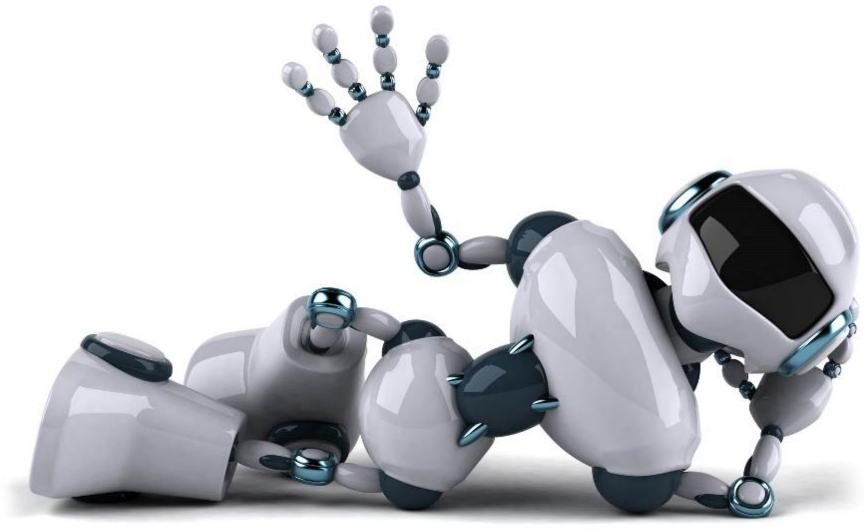
Module 4: Robotic Locomotion

Nicholas Ho

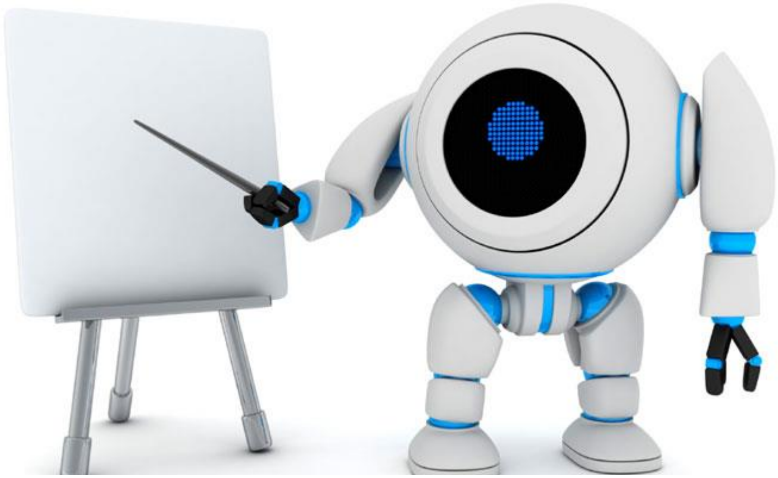
Objectives:

- Classify the **various means of robotic mobility**, their **corresponding models** and **means of computation**



Contents

- Introduction to Robotic Mobility methods
- Mobility models and computation
- Introduction to Robot Simulation Software



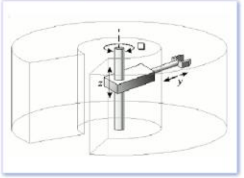
Chapter 1: Introduction to Robotic Mobility Methods

All Types of Robots by Locomotion

STATIONARY ROBOTS

Cartesian Robots

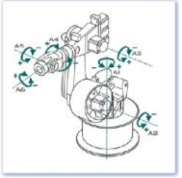
Cylindrical



Spherical

SCARA

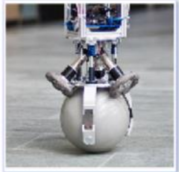
Articulated



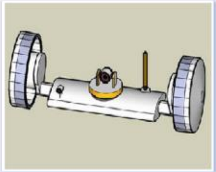
Parallel

WHEELED ROBOTS

Single Wheel



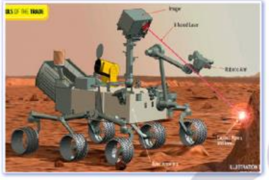
2 Wheeled



3 Wheeled

4 Wheeled

6 Wheeled



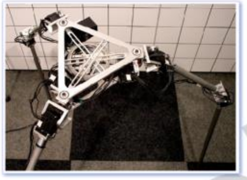
Tracked Robots

LEGGED ROBOTS

One Leg

Bipedal

Tripedal

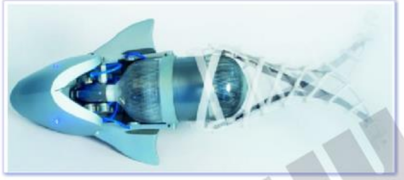


Quadrupedal

Hexapod

Many Legs

SWIMMING ROBOTS



FLYING ROBOTS

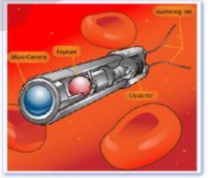
Robotic Balls

SWARM ROBOTS

MODULAR ROBOTS

MICRO Robots

NANO Robots



SOFT ROBOTS

SNAKE Robots

CRAWLER Robots

HYBRID Robots

Understanding Locomotion

Two basic ways of using effectors:

1. to move the robot around → **locomotion**
2. to move other object around → **manipulation**

These divide robotics into **two mostly separate categories:**

1. **mobile robotics**
2. **manipulator robotics**

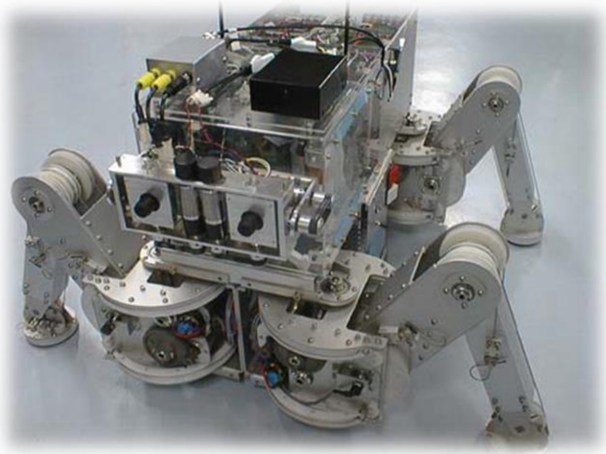
Understanding Locomotion

- **Definition:** The self-powered, patterned motion of limbs or other anatomical parts by which an individual customarily **moves itself from place to place**
- The need for locomotion in mobile robots:
  - Mobility is desired for a mobile robot to **perform its tasks**
  - Locomotion methods **provide mobility** for mobile robots
  - **Functionality** of mobile robots would be severely **impaired with loss of mobility**
  - Some mobile robots / vehicles **may possess more than one method of locomotion to safeguard its mobility** (e.g. Amphibious robots)

Common Methods for Achieving Mobility

Many ways to achieve locomotion:

1. Legged locomotion
   - Bipedal (two-legged)
   - Poly-pedal (multi-legged)

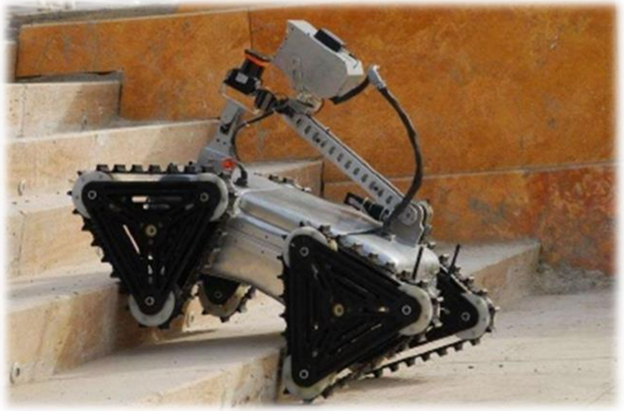


2. Wheeled locomotion
   - Two-wheeled (bicycle, differential drive)
   - Four-wheeled (car-type steering)

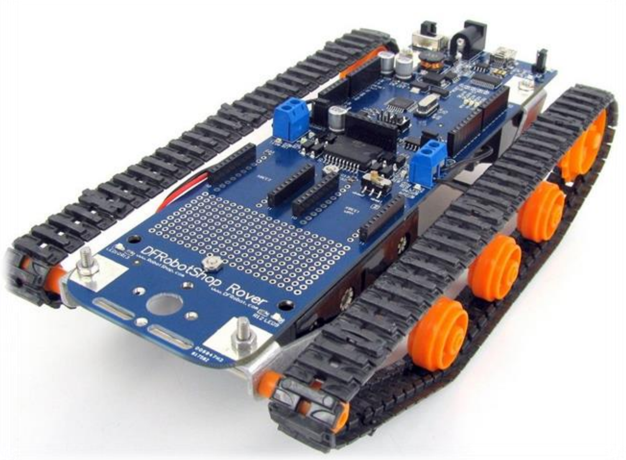
Common Methods for Achieving Mobility

Many ways to achieve locomotion (Cont):

3. Tracked locomotion
   - E.g. Tank Treads



4. Chained locomotion



Common Methods for Achieving Mobility

Many ways to achieve locomotion (Cont):

5. Aerial locomotion

- ➢Vertical Lift
- ➢Aerofoil Lift
- ➢Ornithopters
- ➢Lighter-than-air (e.g. airships, blimps, weather balloons)

Common Methods for Achieving Mobility

Many ways to achieve locomotion (Cont):

6. Underwater locomotion
   - ➢E.g. Submarines

7. Marine locomotion
   - ➢E.g. boats, hovercrafts

8. Combined locomotion
   - ➢E.g. amphibious vehicles (Land + Sea)

Selection of Locomotion Methods

Various factors to consider:

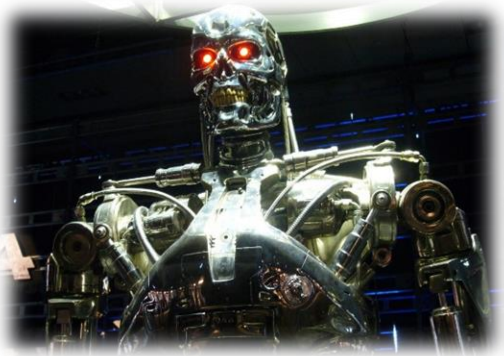
1. **Nature of the terrain**
   - ➢Smooth, planar, structured surfaces –ideal for wheeled robots, since legged robots may not be as efficient / fast
   - ➢Unstructured, undulating and rough terrain –may require more powerful / complicated wheel drives or legged locomotion
2. **Nature of the task**
   - ➢The task must be achievable by the choice of locomotion means used, i.e. must lie within the workspace defined by the locomotion means

Selection of Locomotion Methods

Various factors to consider (Cont):

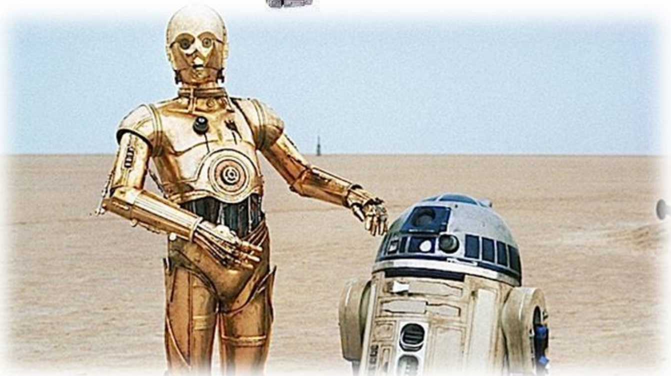
3. Physical considerations

- **a) Power**
  - ➢ Locomotion means requiring large amounts of power (i.e. fuel inefficient), or may limit the maximum range of the mobile robots
- **b) Cost**
  - ➢ More complicated methods of locomotion (e.g. powered flight) are generally more expensive
- **c) Reliability**
  - ➢ Ease of repair, durability
- **d) Range**
  - ➢ Determined by other factors mentioned (e.g. power)
- **e) Desired speed**
- **f) Weight of the mobile robot**






TARS



Mobility for mobile robots

- Most-commonly-used locomotion methods for mobile robots operating on land makes use of **wheels** or **legs**
- Food for thought (to be discussed later):

Can mobile robots be modified to harass the benefits of both legged and wheeled locomotion???

Mobility for mobile robots: Wheeled Locomotion

- Wheeled locomotion can be seen as the **joined effect of two actions**
  1. **Driving**
     - Provides the action to set the mobile robot in motion
     - Usually provided by the torque from an engine / motor
  2. **Steering**
     - Allows a mobile robot to turn to follow the desired course or trajectory
- **Various types of steering:**
  - Ackermann (car-type) steering
  - Differential drive steering
  - Steering with independent wheels

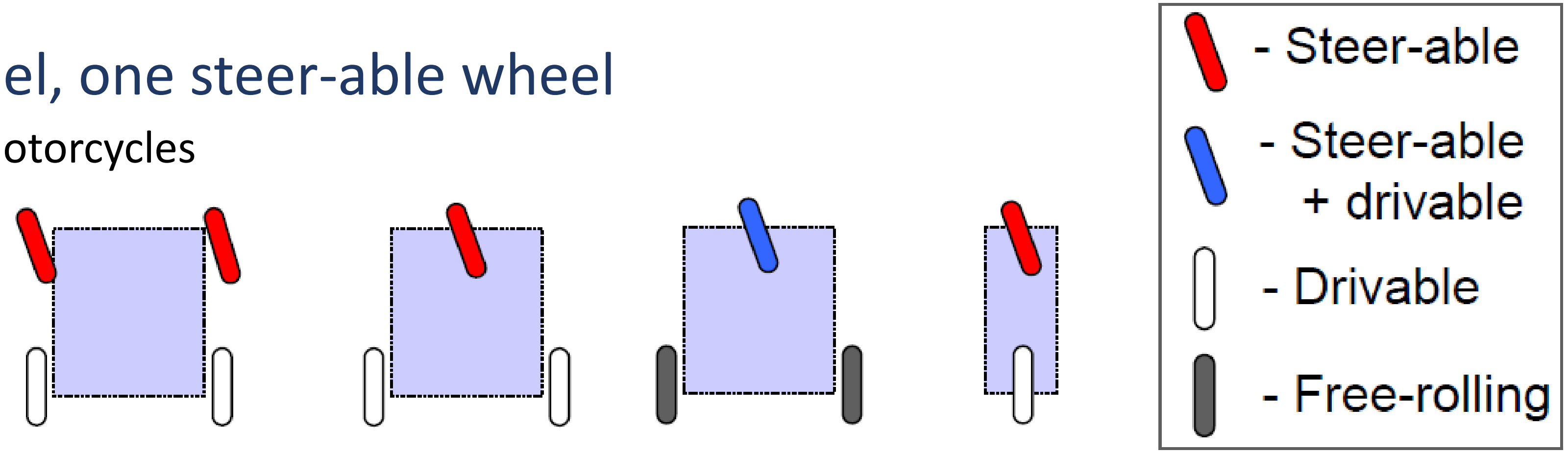
Mobility for mobile robots: Wheeled Locomotion – Ackermann Steering

- **Commonly-used car-type steering** found in the majority of today's automobiles
- Drive (for a typical 4-wheel vehicle)
  - Front-wheel drive
  - Rear-wheel drive
  - 4-wheel drive (4WD)
- **Front wheels are usually used for steering**
  - Steering is accomplished by collectively changing the angle of the front wheels relative to the rear wheels
- **Locomotion is achieved by determining:**
  - The **torque to deliver to the motors** – Driving
  - The **angles to turn the axles of the wheels** – Steering

Mobility for mobile robots: Wheeled Locomotion – Ackermann Steering

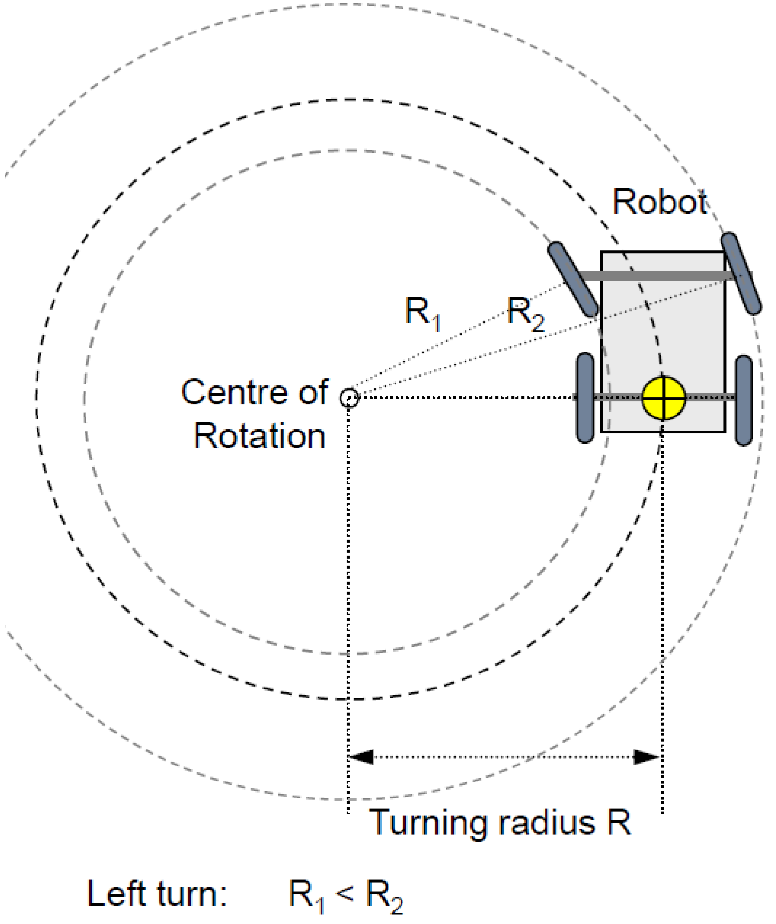
Popular configurations:

1. Two drive wheels, two steer-able wheels
   - E.g. Typical automobiles
2. Two drive wheels, one steer-able wheel
3. Two free-rolling wheels, one drive-able and steer-able front wheel
   - E.g. tricycles
4. One drive wheel, one steer-able wheel
   - E.g. Bicycles, motorcycles



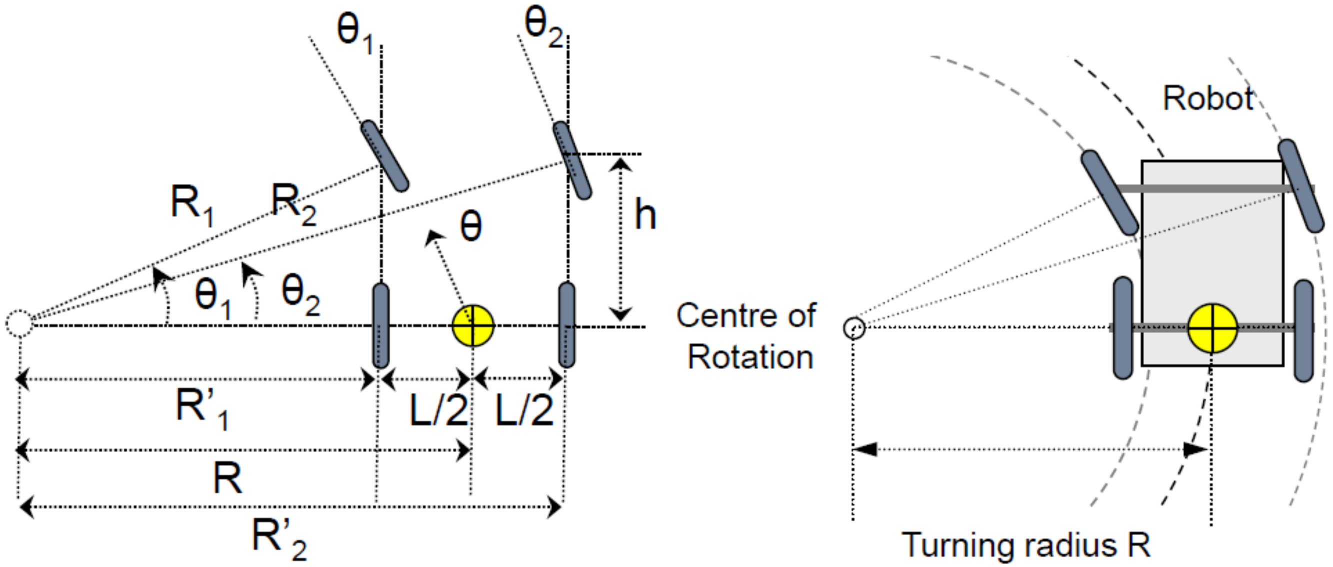
Mobility for mobile robots: Wheeled Locomotion – Ackermann Steering

- As a vehicle is being steered, the traversed path falls on the circumference of a circle with a radius known as the **turning radius**, with a centre of rotation that falls along the line extended from the fixed axle.
- Requirement
  - ➢ **Wheels on the inside and outside of a turn need to trace out circles of different radii**



Mobility for mobile robots: Wheeled Locomotion – Ackermann Steering

- **Computing the instantaneous turning radius and heading:**



$$R = R'_1 + L/2 = R'_2 - L/2$$
$$= h\cot\theta_1 + L/2 = h\cot\theta_2 - L/2 \qquad (4.1)$$
$$\theta = \frac{(\theta_1 + \theta_2)}{2} \qquad (4.2)$$

Mobility for mobile robots: Wheeled Locomotion – Ackermann Steering

Advantages:

- **Efficient**
- More **precise** than differential steering in following a specific path
- **Suitable for higher speeds due to its stability**

Disadvantages:

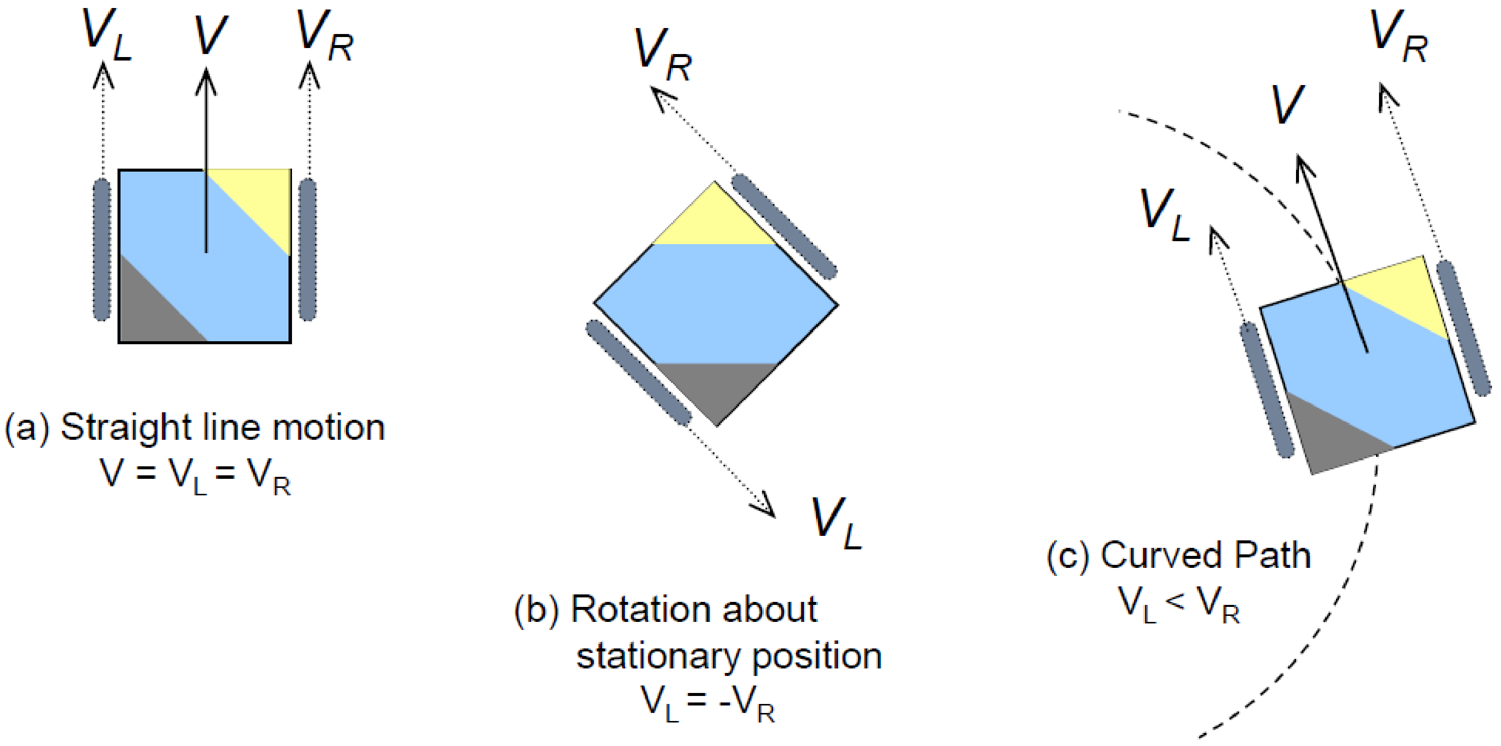
- **Inability to change orientation** / rotate on the spot
- There is a **limit to the tightness of the turn traversable**
- **Motion planning is more difficult** than differential drive steering
- Correct implementation of linkages and steering mechanisms is non-trivial

