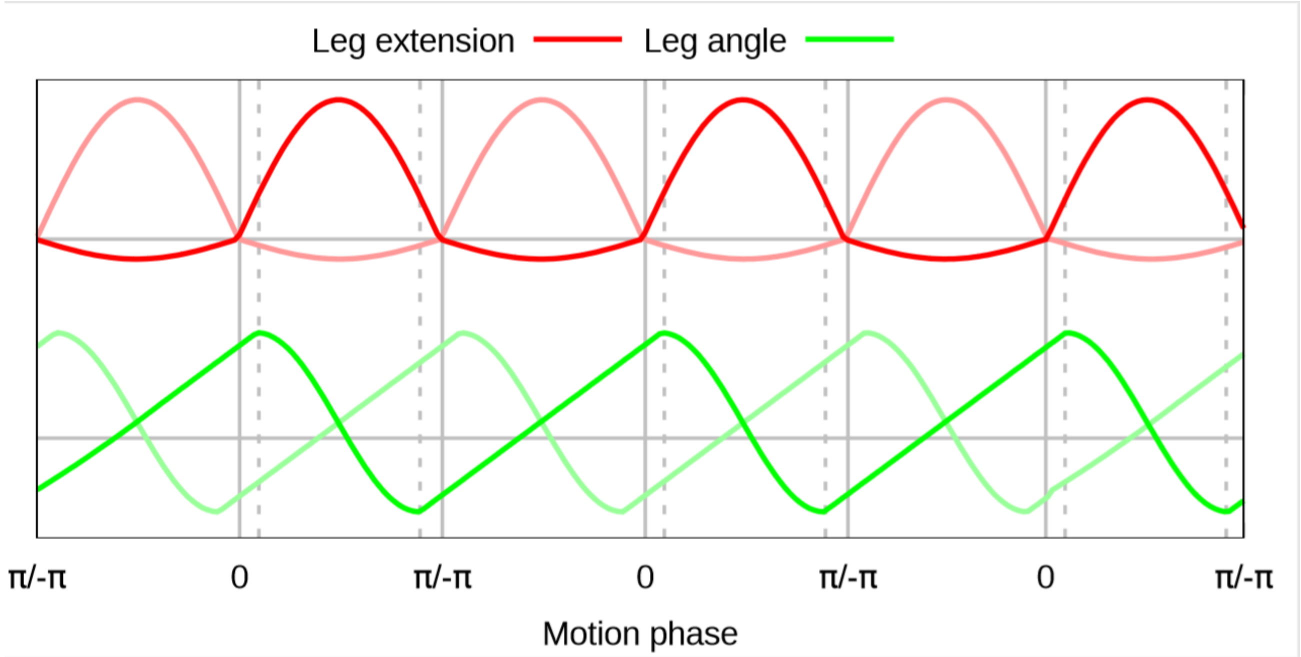
$$q_1 = \theta_{Leg} - \frac{q_2}{2}$$

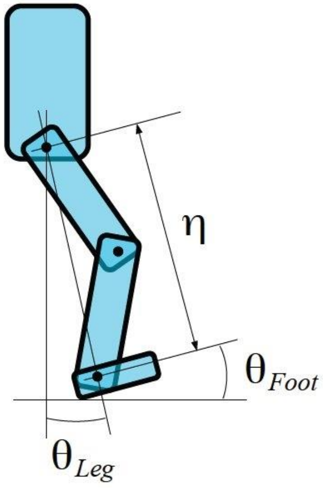
$$q_3 = \theta_{Foot} - \theta_{Leg} - \frac{q_2}{2}$$

Motion Pattern Generation

Step Motion Pattern Generator

- Using either Inverse Kinematics or Abstract Kinematics, we can make a robot walk with the help of hand-crafted oscillating motion signals for the legs.

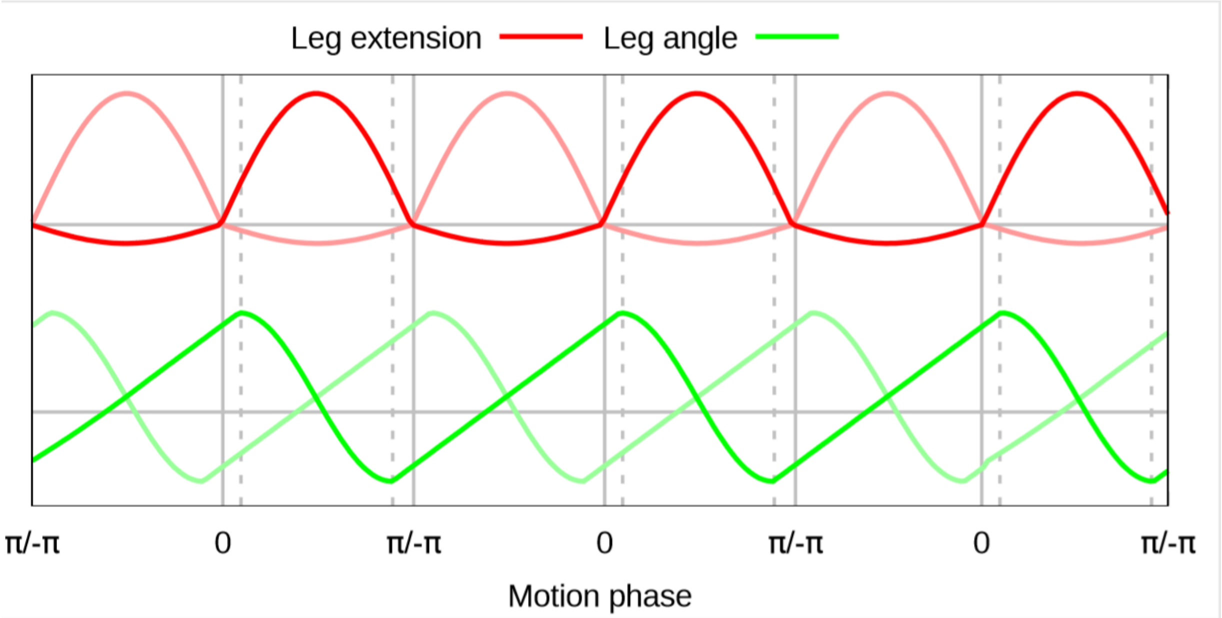


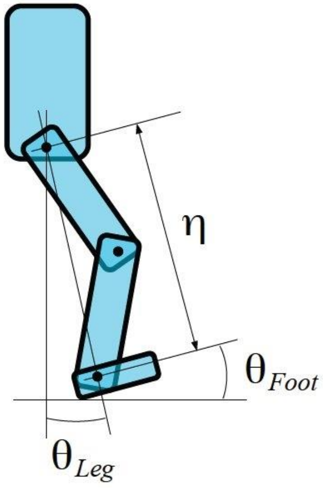


Step Motion Pattern Generator

Leg Lifting:
$$\eta(\mu) = \begin{cases} C_1 \sin\mu\,, & \mu < 0 \\ C_2 \sin\mu\,, & \mu \geq 0 \end{cases}$$

Leg Swing:
$$\theta_{Leg}(\mu, S) = \begin{cases} S\left(\dfrac{2\mu+\pi}{\pi}\right), & \mu < 0 \\ S\cos\mu\,, & \mu \geq 0 \end{cases}$$





Minimal Walking Function

while(true)

{

$\mu = \mu + \Delta\mu$

$(q_1, q_2, q_3) = \text{abstractKinematics}\left(\eta(\mu), \theta_{Leg}(\mu, S)\right)$

$\text{sendToRobot}(q_1, q_2, q_3)$

}