Mobility for mobile robots: Wheeled – <u>Differential Drive Steering</u>

- The steering is provided by driving wheels on one side of the mobile robot **at a different rotational velocity relative to wheels on the other side** of the mobile robot
- Minimum number of driving wheels required is two
  - E.g. Soccer robots usually employ 2 wheels
  - 4-wheel, or even 6-wheel configurations are common

Mobility for mobile robots: Wheeled – Differential Drive Steering

- **Consider the two-wheeled differential drive robot:**



Mobility for mobile robots: Wheeled – Differential Drive Steering

Advantages:

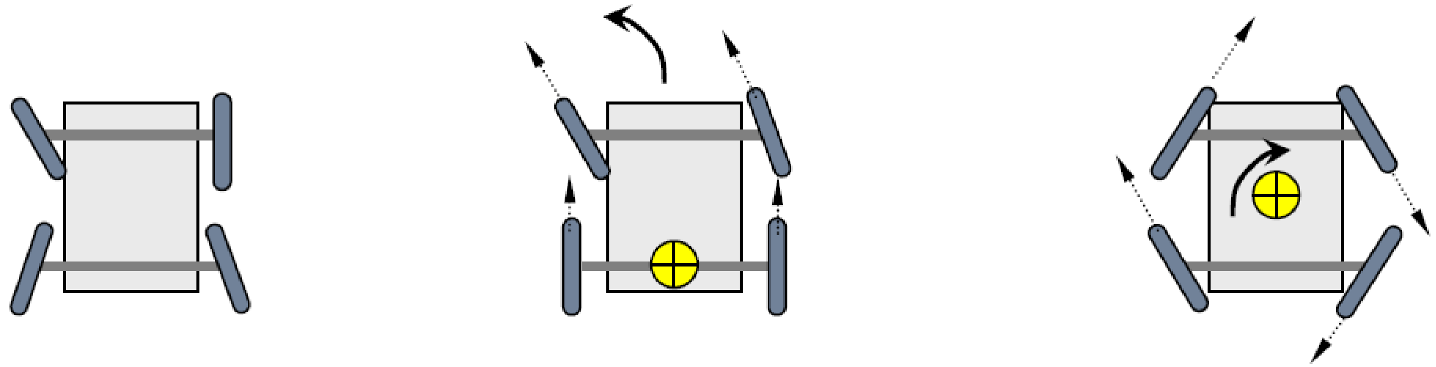
- **Ease of implementation**
- **Simpler to control**
- **Ability to change orientation** on the same spot

Disadvantages:

- **Not as efficient** as Ackerman steering or independently steered wheels when 4 or more wheels are used
  - Some of the wheels need to skid to some extent in order to complete the turn
- **Unstable at high speeds**

Mobility for mobile robots: Wheeled – <u>Steering with Independent Wheels</u>

- **Each wheel can pivot independently**
- This is the most general form of steering
  - ➢ In the case of a 4-wheeled robot, it reduces to Ackermann steering when only the front wheels are allowed to pivot
  - ➢ Depending on the desired posture, **the angle and velocity of each wheel can be determined using inverse kinematics**



(a) Independent wheels (b) Fixing the angle of rear wheels reduces to car-type steering (c) Rotation on the spot

Mobility for mobile robots: Wheeled – Steering with Independent Wheels

Advantages:

- **Able to change orientation** on the spot
- **Able to allow robots to move sideways** (overcoming the non-holonomic constraints of the other types of steering)

Disadvantages:

- Solving for each of the wheel's velocity and angle is **complicated**

Mobility for mobile robots: Wheeled Locomotion

- **Differential Drive Steering <u>OR</u> Steering with Independent Wheels?**

Source: https://www.youtube.com/watch?v=iD2zvpv3vfQ

Mobility for mobile robots: Legged Locomotion

- Legs provide a very powerful form of locomotion that **can handle many different types of terrain**
- Predominant type of locomotion in the animal kingdom especially higher beings (e.g. mammals and reptiles)
- **Two types:** Bipedal (two-legged) & Poly-pedal (>2 legs)
- Advantages of legs over wheels
  - Legged robots are able to **access multiple types of terrain**, especially unstructured, undulating ones
- Disadvantage of legs
  - Typically **slower than wheels**

Mobility for mobile robots: Legged Locomotion – Bipedal

- Bipedal (two-legged) locomotion is a very active research focus in humanoid robotics
- **Many challenges** in achieving good bipedal motion and the complexity of this problem separates bipedal locomotion from other forms of legged locomotion as a class of its own
- Challenges in Bipedal locomotion
  - ➢ **Stability** –Static stability versus Dynamic Stability
  - ➢ **Large numbers of degrees of freedom**

Mobility for mobile robots: Both Legged & Wheeled Locomotion???

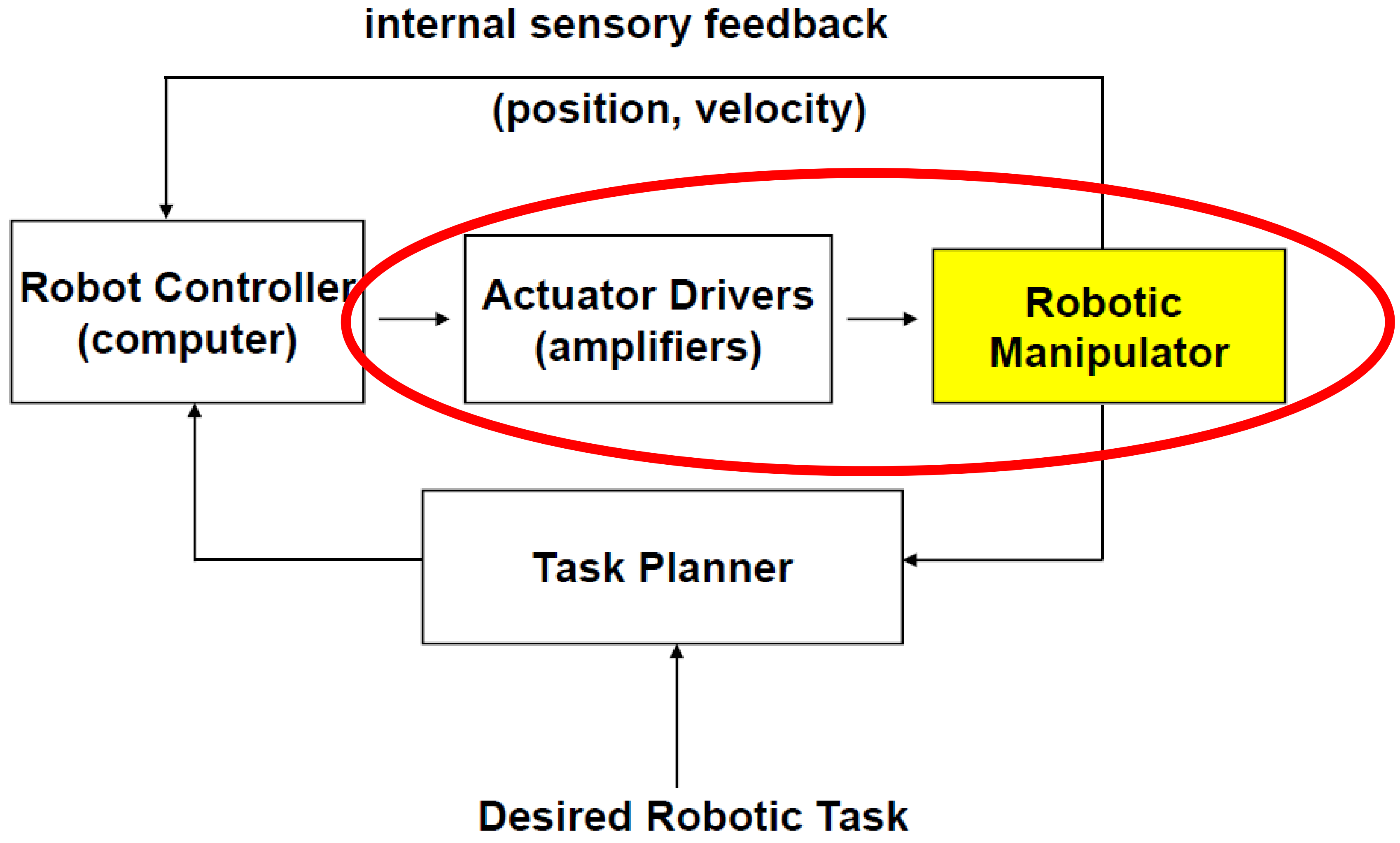
Source: https://www.youtube.com/watch?v=syUfxj4OYi8

Mobility for mobile robots: Both Legged & Wheeled Locomotion???

Source: https://www.youtube.com/watch?v=v7OVqXm7_Pk

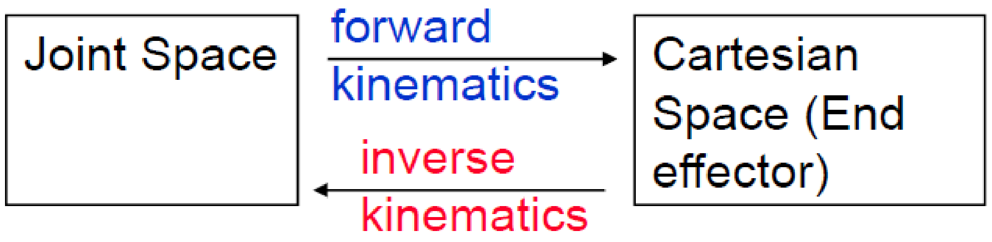
Chapter 2: Mobility Models and Computation

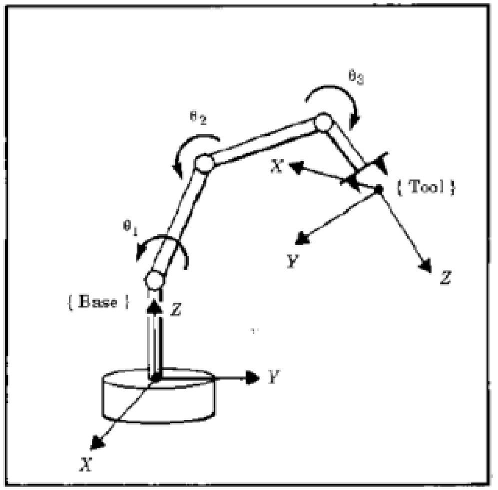
Robot System Components



Kinematics

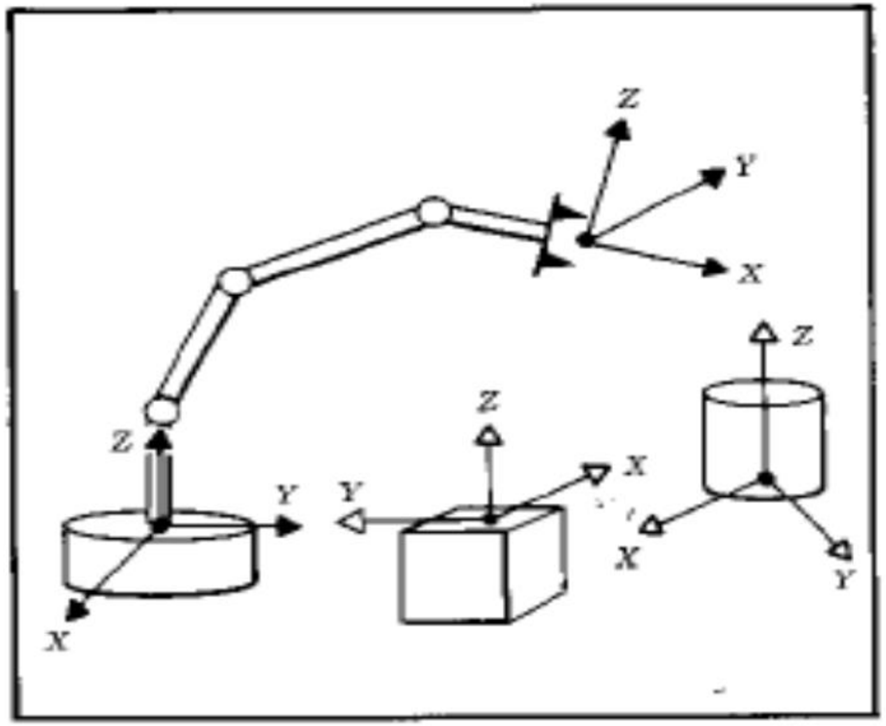
- **Kinematics:** Study of motion without regard to the forces/torques that cause it (i.e. without consideration of why the motion quantities have the values they do)
- Interested in position, velocities, accelerations, etc. of each joint and link
- Forward and inverse kinematics (FK & IK): Relationship between "joint's position" and "position and orientation of end-effector"





Kinematics – Position & Orientation Representation

- **How to locate** objects (e.g links of manipulator, Parts, Tools, etc) in three-dimensional space
  - ➢ **Frame:** a coordinate system rigidly attached to each object
  - ➢ How to describe position and orientation of one frame with respect to another frame



Dynamics

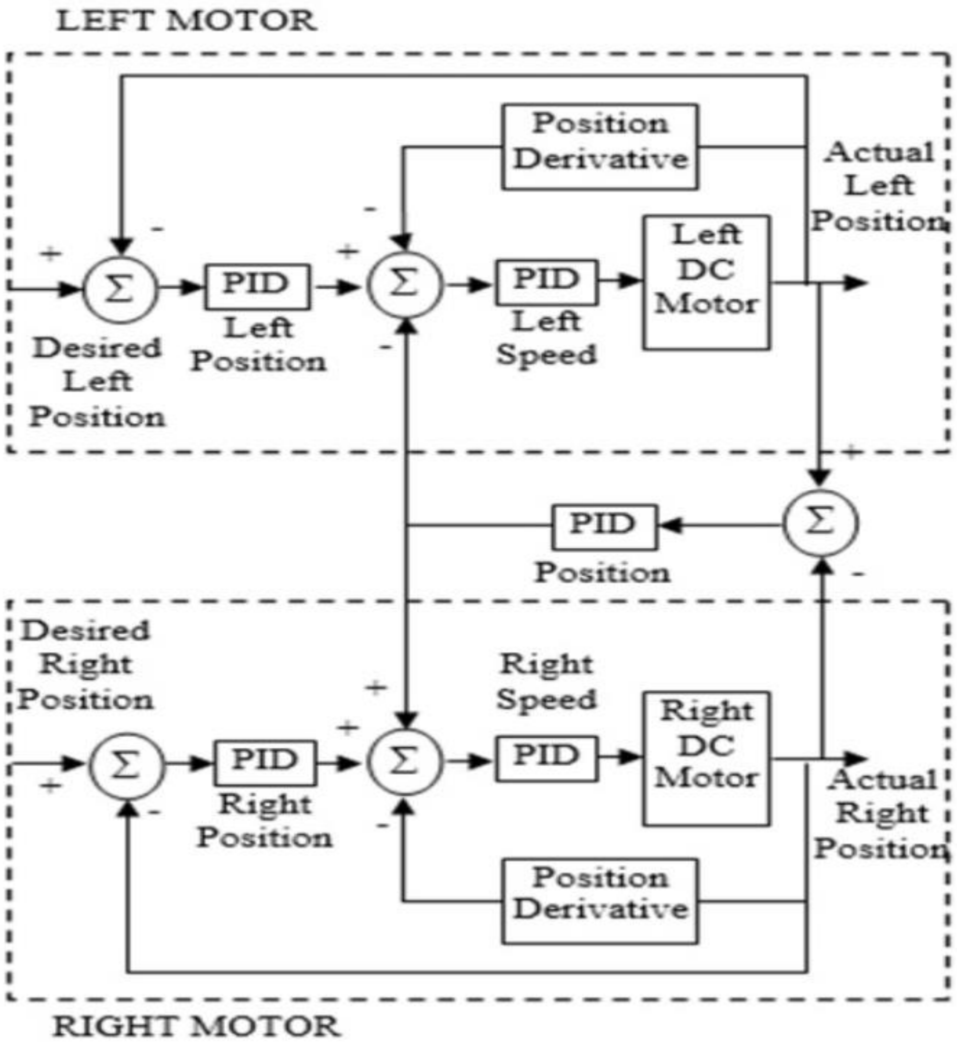
- **Study of the motion of bodies due to externally applied forces/torques** ($F = ma, \tau = m\alpha$)
- Robot dynamics is concerned with the relationship between the forces acting on a robot mechanism (some examples?) and the accelerations they produce
- **Why robot dynamics?** Robot Dynamic Equations are used for:
  a) Simulation of Robot Dynamics – It is a way of designing prototype robots and testing control strategies without the expenses of working with actual robots (Economical)
  b) Controller Design – Most control design techniques are based on the plant model. And Model-based controller is Superior to non-model-based control.

Recap on Wheeled & Bipedal robots!

- Ackermann Steering mobility model for wheeled robots can be easily adapted from those of automobiles or other wheeled vehicles (**Focus more on two-wheeled differential drive robot**)
- On the other hand, **understanding mobility models (walking) for bipedal robots is more challenging** as it involves two manipulators (the legs) working and coordinating together. The robot body weight (if significant enough) also has to be considered in the calculations
- **Stability is the key** to achieve successful locomotion for bipedal robots; robots are required to:
  - walk/run for long distances without falling,
  - maintain a stable posture when they come to a stop,
  - withstand any form of terrains

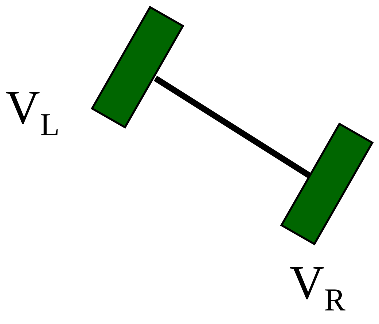
Chapter 2a:
For Wheeled Mobile Robots
(two-wheeled differential drive robot)

PID Control


LEFT MOTOR
Position Derivative
Actual Left Position
Left DC Motor
PID
PID
Desired Left Position
Left Position
Left Speed
PID
Position
Desired Right Position
Right Speed
Right DC Motor
PID
PID
Right Position
Actual Right Position
Position Derivative
RIGHT MOTOR

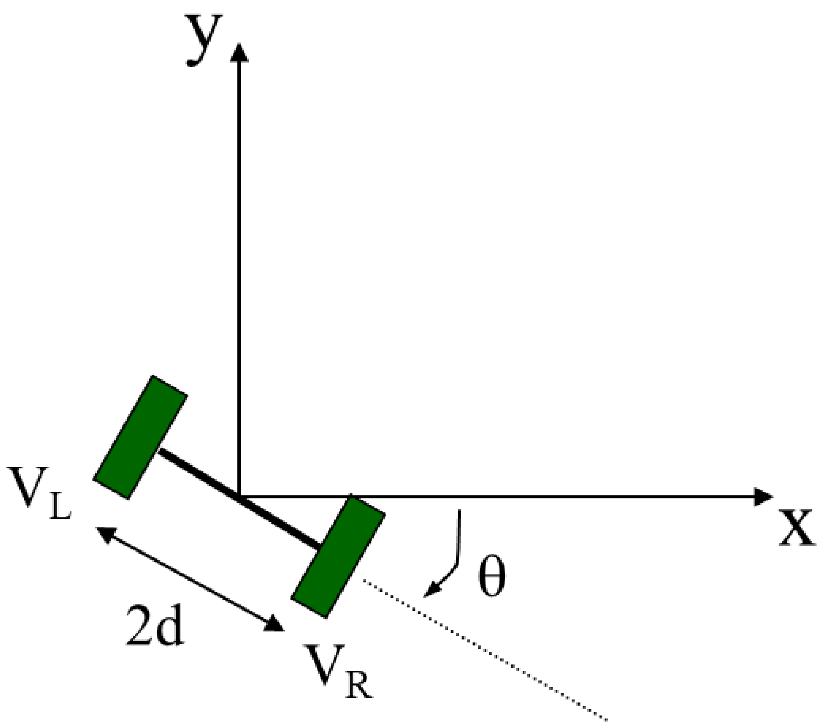
Forward Kinematics

- Difference in wheels' speeds determines the robot turning angle
- **Questions:**
  i. Given the wheel's velocities or positions, what is the robot's velocity/position ?
  ii. Are there any inherent system constraints?



- **Steps:**
  1. Specify system measurements
  2. Determine the point (the radius) around which the robot is turning
  3. Determine the speed at which the robot is turning to obtain the robot velocity
  4. Integrate to find position

Forward Kinematics



(assume a wheel radius of 1)

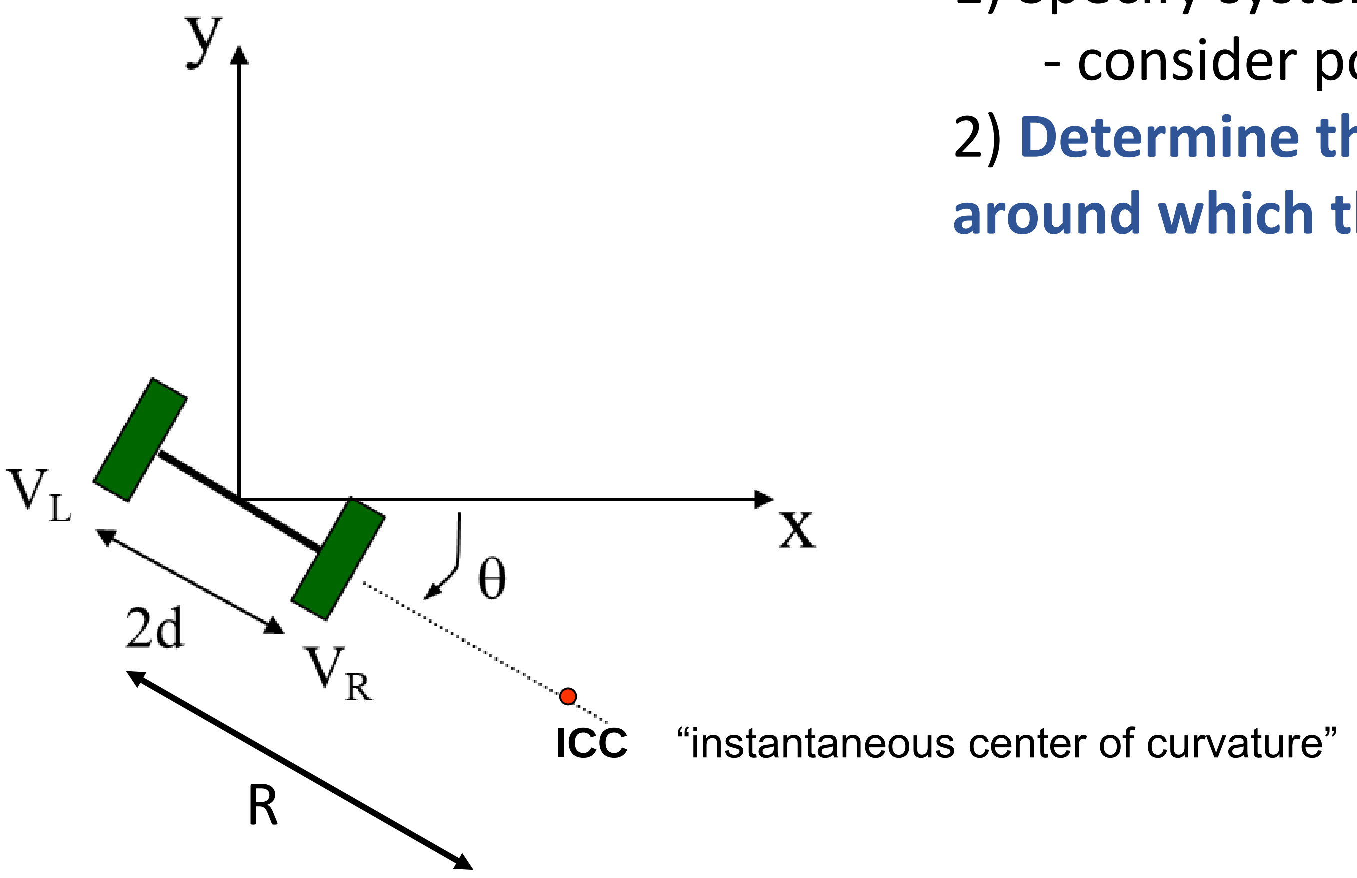
1) **Specify system measurements**

- consider possible coordinate systems

Forward Kinematics (Radius of Turning)

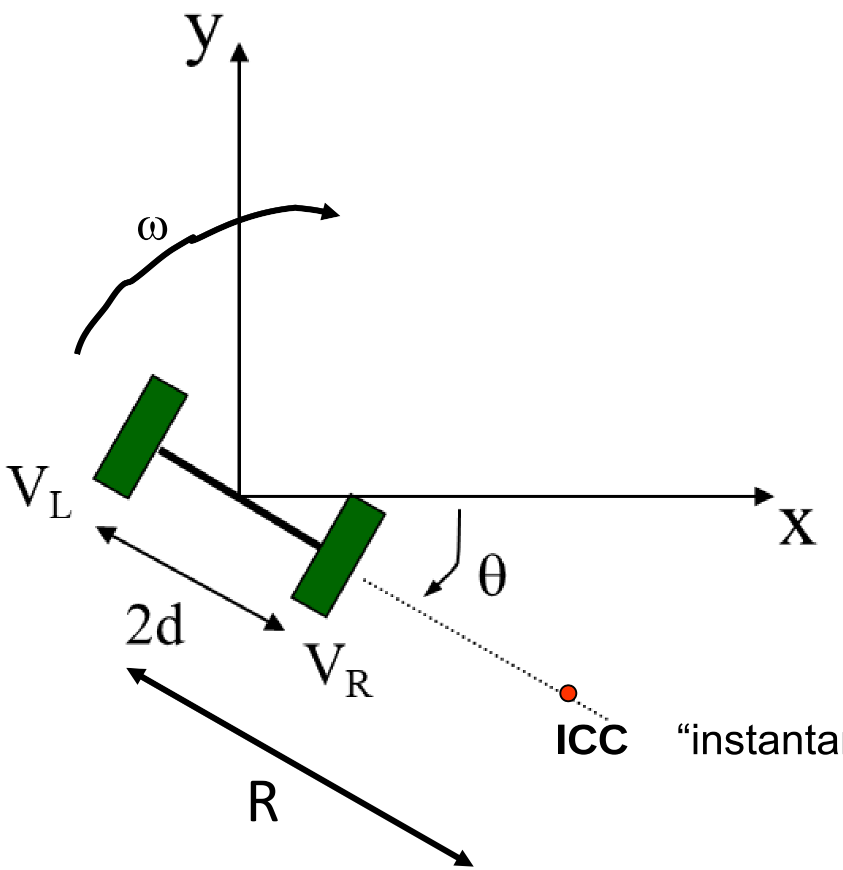
1) Specify system measurements
   - consider possible coordinate systems

2) **Determine the point (the radius) around which the robot is turning**



(assume a wheel radius of 1)

Forward Kinematics (Angular Velocity)



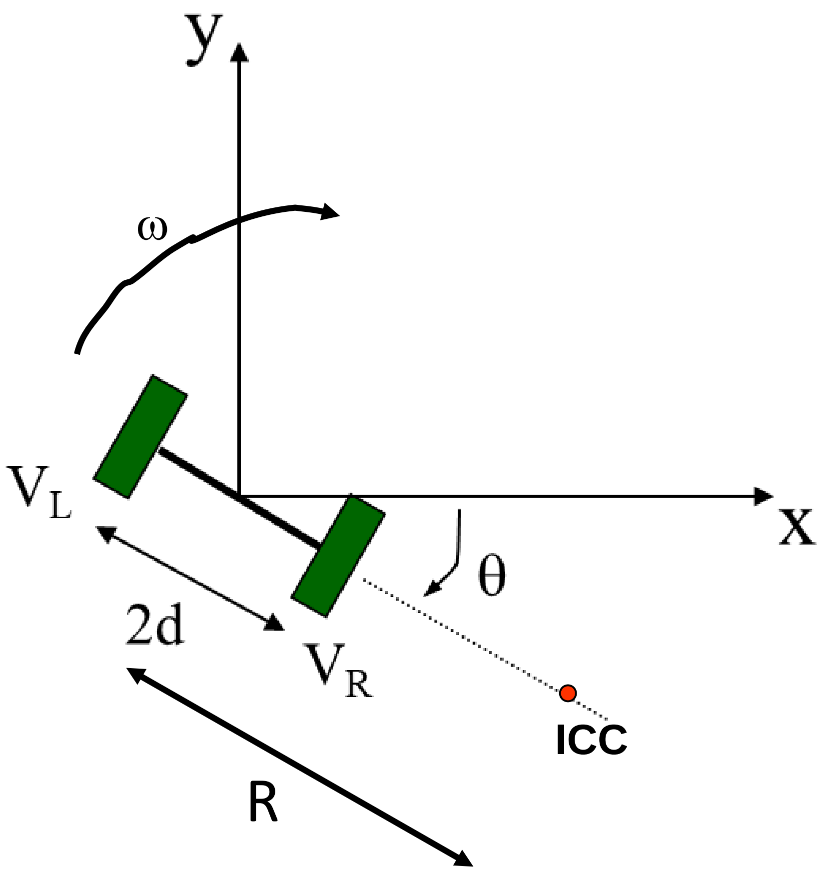
ICC "instantaneous center of curvature"

(assume a wheel radius of 1)

1) Specify system measurements
    - consider possible coordinate systems

2) Determine the point (the radius) around which the robot is turning
    - to minimize wheel slippage, this point (the ICC) must lie at the intersection of the wheels' axles
    - each wheel must be traveling at the **same angular velocity around the ICC**

Forward Kinematics (Angular Velocity)



(assume a wheel radius of 1)

3) **Determine the robot's speed around the ICC and its linear velocity**

$$\omega\,(R+d) = V_L$$

$$\omega\,(R-d) = V_R$$

Thus, solving the above equations will give you:

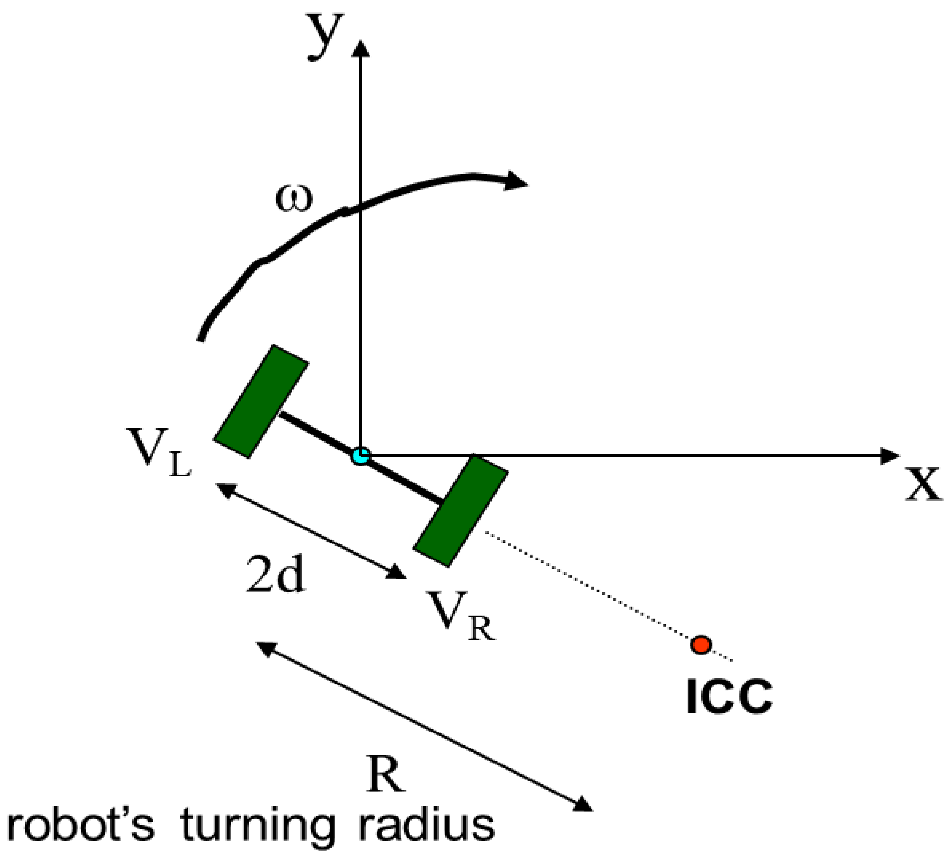
$$\omega = (V_R - V_L) / 2d$$

$$R = 2d\,(V_R + V_L) / (V_R - V_L)$$

The **robot's velocity** is:

$$V = \omega R = (V_R + V_L) / 2$$

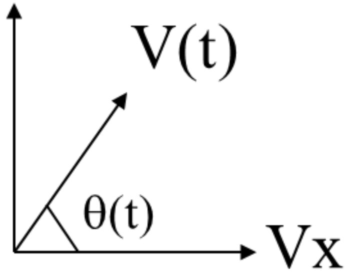
Forward Kinematics (Angular Velocity)



(assume a wheel radius of 1)

4) **Integrate to obtain position**

$$V_x = V(t)\cos(\theta(t))$$
$$V_y = V(t)\sin(\theta(t))$$



Thus,

$$x(t) = \int V(t)\cos\big(\theta(t)\big)\,dt$$

$$y(t) = \int V(t)\sin\big(\theta(t)\big)\,dt$$

$$\theta(t) = \int \omega(t)\,dt$$

Forward Kinematics (Angular Velocity)

Knowing the velocities and solving for equations involving ω and R, we can find the ICC location:

$$ICC = [x - R\,sin(\theta)\,,\, y + R\,cos(\theta)]$$

and at time t + δt, the robot's pose will be:

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} cos(\omega\delta t) & -sin(\omega\delta t) & 0 \\ sin(\omega\delta t) & cos(\omega\delta t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x - ICC_x \\ y - ICC_y \\ \theta \end{bmatrix} + \begin{bmatrix} ICC_x \\ ICC_y \\ \omega\delta t \end{bmatrix}$$

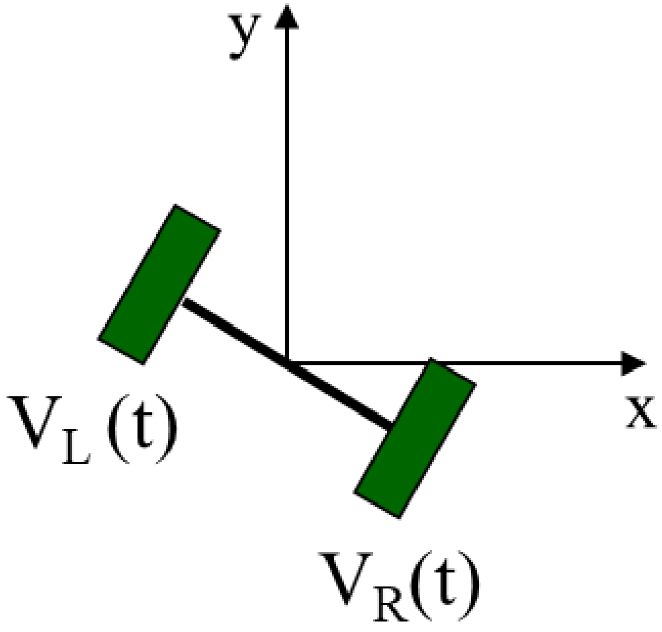
Transformed Point X, Y	Rotation by ωδt about Z-axis	Translate ICC to origin	Translate ICC to back to original location

This equation simply describes the motion of a robot rotating a distance R about its ICC with an angular velocity of ω

Inverse Kinematics (<u>the problem</u>)

Key question:

Given a desired position or velocity, what can we do to achieve it?



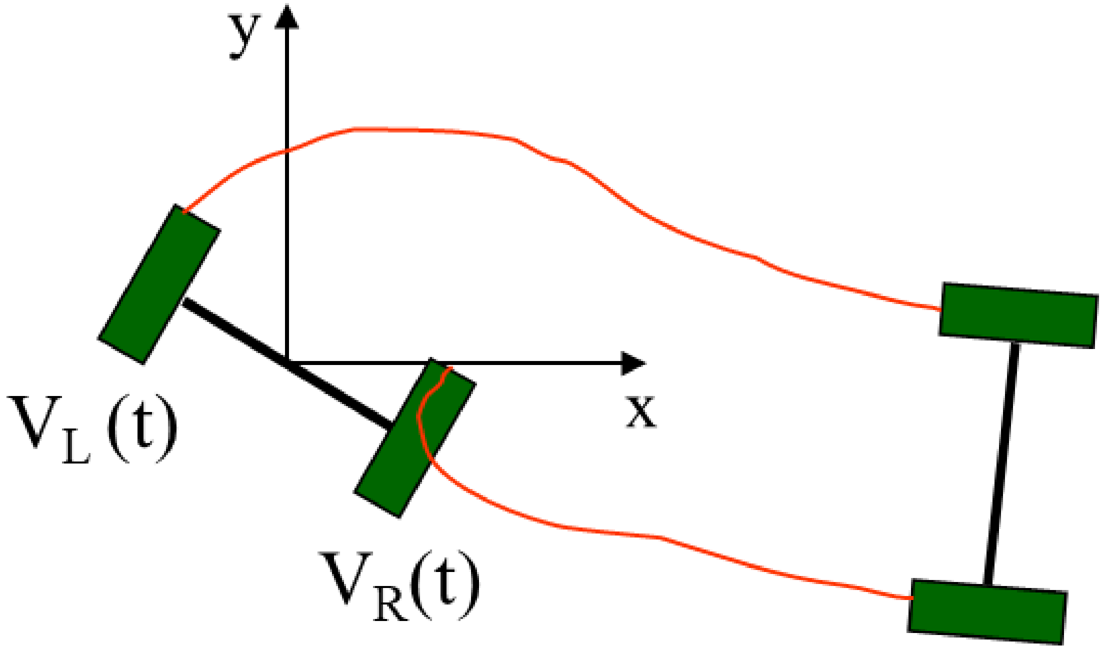
starting position

final position

Inverse Kinematics (one solution)

Key question:

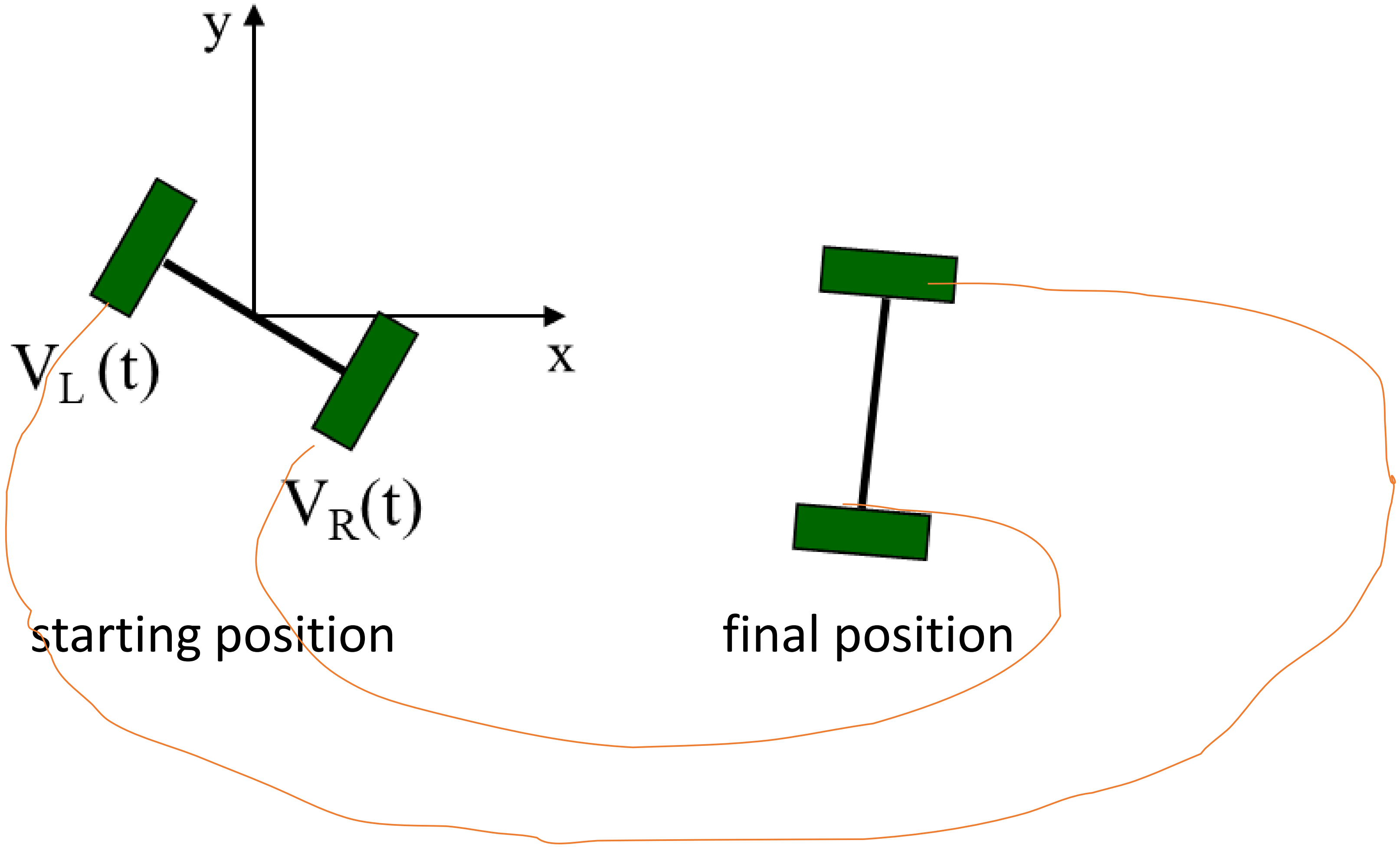
Given a desired position or velocity, what can we do to achieve it?



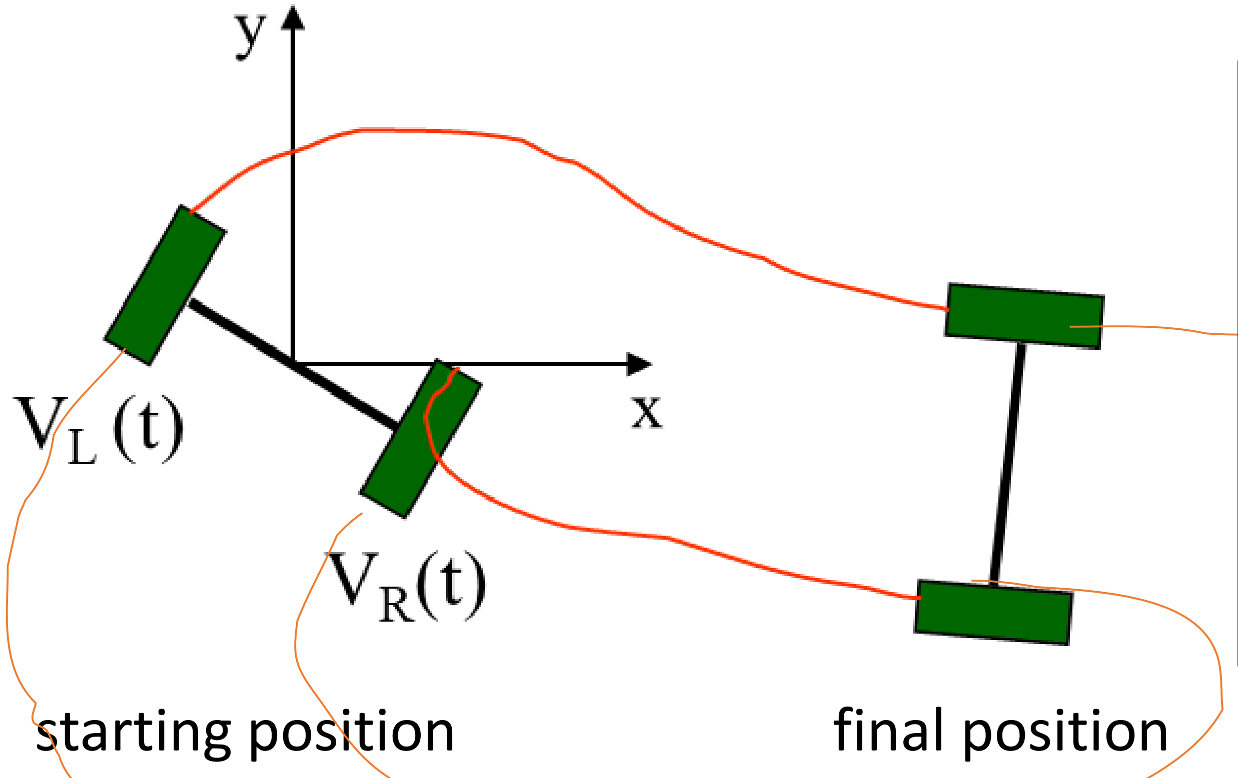
starting position

final position

Inverse Kinematics (another solution)



Inverse Kinematics (many solutions)

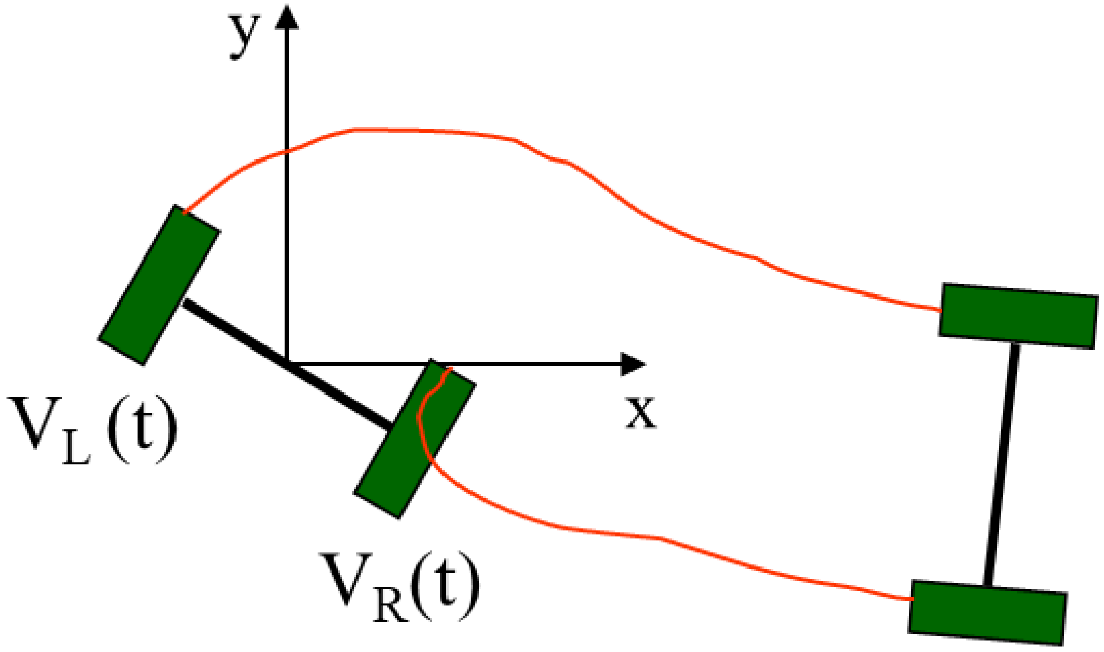


Need to solve these equations:

$$x = \int V(t)\cos(\theta(t))\,dt$$
$$y = \int V(t)\sin(\theta(t))\,dt$$
$$\theta = \int \omega(t)\,dt$$
$$\omega = (V_R - V_L) / 2d$$
$$V = \omega R = (V_R + V_L) / 2$$

for $V_L(t)$ and $V_R(t)$.

Inverse Kinematics (find the best solution)

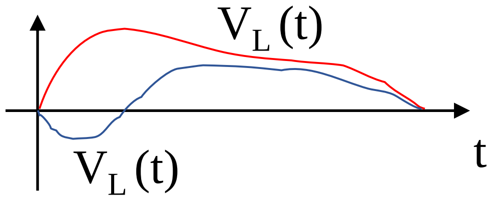


starting position

final position

Finding *some* solution is not hard, but finding the "best" solution is **very difficult...**

- quickest time
- most energy efficient
- smoothest velocity profiles

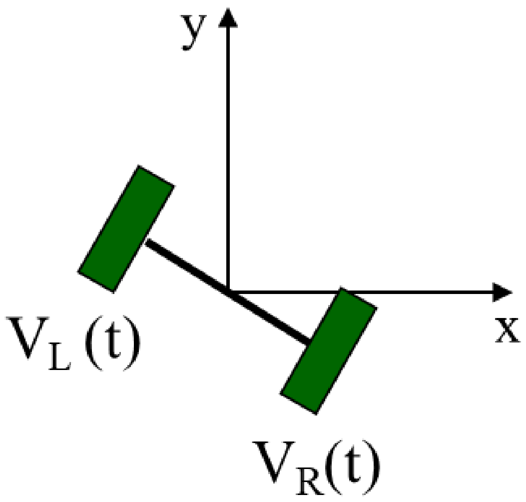


It all depends on who gets to define "best"...

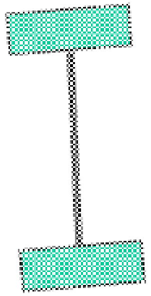
Inverse Kinematics (Decomposition)

Usual approach: decompose the problem and control only a few DOF at a time

Differential Drive



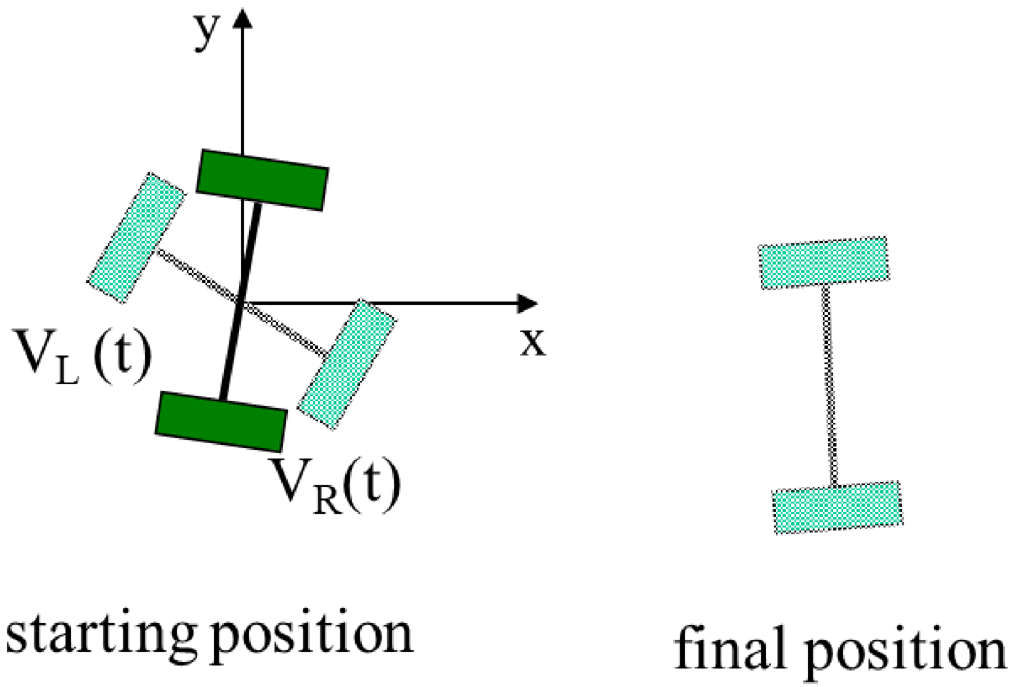
starting position



final position

Inverse Kinematics (Decomposition)

Usual approach: decompose the problem and control only a few DOF at a time



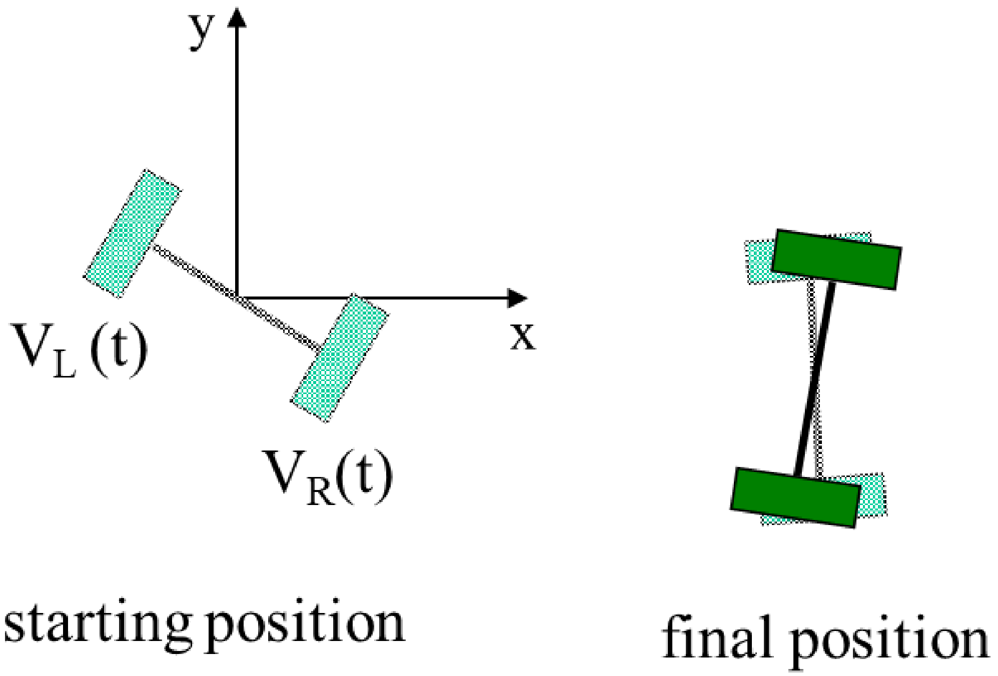
Differential Drive

(1) turn so that the wheels are parallel to the line between the original and final position of the robot origin

$$V_L(t) = -V_R(t) = V_{max}$$

Inverse Kinematics (Decomposition)

Usual approach: decompose the problem and control only a few DOF at a time



Differential Drive

(1) turn so that the wheels are parallel to the line between the original and final position of the robot origin

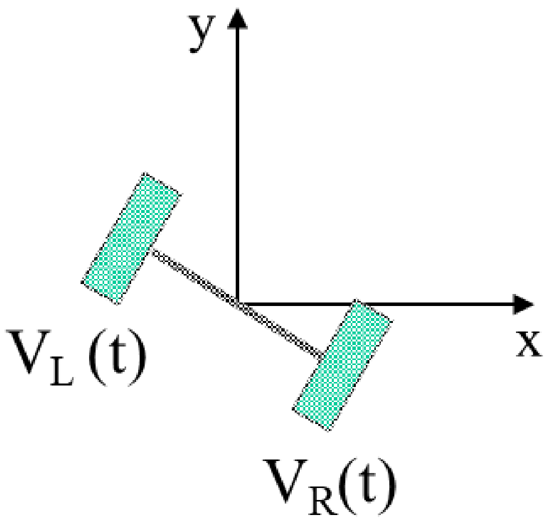
$$V_L(t) = -V_R(t) = V_{max}$$

(2) drive straight until the robot's origin coincides with the destination

$$V_L(t) = V_R(t) = V_{max}$$

Inverse Kinematics (Decomposition)

Usual approach: decompose the problem and control only a few DOF at a time



starting position

final position

Differential Drive

(1) turn so that the wheels are parallel to the line between the original and final position of the robot origin

$$V_L(t) = -V_R(t) = V_{max}$$

(2) drive straight until the robot's origin coincides with the destination

$$V_L(t) = V_R(t) = V_{max}$$

(3) rotate again in order to achieve the desired final orientation

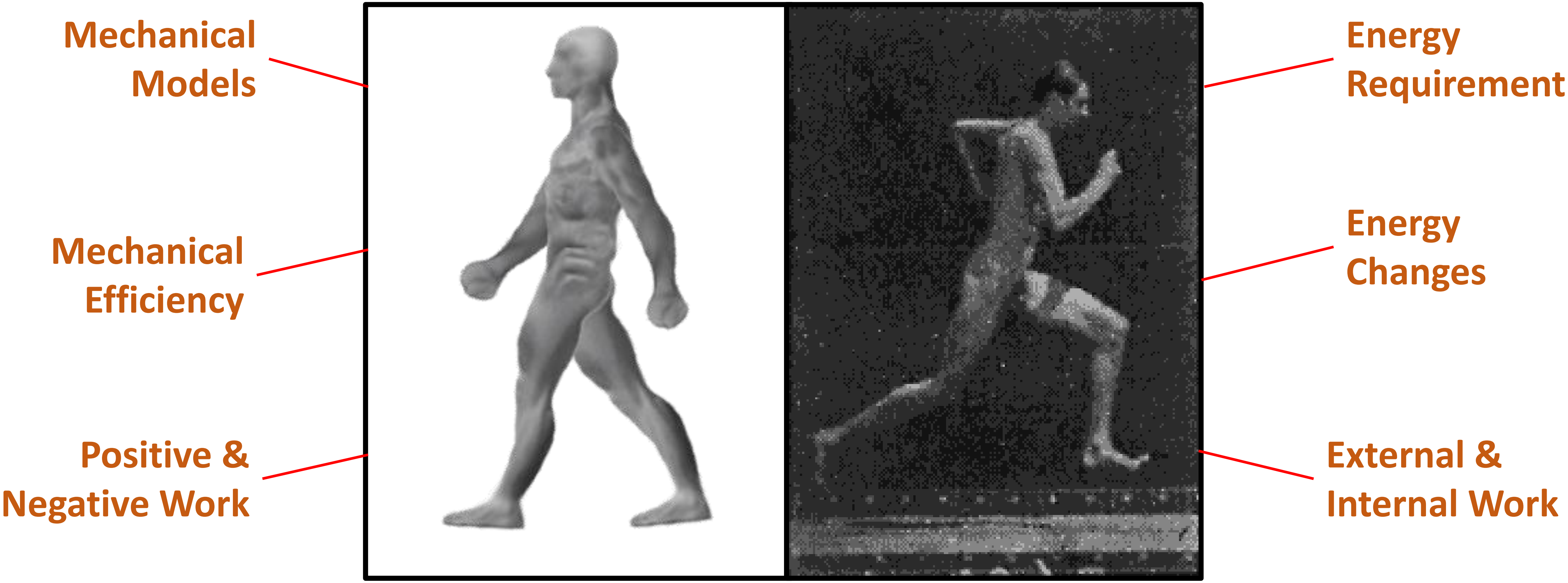
$$-V_L(t) = V_R(t) = V_{max}$$

Chapter 2b:
For Bipedal Robots

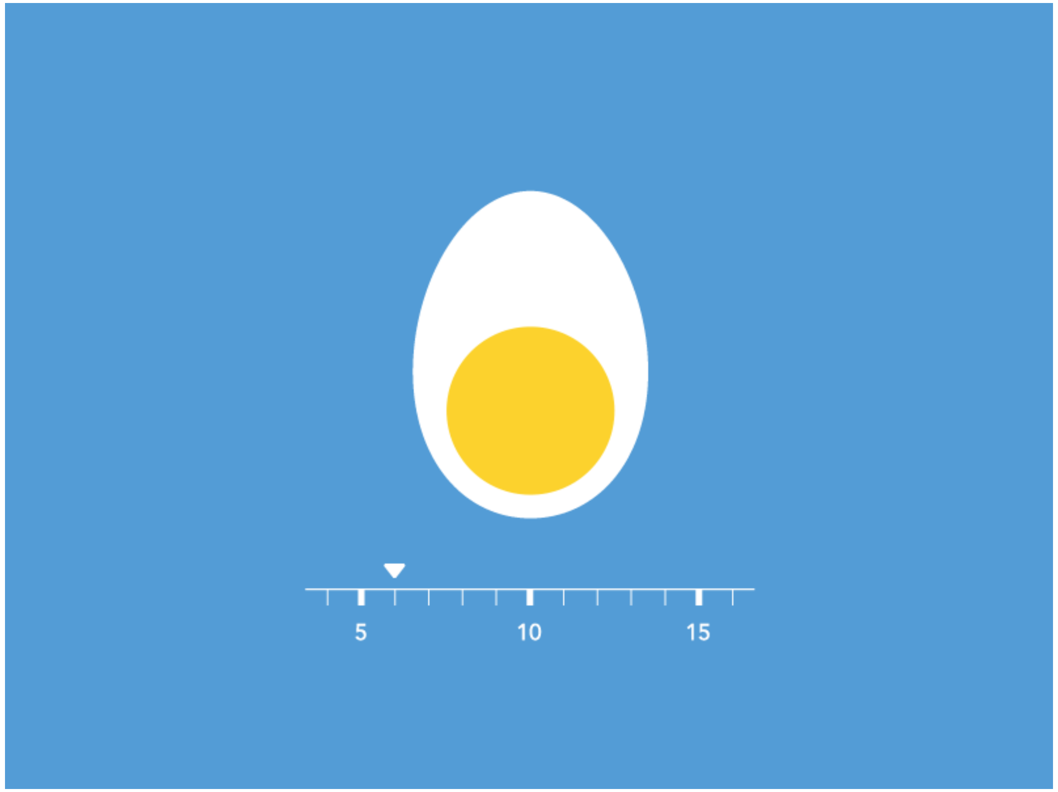
Why Study Biomechanics of Human Locomotion?

- **Human locomotion is naturally perfect**
- Understanding the biomechanics of human locomotion can **help us in our research/development on biped robots**, especially on their locomotion
- Hence, this allows us to design and construct biped robots to a near-perfection performance level, almost comparable to human locomotion

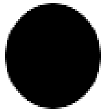
Biomechanics of Human Locomotion



(a) Walking vs (b) Running Mechanical Models

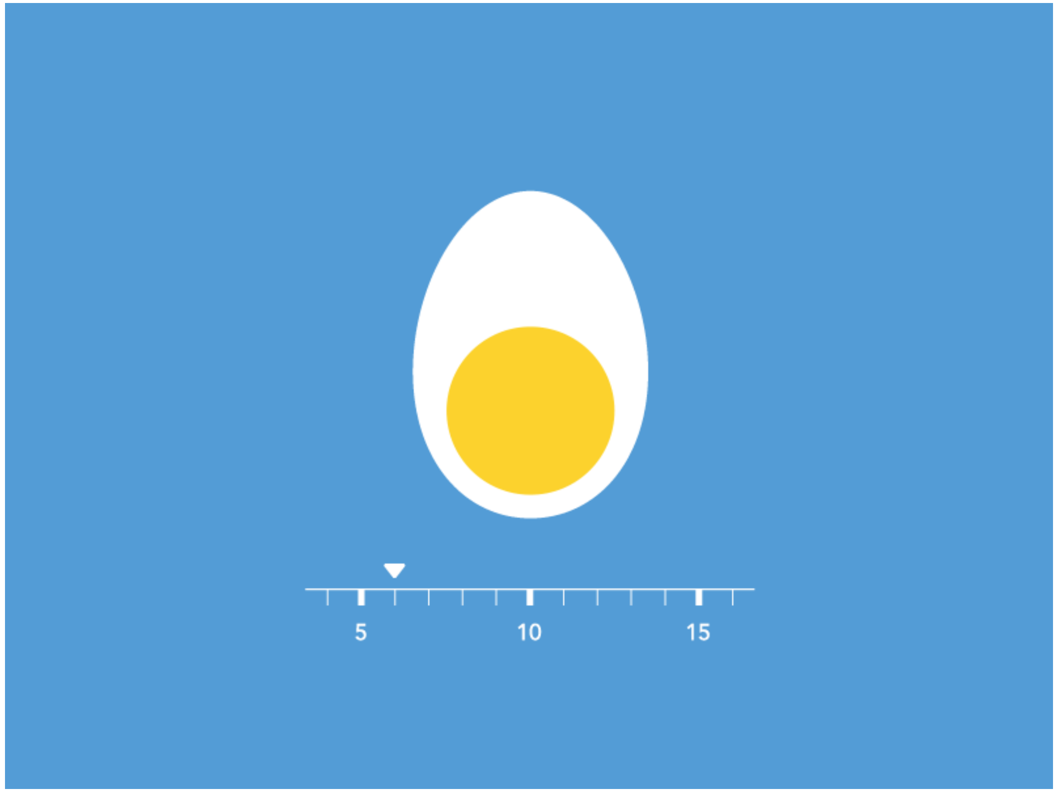


Walking – Rolling Egg



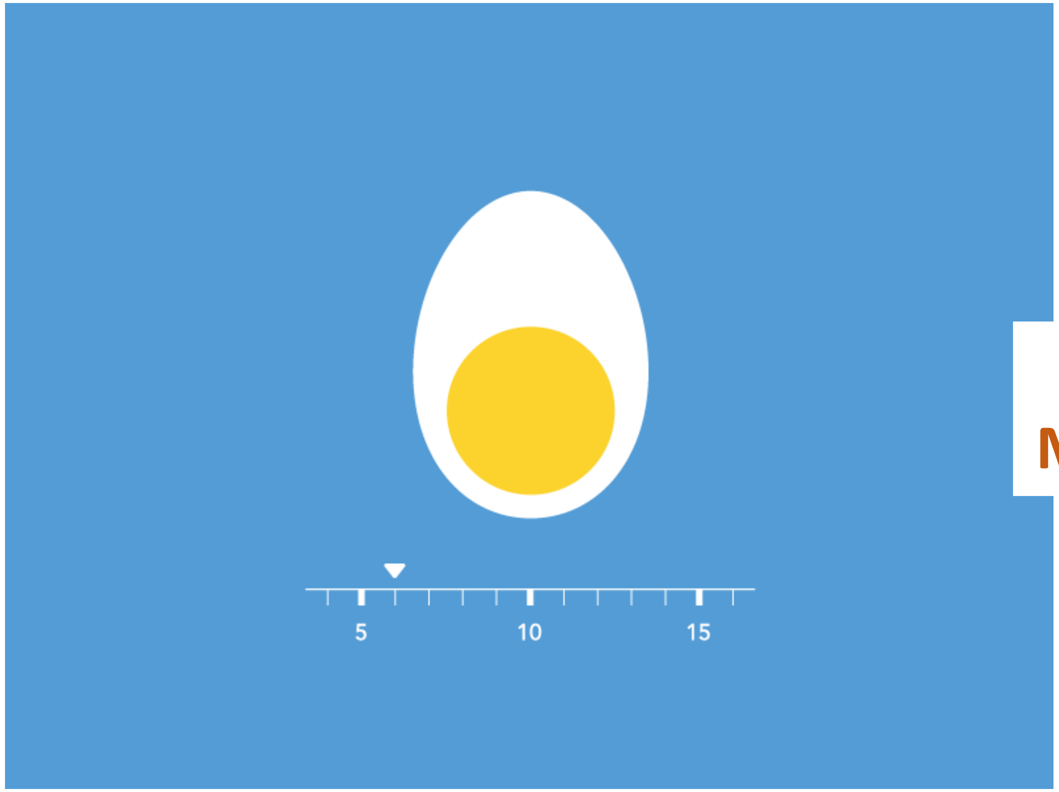
Running – Elastic Bouncing ball

(a) Walking Mechanical Model



Walking – Rolling Egg

(a) Walking Mechanical Model



Walking – Rolling Egg

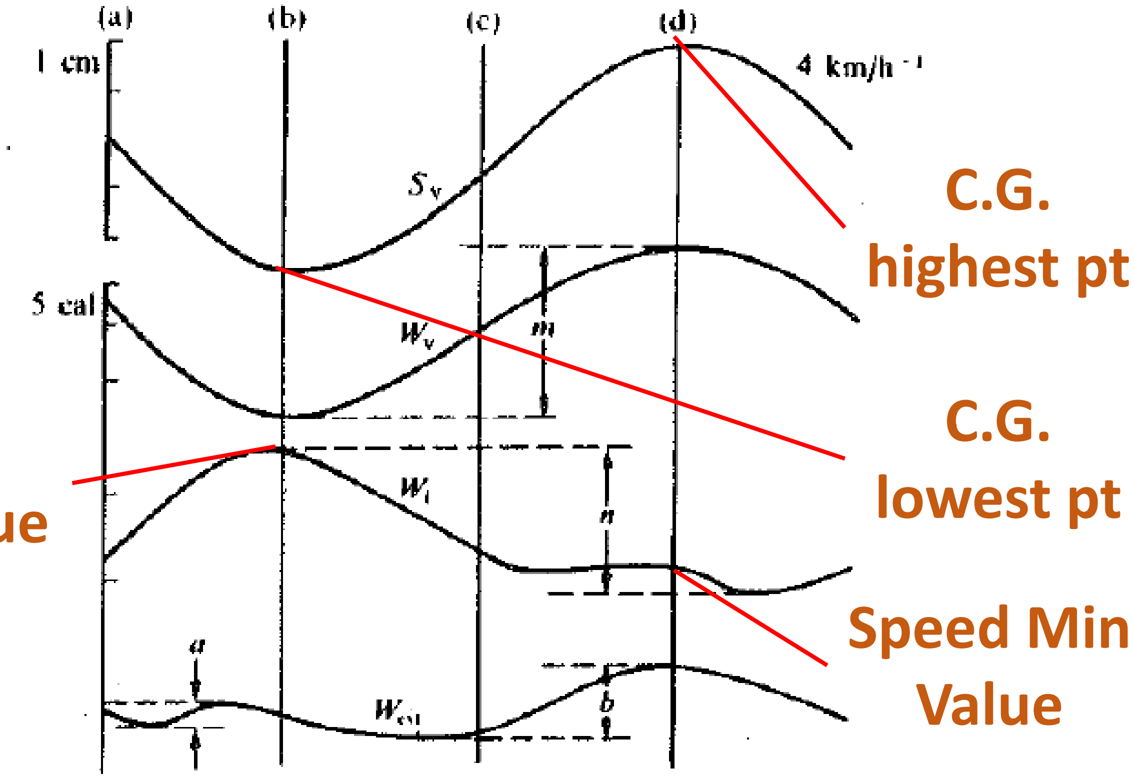
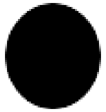
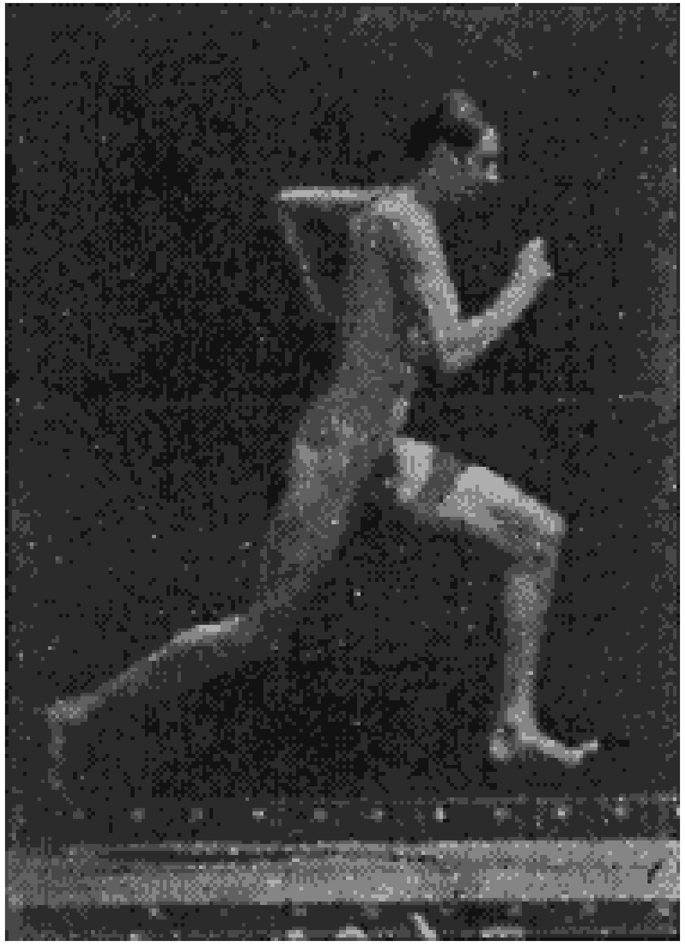


Fig. 3.13. Potential (W_V) and kinetic (W_F) energy changes during a step, walking at a speed of 4·0 km h^{-1}. W_V has been calculated from the vertical displacements of the centre of gravity of the body S_V; W_F from the speed changes in the direction of progression. W_{ext} is the sum of the two curves W_V and W_F; m is the antigravitational work, n the work due to frontal speed changes; $a + b$ is the total external work. (From Cavagna *et al.* 1963.)

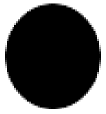
(b) Running Mechanical Model



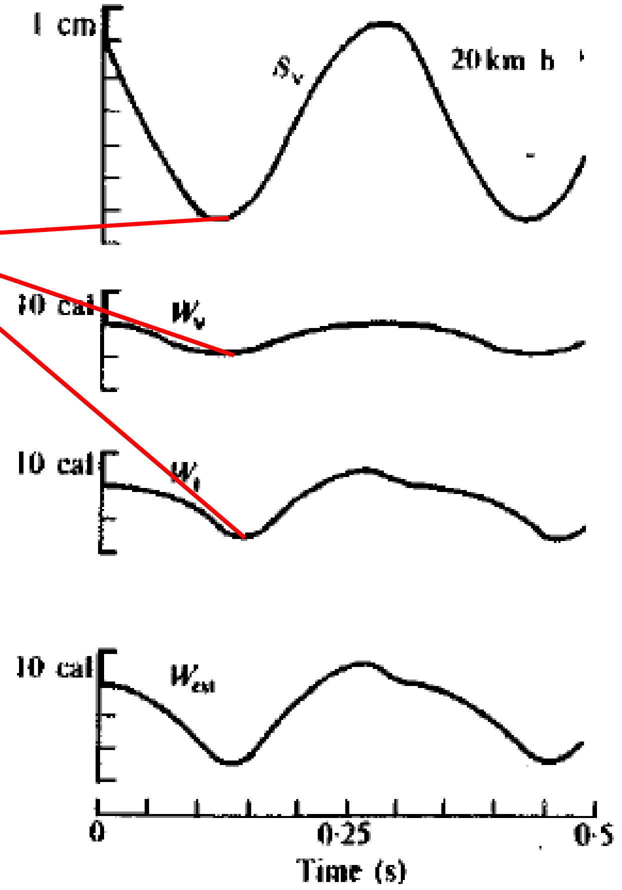


Running – Elastic Bouncing ball

(b) Running Mechanical Model



When the foot strikes the ground



Running – Elastic Bouncing ball

Fig. 3.17. Vertical displacement of the centre of gravity S_V, and changes in potential energy W_V, kinetic energy W_F and total energy W_{ext} during a single step when running at 20 km h^{-1}. (From Cavagna *et al.* 1964.)

Phases of P.E. & K.E. Changes in Walking & Running

Heel touches ground

When K.E. changes to P.E.

When P.E. & K.E.

When elastic energy changes to P.E. & K.E.

Heel

When P.E. & K.E. decreases

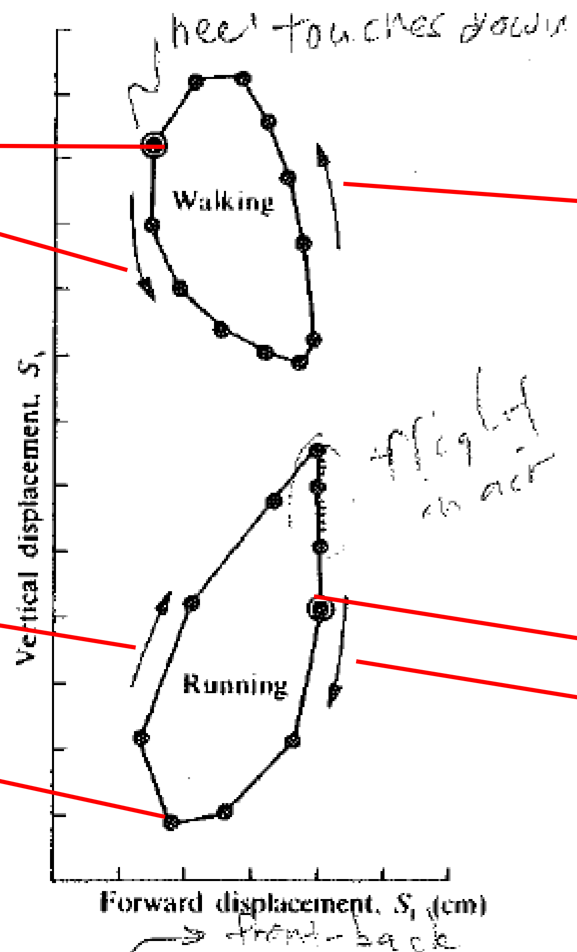
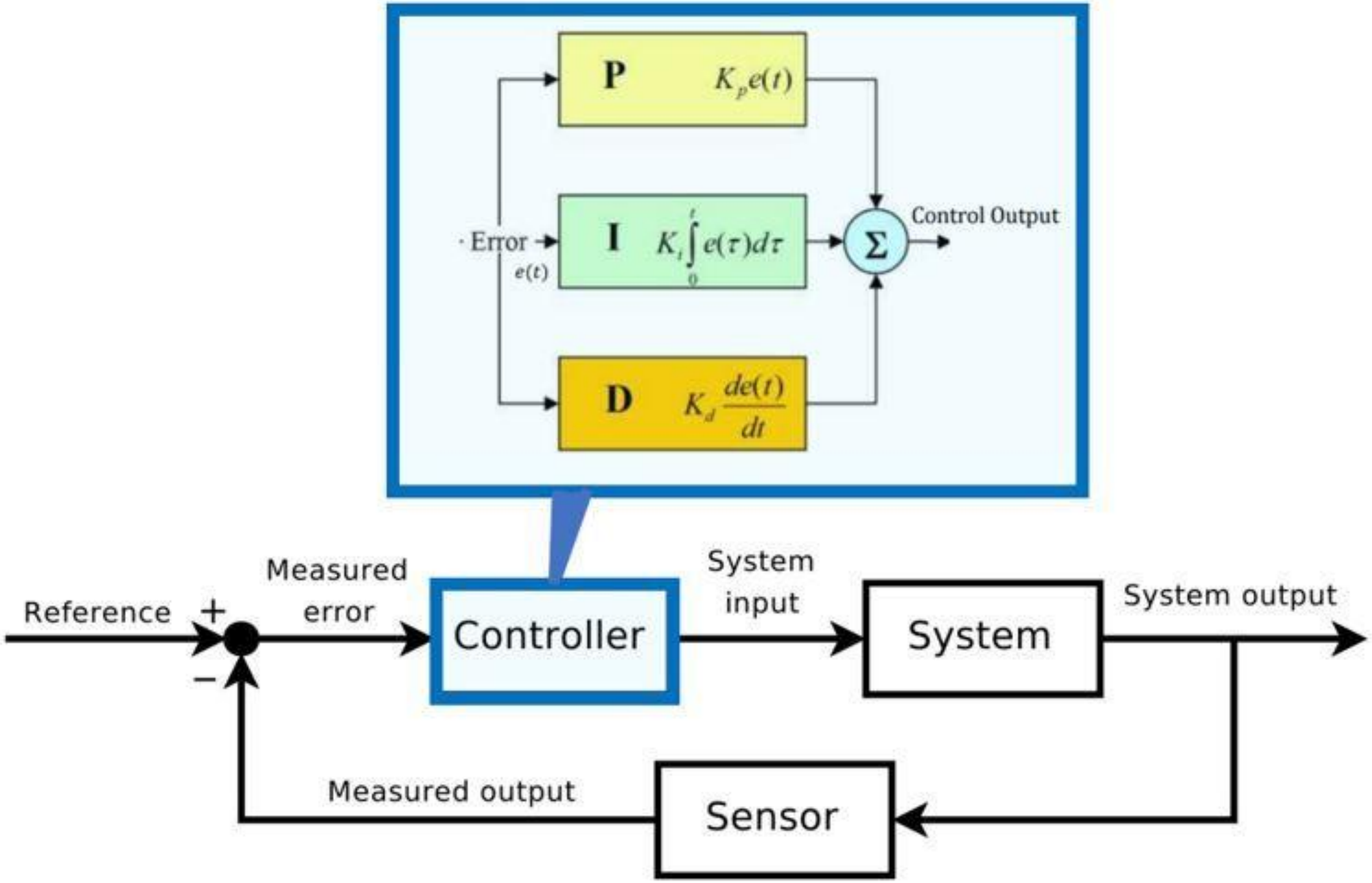


Fig. 3.20. Vertical S_V and frontal S_F displacement of the centre of gravity of the body in a subject walking or running on a treadmill. All the points are at 1/30 s interval. The larger point on each contour marks the moment at which heel touches the ground. In two dotted intervals of the contour for running the subject has no contact with the ground. (From Cavagna *et al.* 1964.)

Various Bipedal Robots Mobility Models

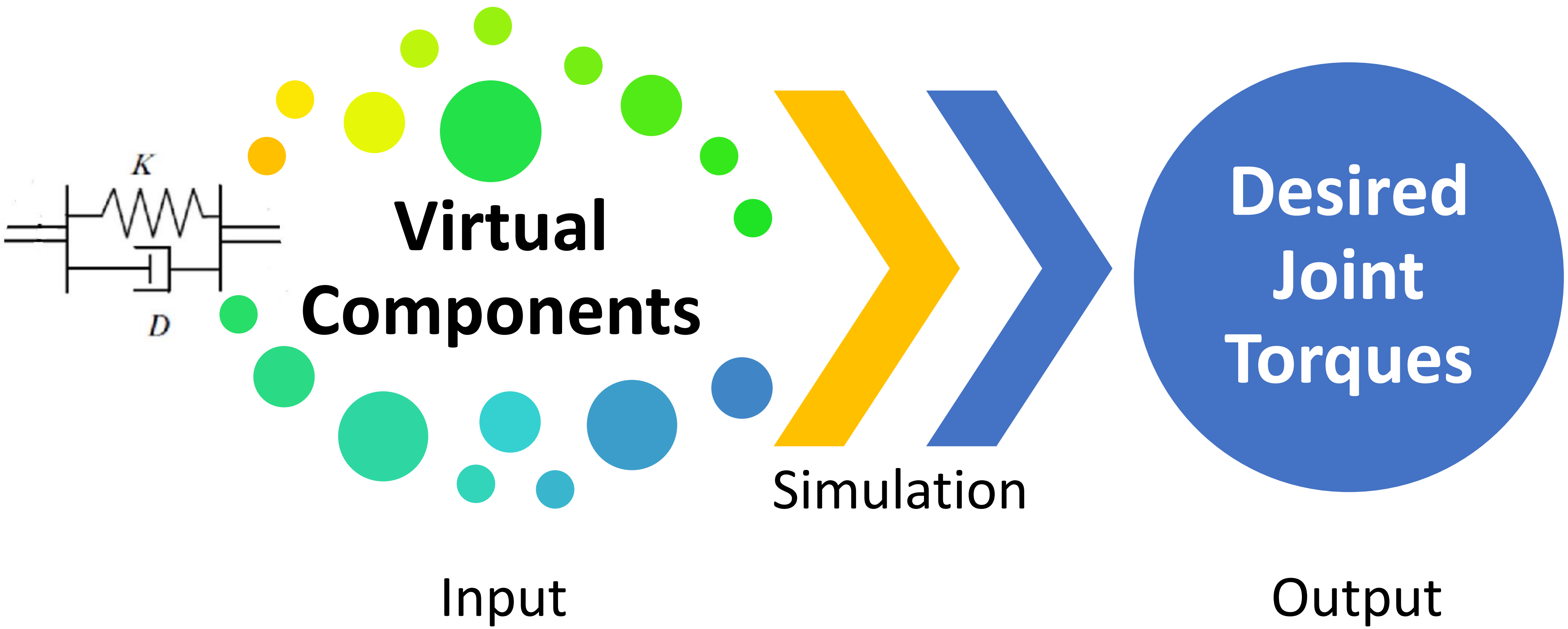
1. **PID Control**
2. **Virtual Model Control**
3. **3D-LIPM (Three-Dimensional Linear Inverted Pendulum Mode)**
4. **Reinforcement learning**

1. PID Control



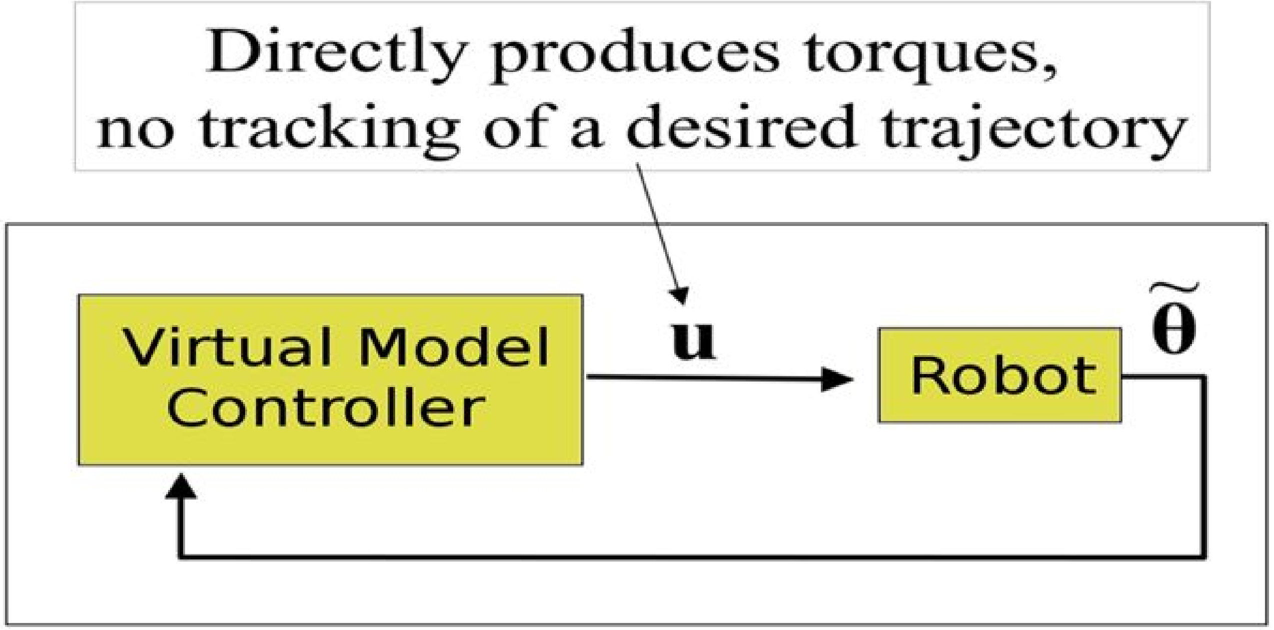
2. Virtual Model Control

Overview of Virtual Model Control (VMC)



2. Virtual Model Control

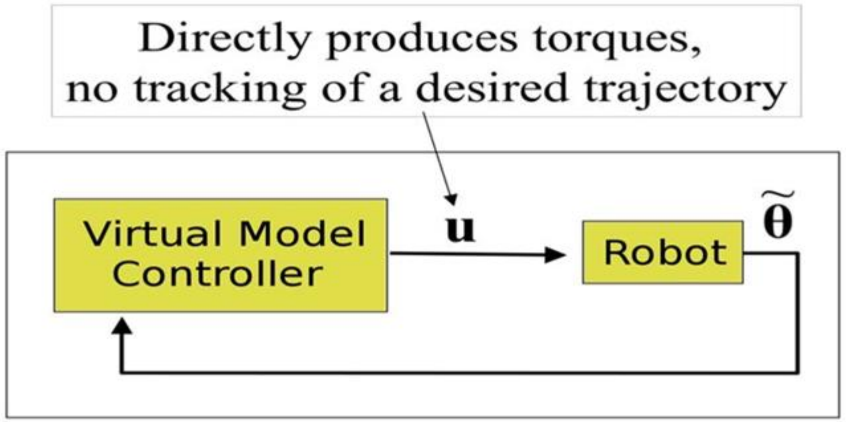
Overview of Virtual Model Control (VMC)



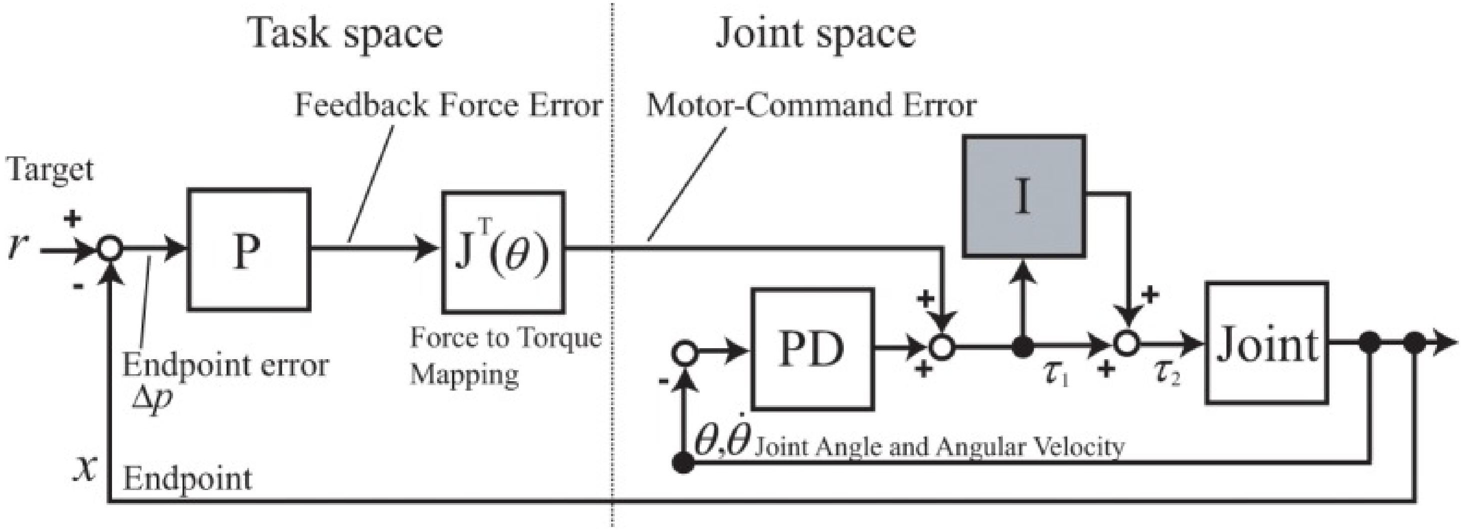
$\tilde{\theta}$	Actual robot posture
$\mathbf{u}$	Command (torque)

2. Virtual Model Control

VMC vs Proportional-Derivative (PD) Control



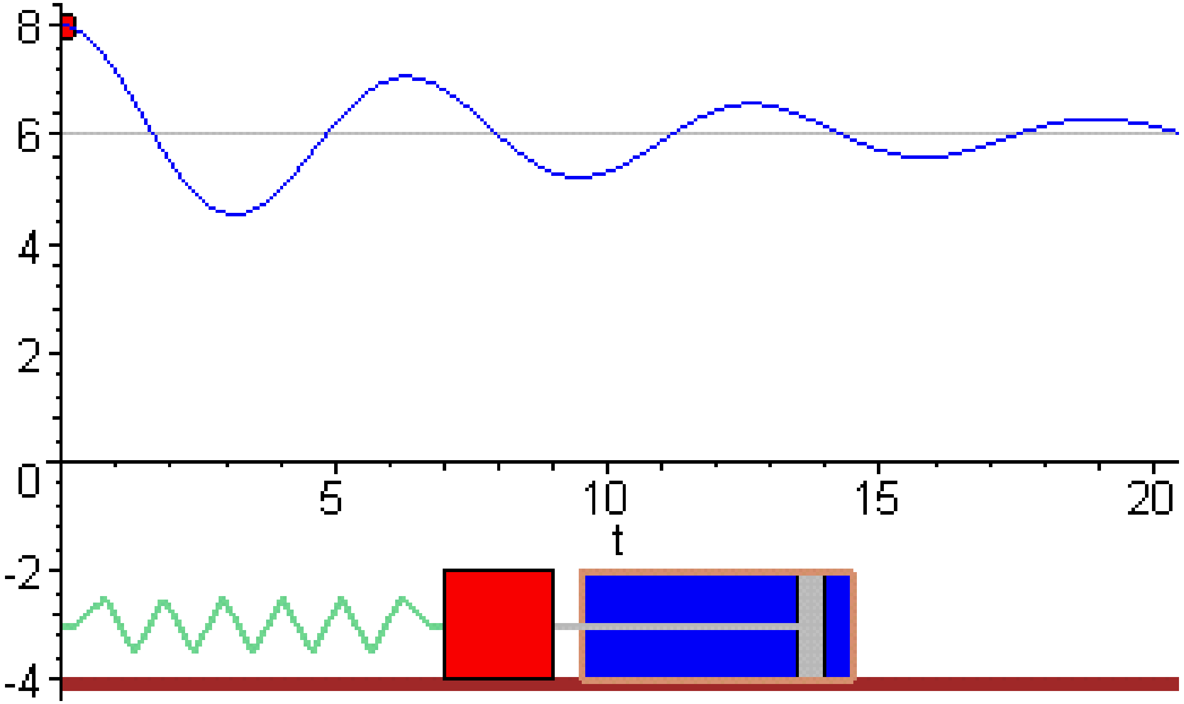
Virtual Model Control



PD Control

2. Virtual Model Control

Example



Attach virtual mass to robot's hand via virtual spring-damper

Virtual Model Application for Bipeds

(i) Single-Leg (Single support)

$$\begin{bmatrix}\tau_k\\ \tau_h\end{bmatrix} = \begin{bmatrix} \frac{-L_1 L_2\, s_k}{L_1\, c_a - \cdots} & \frac{-L_1\, c_a}{\cdots} \\ 0 & 1 \end{bmatrix}\begin{bmatrix} f \\ f_\theta \end{bmatrix} \quad (7)$$

Reference Frame {O}

Required joint torques

Frame {A} & {B} connected by a virtual component

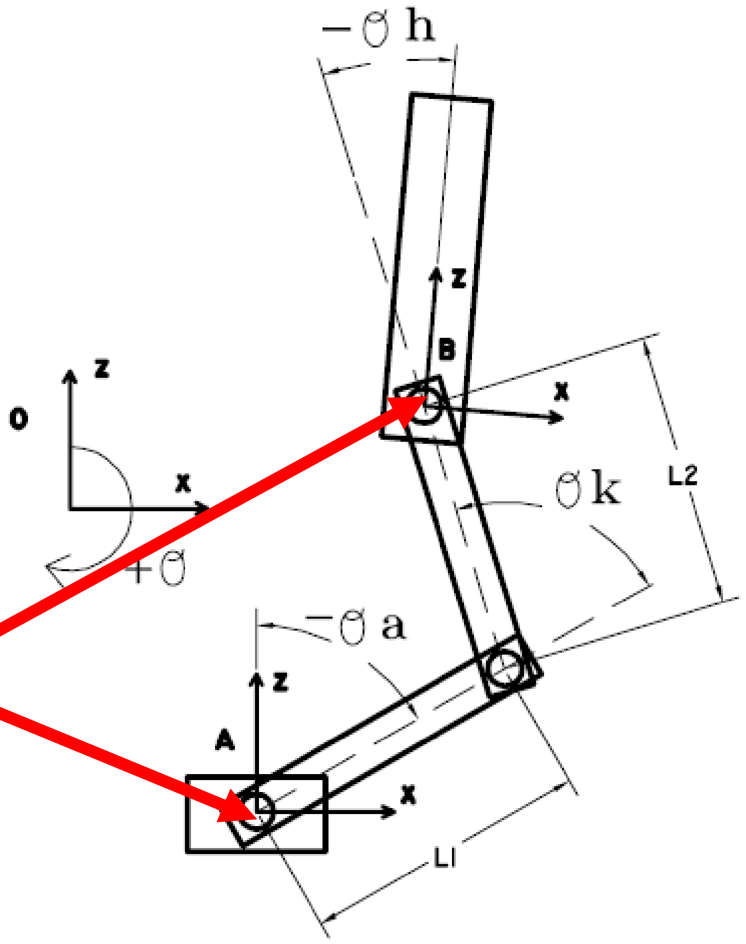


Fig. 3. Single-leg implementation. Reaction frame $\{A\}$ is assumed to be in the same orientation as reference frame $\{O\}$ so that ${}^{O}_{A}R = I$.

Virtual Model Application for Bipeds

(ii) Dual-Leg (Double support)

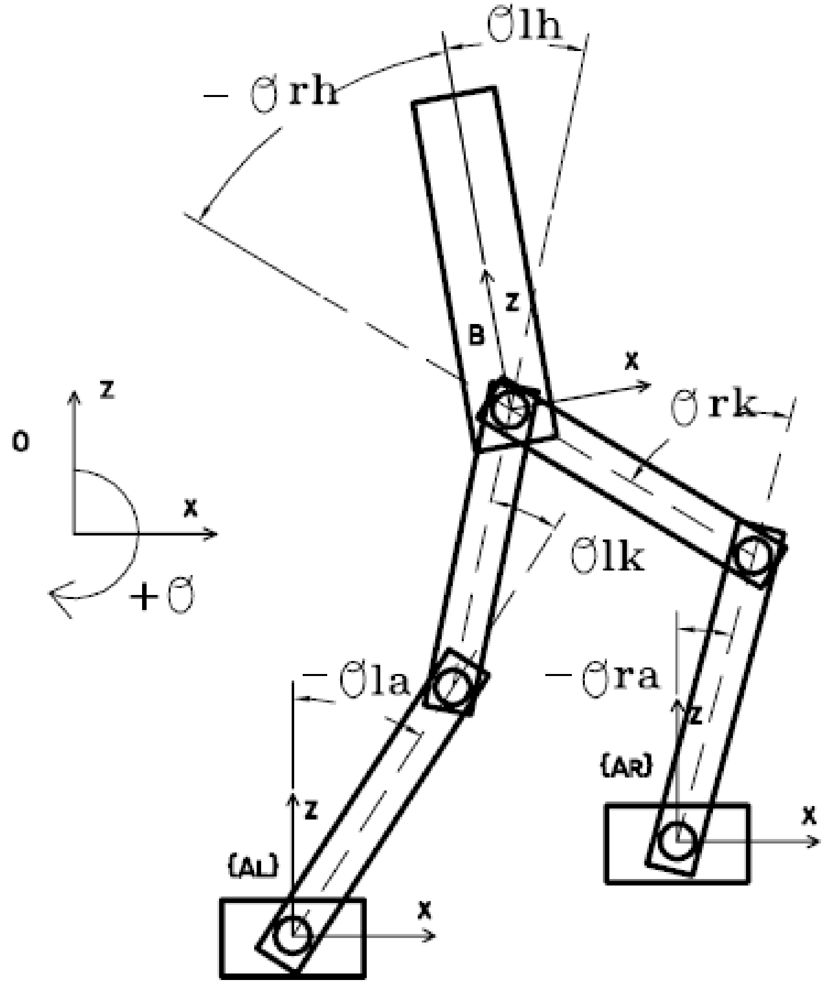


Fig. 4. Dual-leg example. Reaction frames $\{A_l\}$ and $\{A_r\}$ are assumed to be in the same orientation as reference frame $\{O\}$ so that ${}^{O}_{A_l}R = {}^{O}_{A_r}R = I$.

Virtual Model Application for Bipeds

(ii) Dual-Leg (Double support); Cont

$$\begin{bmatrix} \tau_{lk} \\ \tau_{lh} \\ \tau_{rk} \\ \tau_{rh} \end{bmatrix} = \begin{bmatrix} \frac{C\,V}{E} & \frac{D\,V}{E} & \frac{-V-QD+RC}{2\,E} - \frac{1}{2} \\ 0 & 0 & -1/2 \\ \frac{-A\,W}{E} & \frac{-B\,W}{E} & \frac{W+SB-TA}{2\,E} - \frac{1}{2} \\ 0 & 0 & -1/2 \end{bmatrix} \begin{bmatrix} f_x \\ f_z \\ f_\theta \end{bmatrix}, \tag{14}$$

where

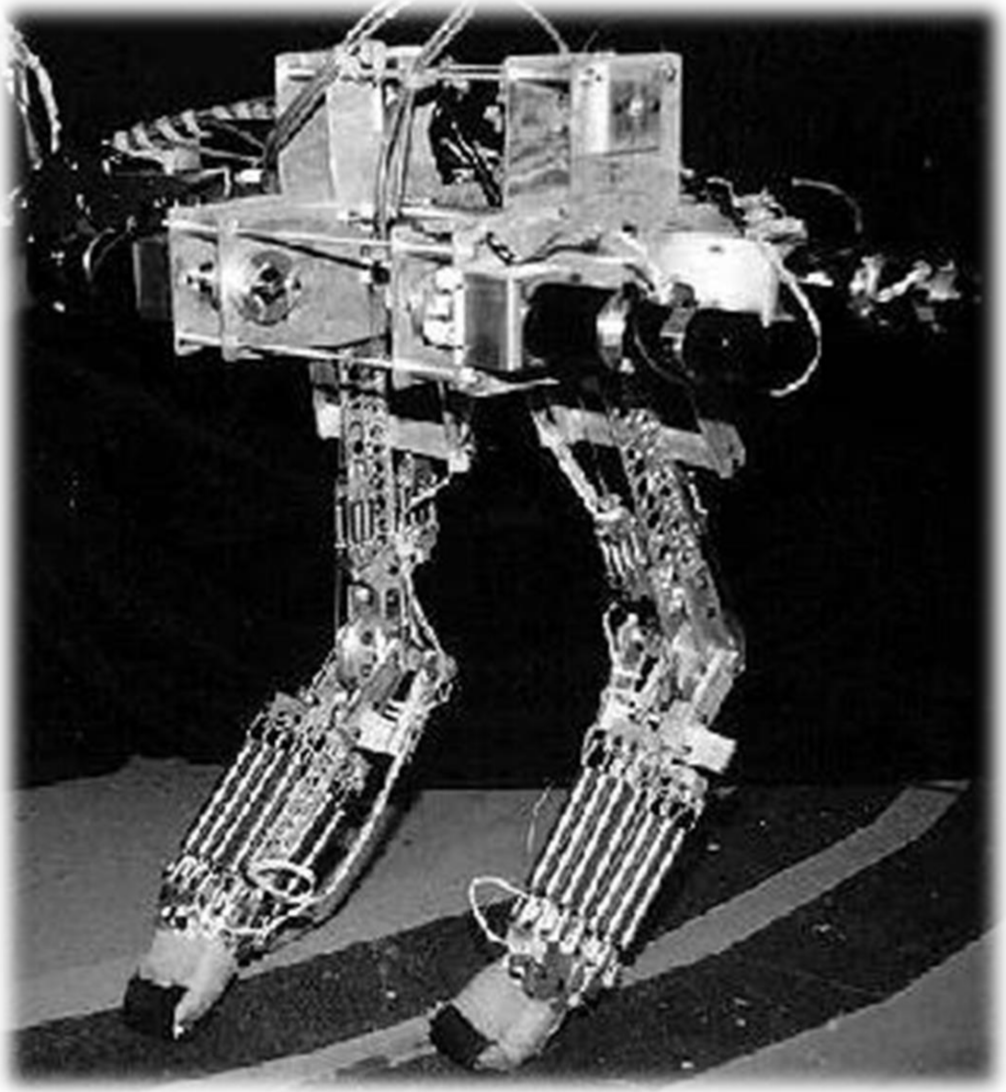
$$E = CB - AD \qquad A = -L_1 \cos(\theta_{la}) - L_2 \cos(\theta_{la} + \theta_{lk}) \qquad D = -L_1 \sin(\theta_{ra}) - L_2 \sin(\theta_{ra} + \theta_{rk})$$

$$V = QB - RA = -L_1 L_2 \sin(\theta_{lk}) \qquad B = -L_1 \sin(\theta_{la}) - L_2 \sin(\theta_{la} + \theta_{lk}) \qquad Q = -L_2 \cos(\theta_{la} + \theta_{lk}), \quad R = -L_2 \sin(\theta_{la} + \theta_{lk})$$

$$W = SD - TC = -L_1 L_2 \sin(\theta_{rk}). \qquad C = -L_1 \cos(\theta_{ra}) - L_2 \cos(\theta_{ra} + \theta_{rk}) \qquad S = -L_2 \cos(\theta_{ra} + \theta_{rk}), \quad T = -L_2 \sin(\theta_{ra} + \theta_{rk}).$$

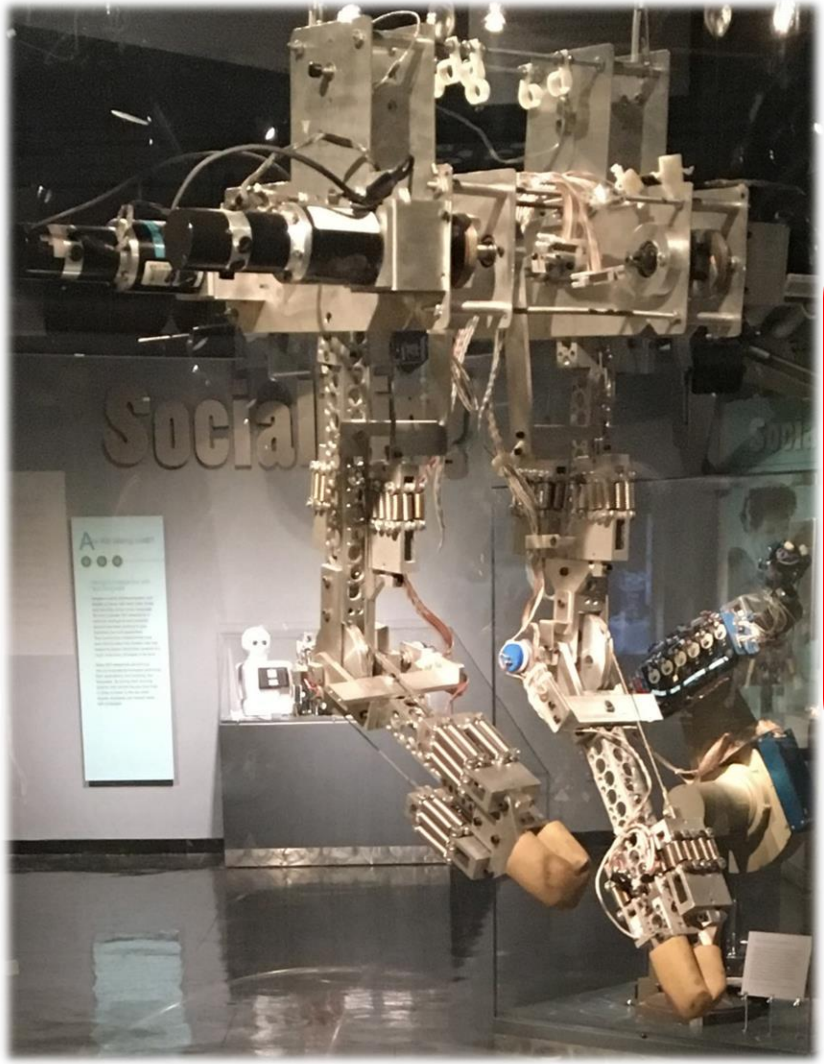
Introducing Spring Turkey

60cm

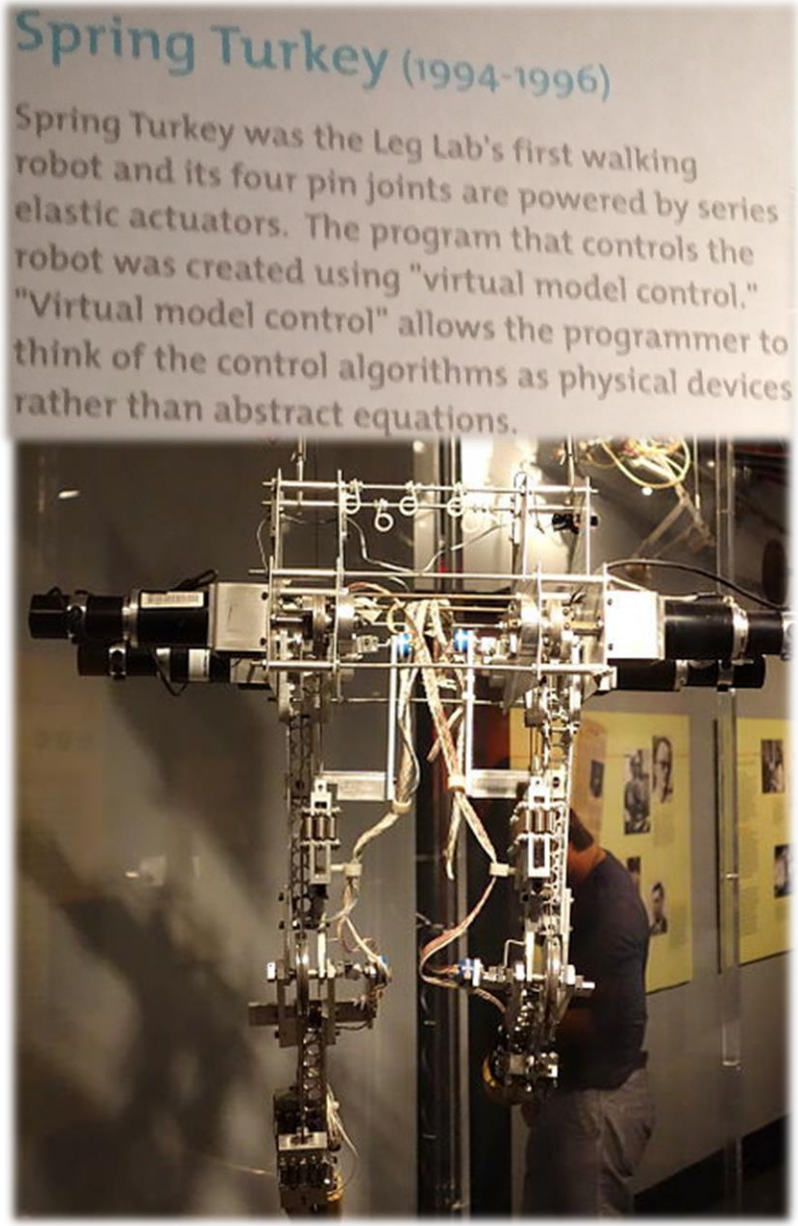


10kg

Introducing Spring Turkey



Mode of "*Virtual Model Control*" Application for Level-Ground Walking



Level-Ground Walking (Spring Turkey)

Virtual Components

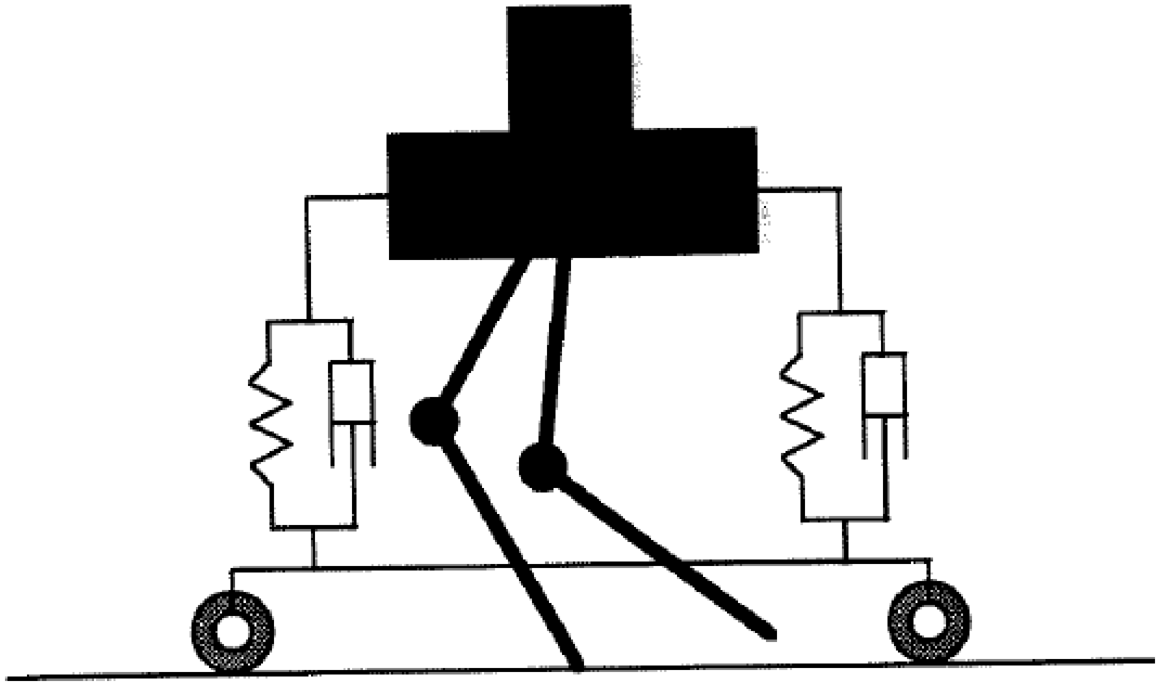


Fig. 6. Spring Turkey with virtual granny walker mechanism.

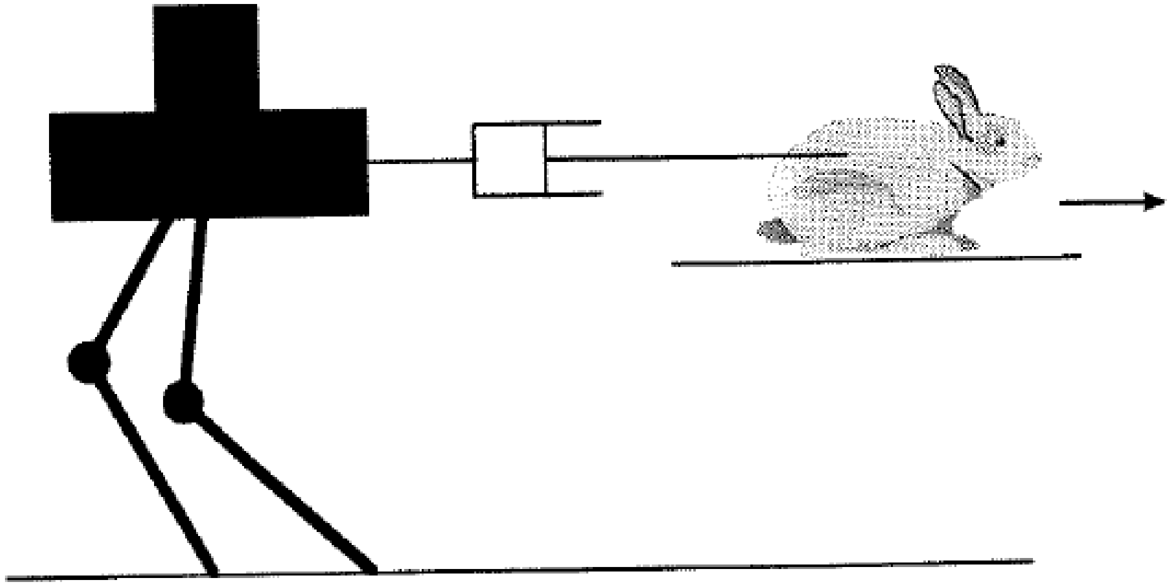


Fig. 7. Spring Turkey with virtual dogtrack bunny mechanism.

Level-Ground Walking (Spring Turkey)

Granny Walker

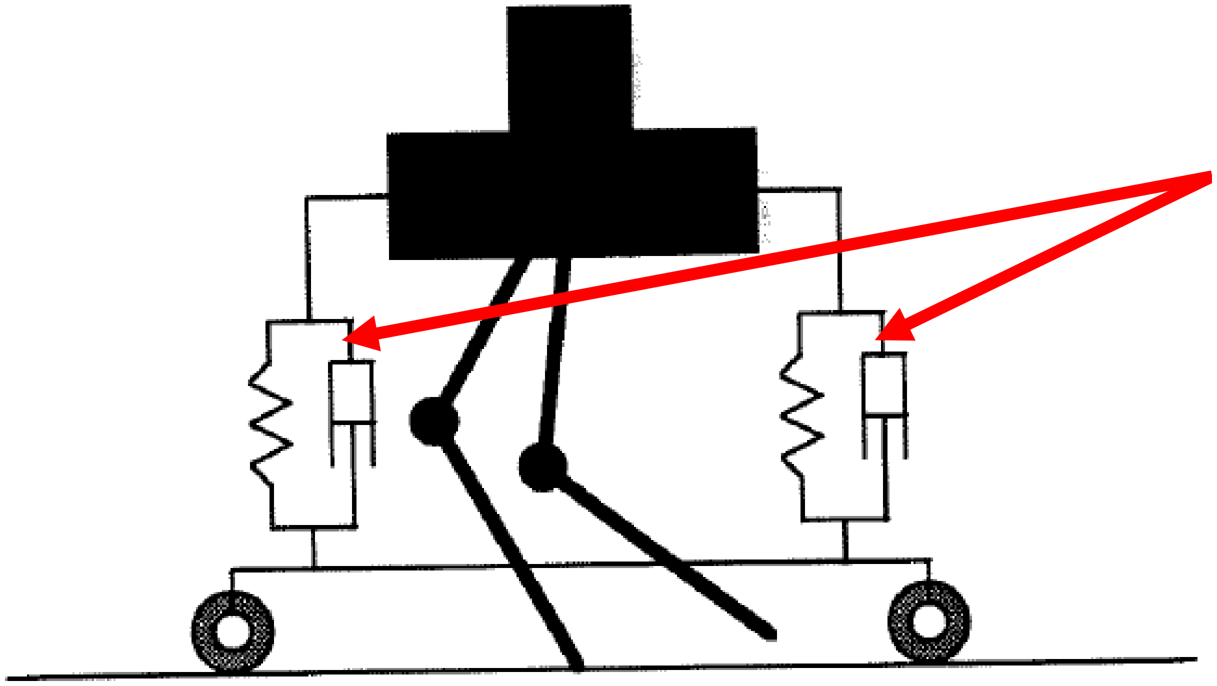


Fig. 6. Spring Turkey with virtual granny walker mechanism.

Maintain constant height and regulates pitch angle to zero

During BOTH double & single support phase

Level-Ground Walking (Spring Turkey)

Dogtrack bunny

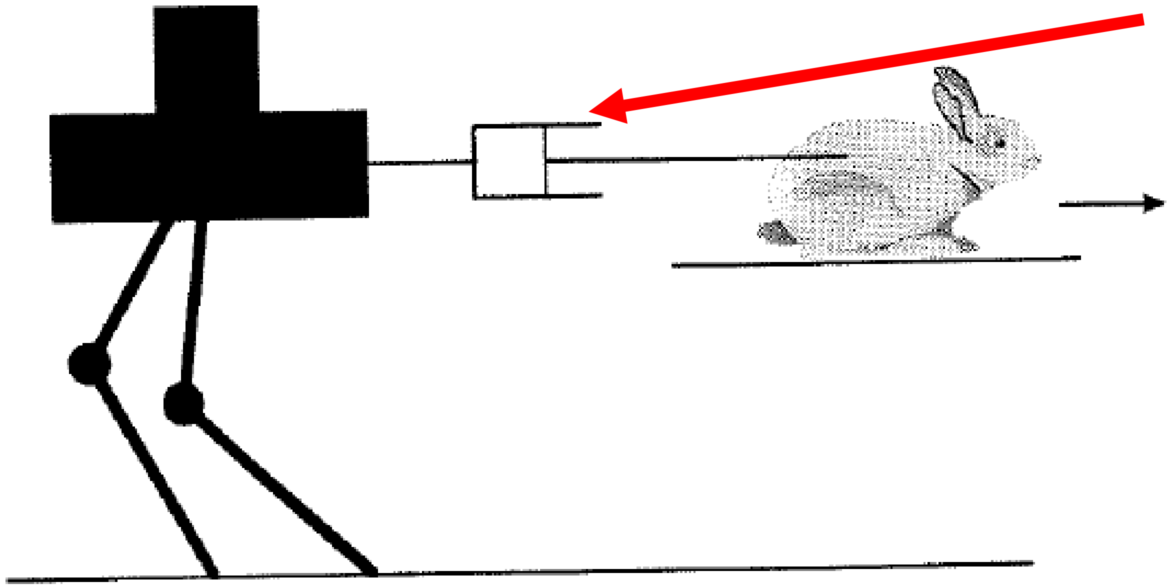


Fig. 7. Spring Turkey with virtual dogtrack bunny mechanism.

Applies a virtual forward horizontal force to maintain desired velocity

During ONLY double support phase

Level-Ground Walking (Spring Turkey)

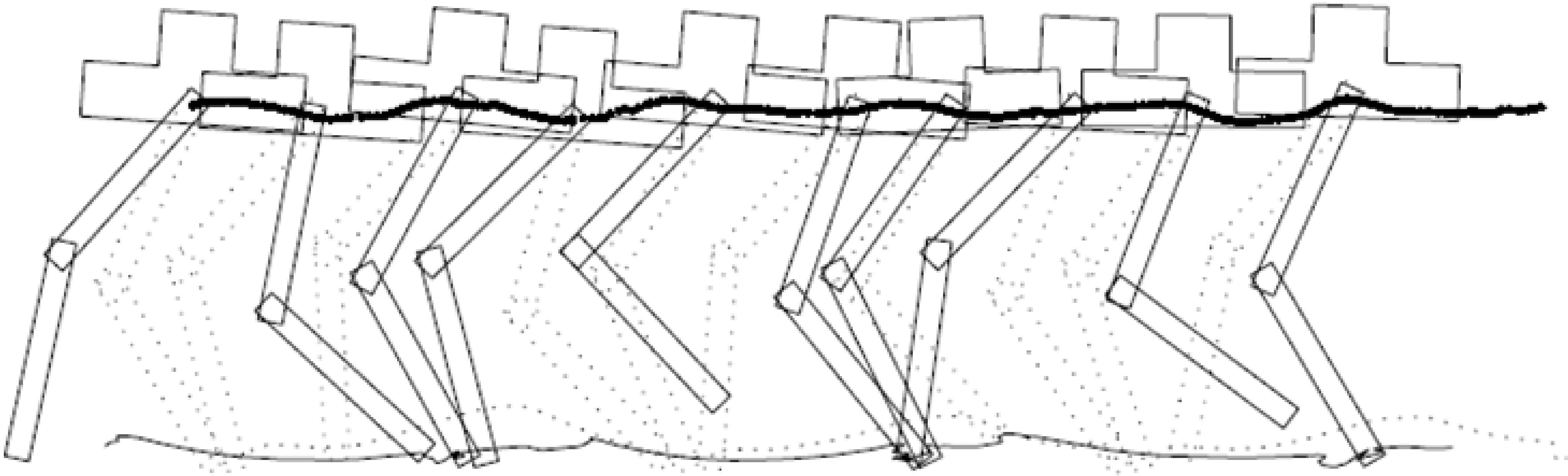
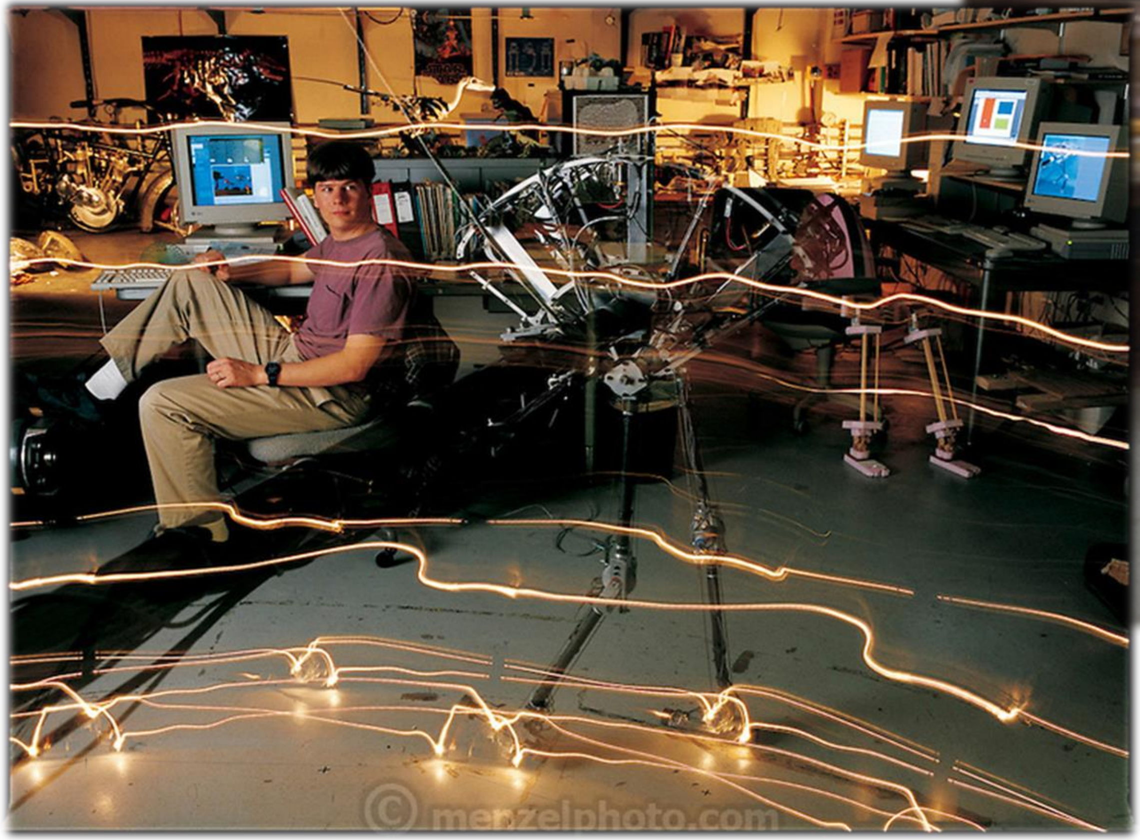
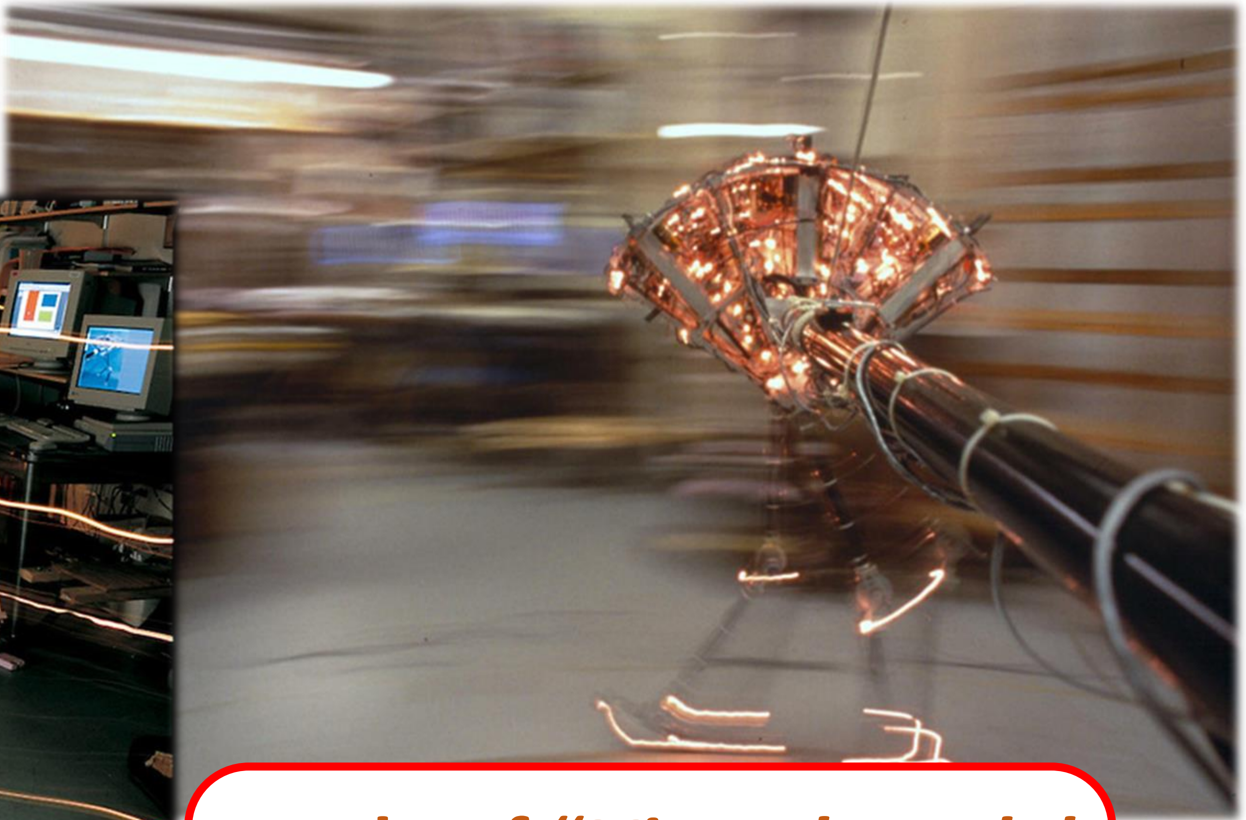


Fig. 8. Elapsed time snapshot of the bipedal walking data in Figure 9. The drawings of the robot are spaced approximately 0.5 s apart. The left leg is dotted, whereas the right leg is solid. Lines show the path of the tips of the foot and the center of the body.

Introducing Spring Flamingo





Mode of "*Virtual Model Control*" Application for Slope Walking

Slope Walking (Spring Flamingo)

1. Geometric Considerations

2. Transition Cases

12kg

84cm

Slope Walking (Geometric Considerations)

Upslope Walking

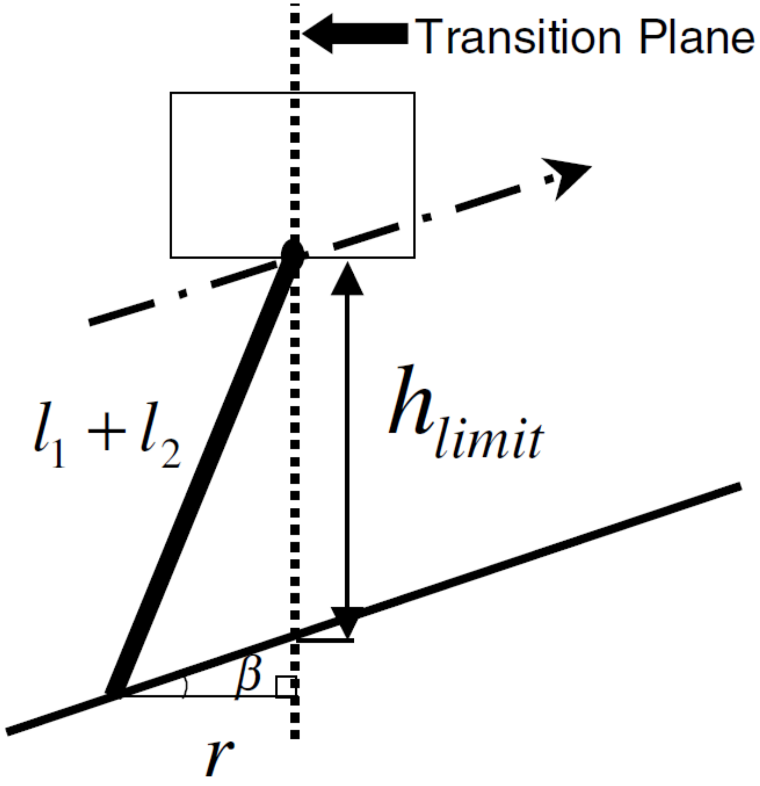


Fig. 10. Geometric constraint to calculate h_{limit}: level and upslope.

$$h_{limit} = \sqrt{(l_1 + l_2)^2 - r^2} - r \tan \beta, \qquad (15)$$

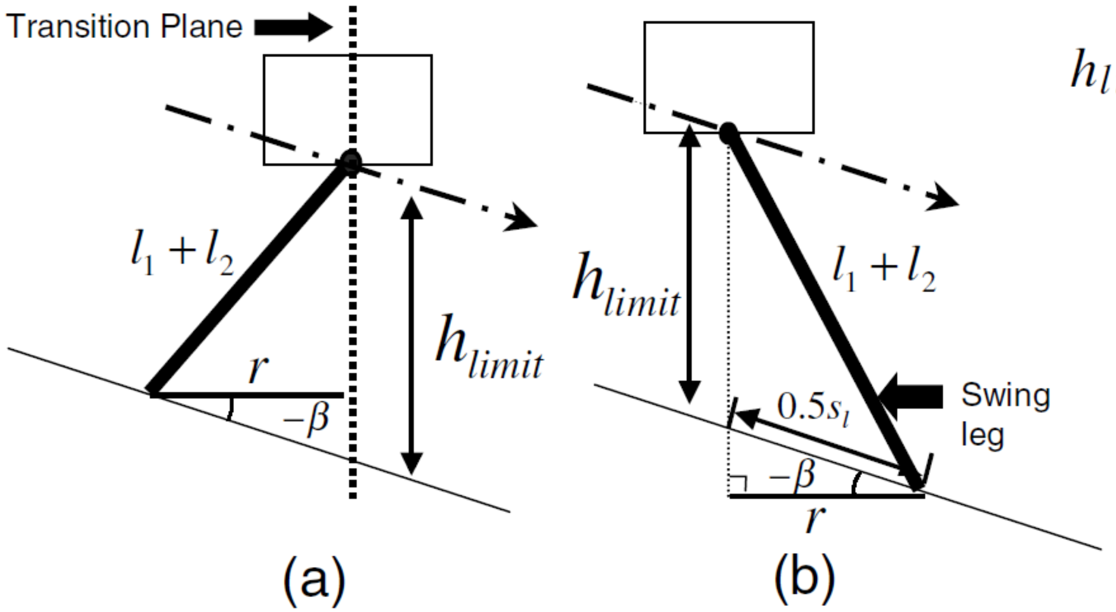
Desired Hip Height

Total Leg Length

Slope Gradient

Slope Walking (Geometric Considerations)

Downslope Walking



Geometric constraint to calculate h_{limit} during downslope walking: (a) case 1, (b) case 2.

$$h_{limit} = \sqrt{(l_1 + l_2)^2 - r_1^2} - r_1 \tan\beta, \qquad (16)$$

Purpose:

To ensure stability during slope walking!

Slope Walking (Transition Cases)

Level-to-Upslope

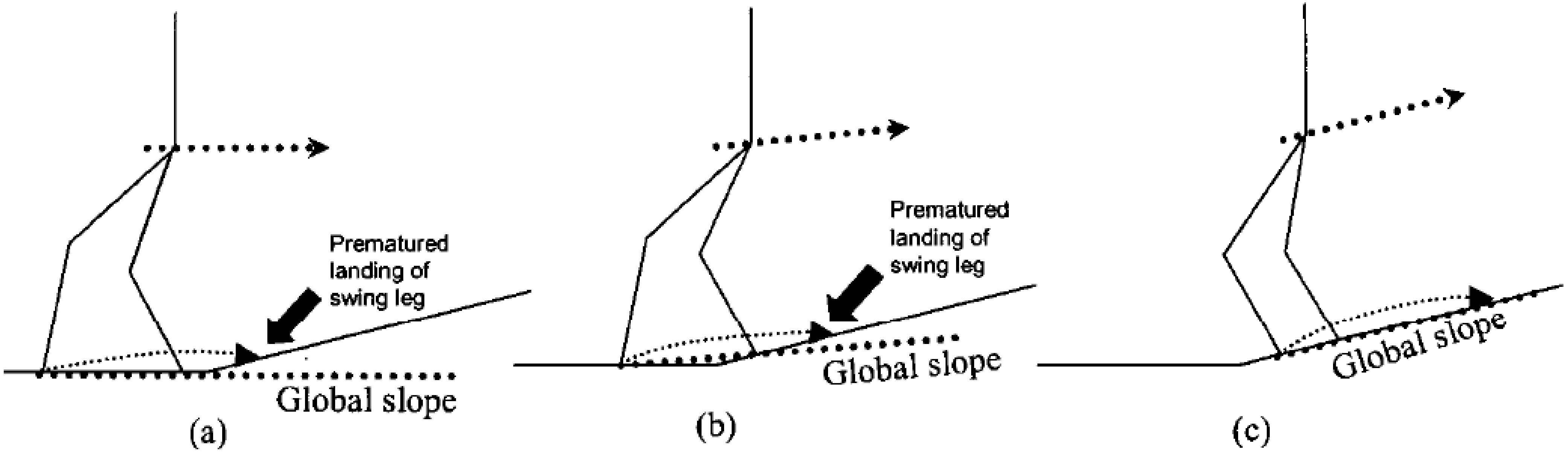


Fig. 12. Sequence for level-to-upslope transition. Note that in (b), global slope gradient is different from the actual slope gradients underneath both feet. In (c), all the detected slopes are the same.

Slope Walking (Transition Cases)

Level-to-Downslope

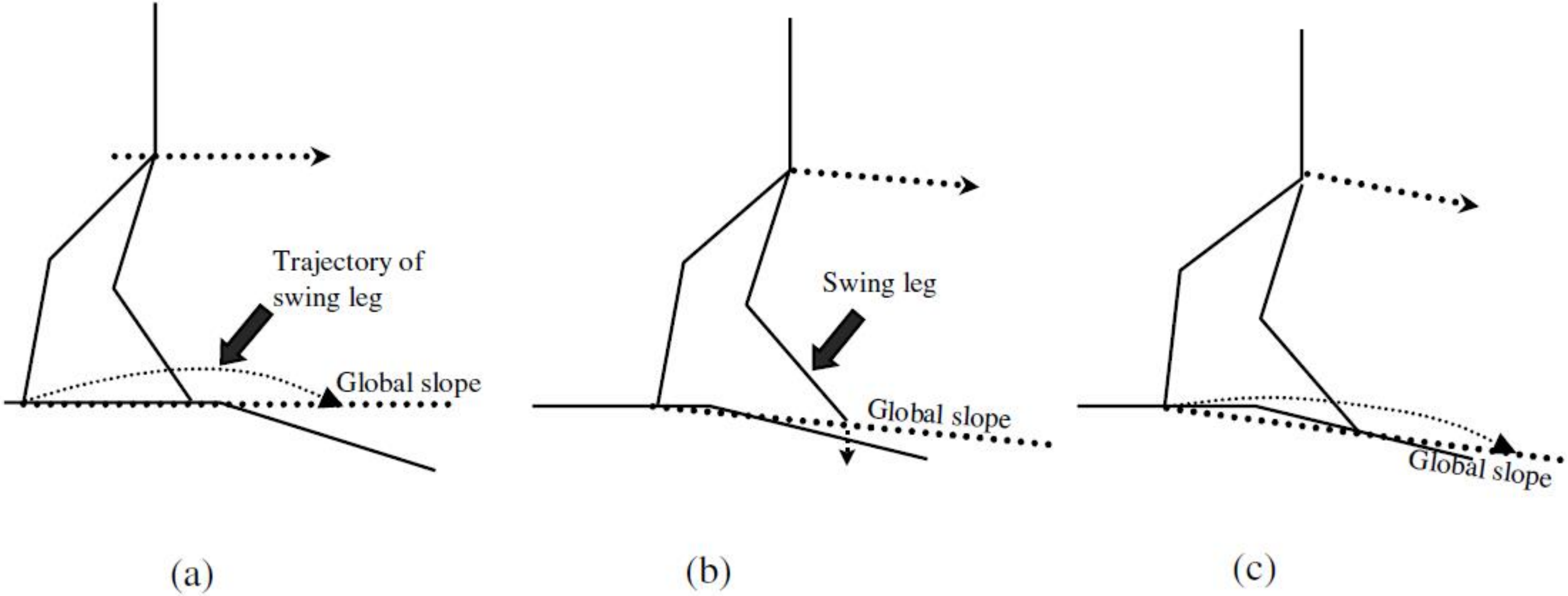
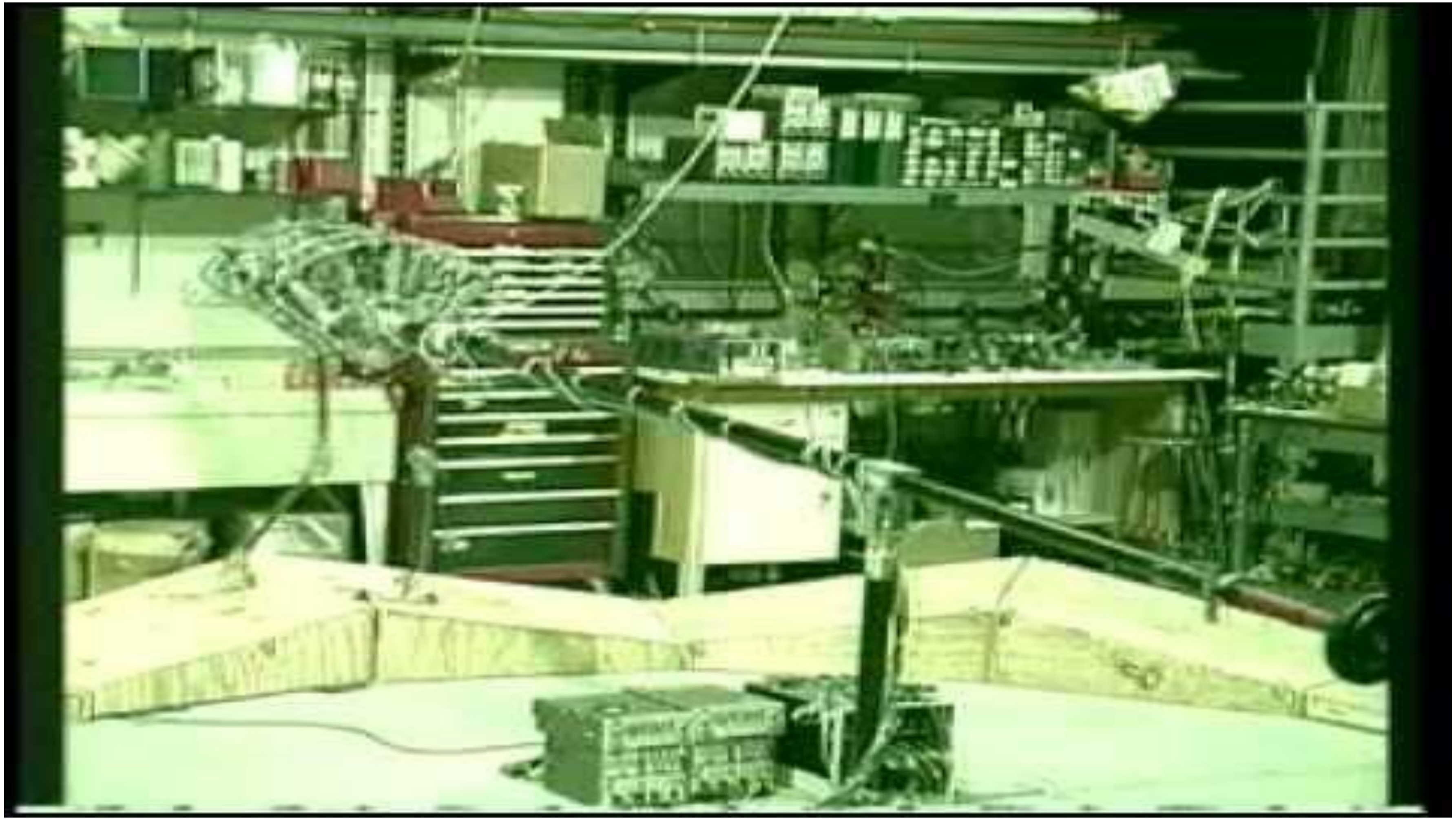


Fig. 13. Sequence for level-to-downslope transition.

Spring Flamingo (Application of VMC)

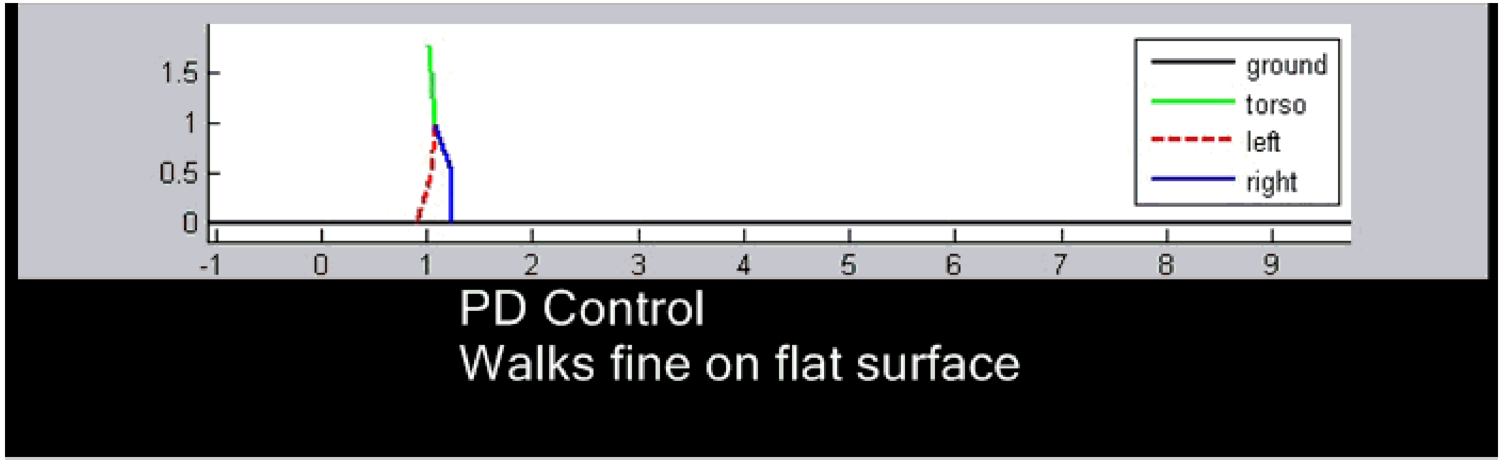


Source: https://www.youtube.com/watch?v=nF0Z_eHTHM8

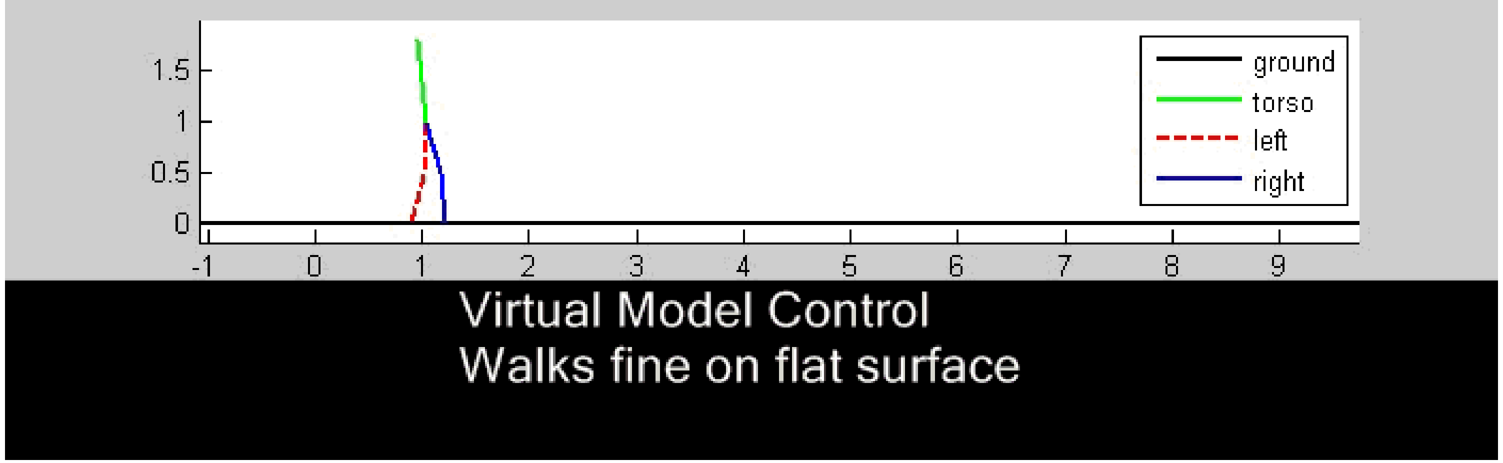
Slope Walking (Simulation Results)

Comparison between PD & Virtual Model Control

PD Control



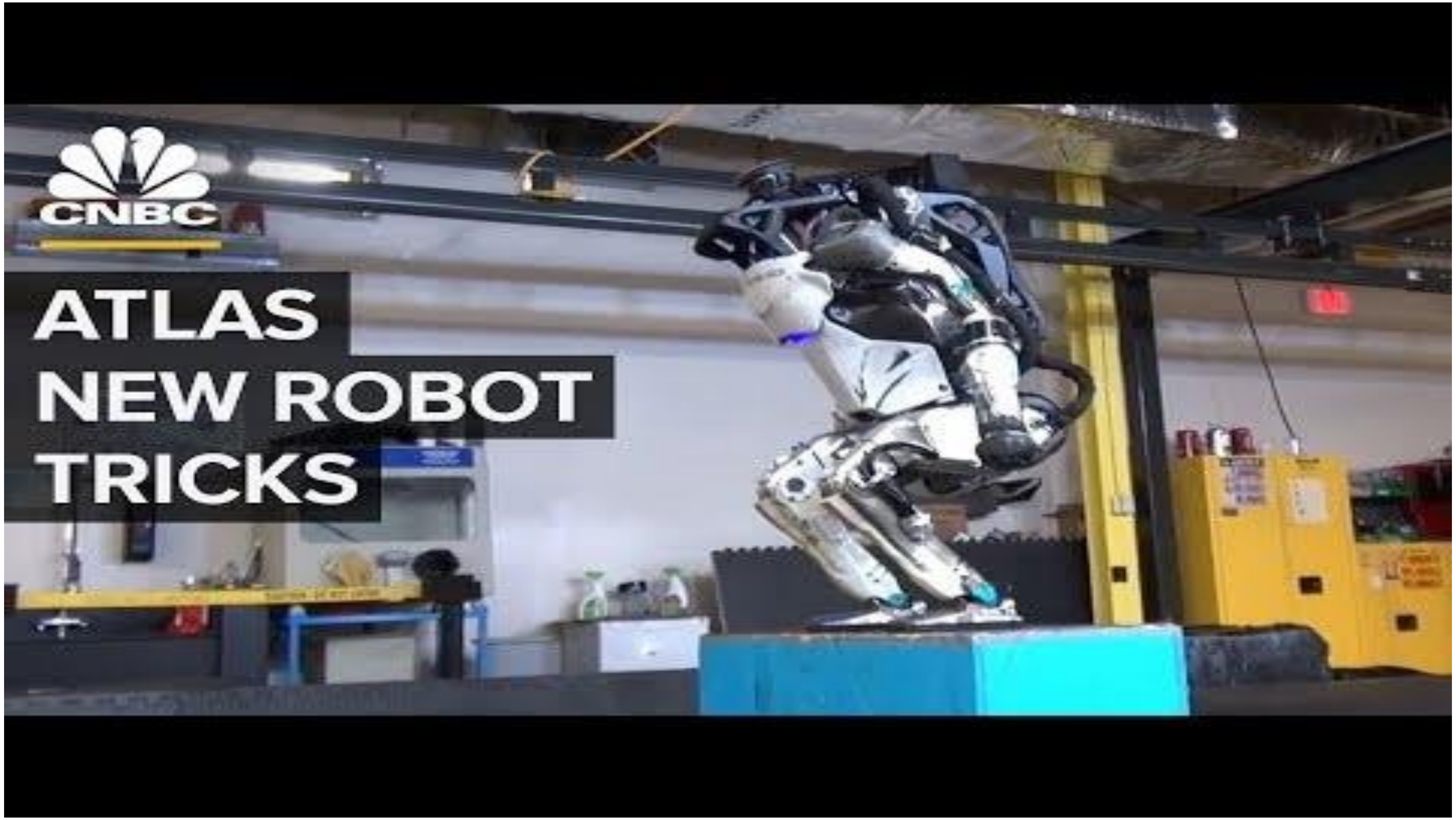
Virtual Model Control



Question:

Which method could possibly solve the extra mass issue?

Introducing Atlas from Boston Dynamics



Source: https://www.youtube.com/watch?v=hSjKoEva5bg

3. 3D-LIPM – 3D Linear Inverted Pendulum Mode

3D-LIPM is useful for controlling the robot, especially **real-time walking control in a 3D space** with the help of **control algorithms for both sideway and forward motions**

3D-LIPM – 3D inverted pendulum

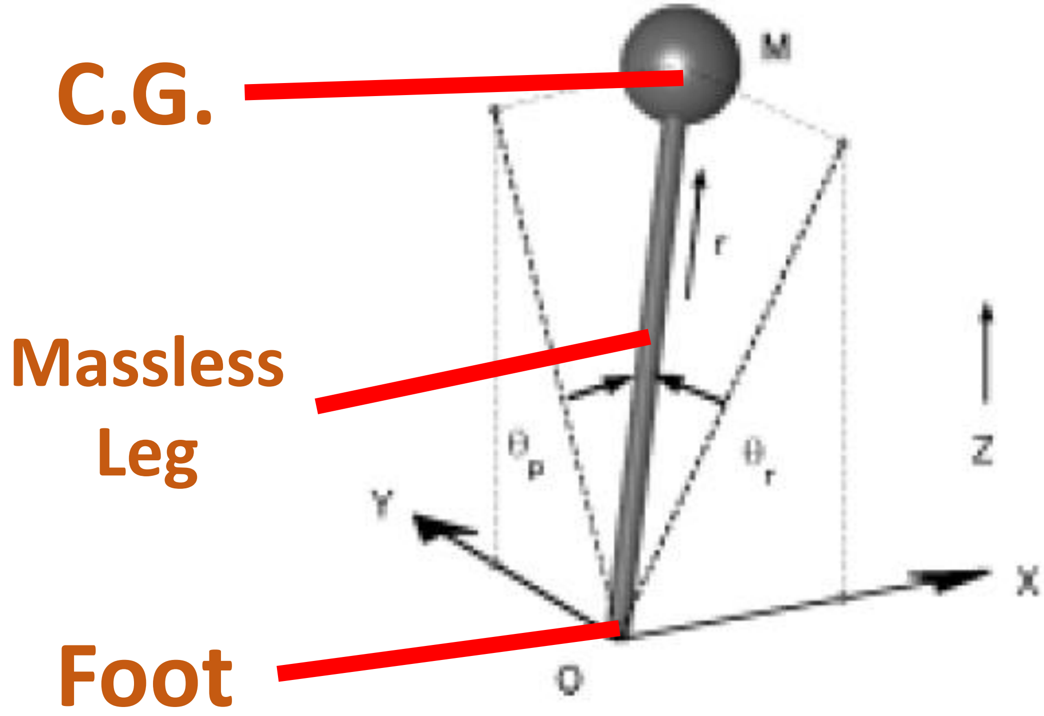
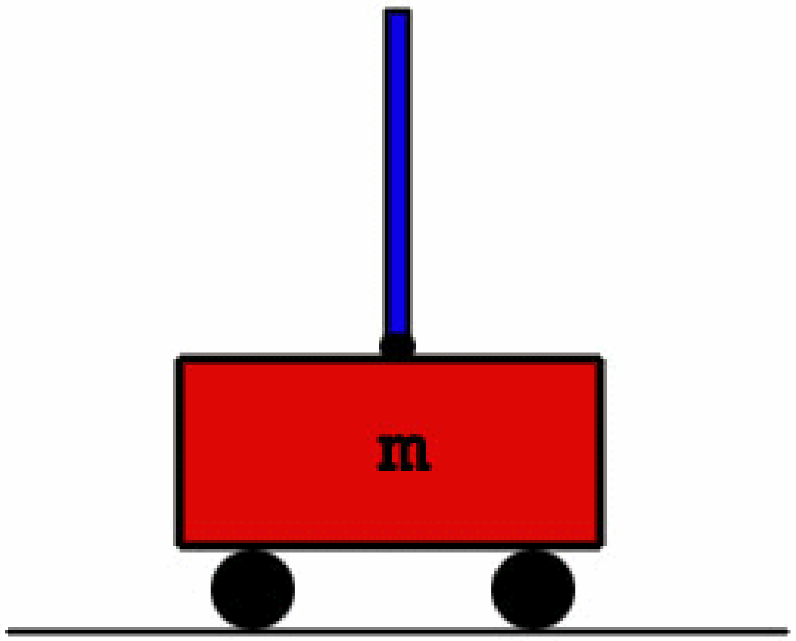
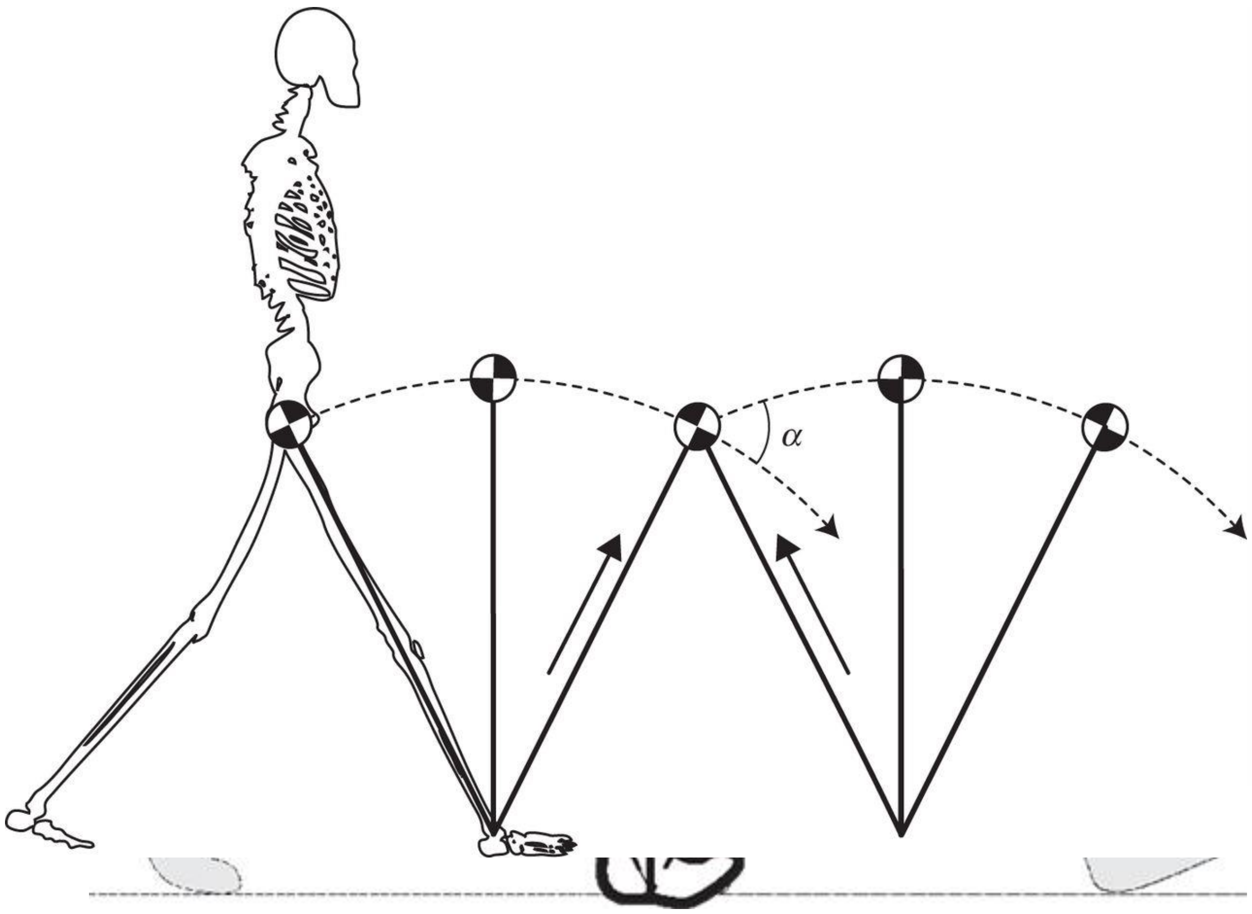
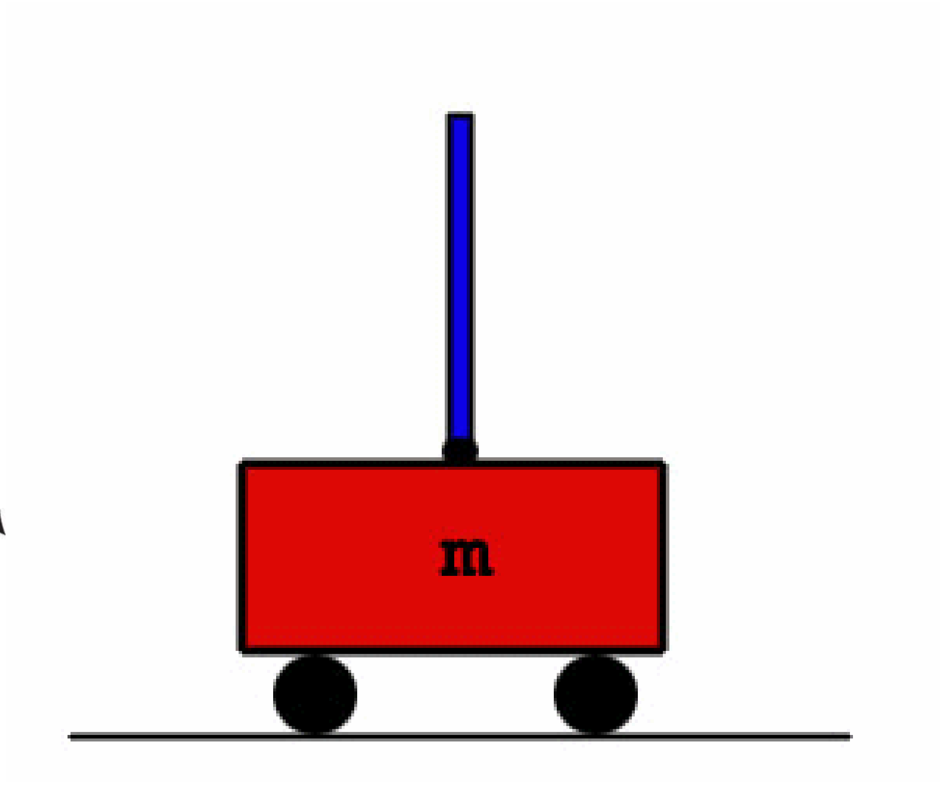


Figure 3: *3D Pendulum*



3D-LIPM – 3D inverted pendulum





3D-LIPM – 3D inverted pendulum

Dynamic equation along y-axis (sideway):

$$m(-z\ddot{y} + y\ddot{z}) = \frac{D}{C_r}\tau_r - mgy \tag{8}$$

Dynamic equation along x-axis (forward):

$$m(z\ddot{x} - x\ddot{z}) = \frac{D}{C_p}\tau_p + mgx \tag{9}$$

Non-linear Equations!

3D-LIPM – Linearization

Apply constraints instead of approximation to linearize the non-linear dynamics equations!

3D-LIPM – Linearization

General constraints (to limit the motion of the pendulum):

$$z = k_x x + k_y y + z_c \qquad (10)$$

Motion is limited in a plane with (k_x,k_y,-1) and z_c

$$\ddot{z} = k_x \ddot{x} + k_y \ddot{y} \qquad (11)$$

2nd derivative of Eqn (10)

For walking on a flat plane, the following constraint is applied:

$$k_x = 0, k_y = 0$$

No slope

3D-LIPM – Linearization

Hence,

From

$$m(-z\ddot{y} + y\ddot{z}) = \frac{D}{C_r}\tau_r - mgy \qquad (8)$$

$$m(z\ddot{x} - x\ddot{z}) = \frac{D}{C_p}\tau_p + mgx \qquad (9)$$

Apply General Constraints

$$\ddot{y} = \frac{g}{z_c}y - \frac{k_1}{z_c}(x\ddot{y} - \ddot{x}y) - \frac{1}{mz_c}u_r \qquad (12)$$

$$\ddot{x} = \frac{g}{z_c}x + \frac{k_2}{z_c}(x\ddot{y} - \ddot{x}y) + \frac{1}{mz_c}u_p \qquad (13)$$

$$z = k_x x + k_y y + z_c \qquad (10)$$

$$\ddot{z} = k_x\ddot{x} + k_y\ddot{y} \qquad (11)$$

3D-LIPM – Linearization

And then……

From

$$\ddot{y} = \frac{g}{z_c} y - \frac{k_1}{z_c}(x\ddot{y} - \ddot{x}y) - \frac{1}{mz_c} u_r \quad (12)$$

$$\ddot{x} = \frac{g}{z_c} x + \frac{k_2}{z_c}(x\ddot{y} - \ddot{x}y) + \frac{1}{mz_c} u_p \quad (13)$$

Apply Constraints for Flat Plane

$$\boxed{\begin{aligned} \ddot{y} &= \frac{g}{z_c} y - \frac{1}{mz_c} u_r \quad (16) \\ \ddot{x} &= \frac{g}{z_c} x + \frac{1}{mz_c} u_p. \quad (17) \end{aligned}}$$

$$k_x = 0, k_y = 0$$

Control of Lateral Motion (y-z) – Sideways

From

$$\ddot{y} = \frac{g}{z_c}y - \frac{1}{mz_c}u_r \qquad (16)$$

Virtual Input $u_r = \dfrac{\tau_r D}{C_r} = 0$

$$\ddot{y} = \frac{g}{z_c}y \qquad (20)$$

Ideal dynamics of equation

Control of Lateral Motion (y-z) – Sideways

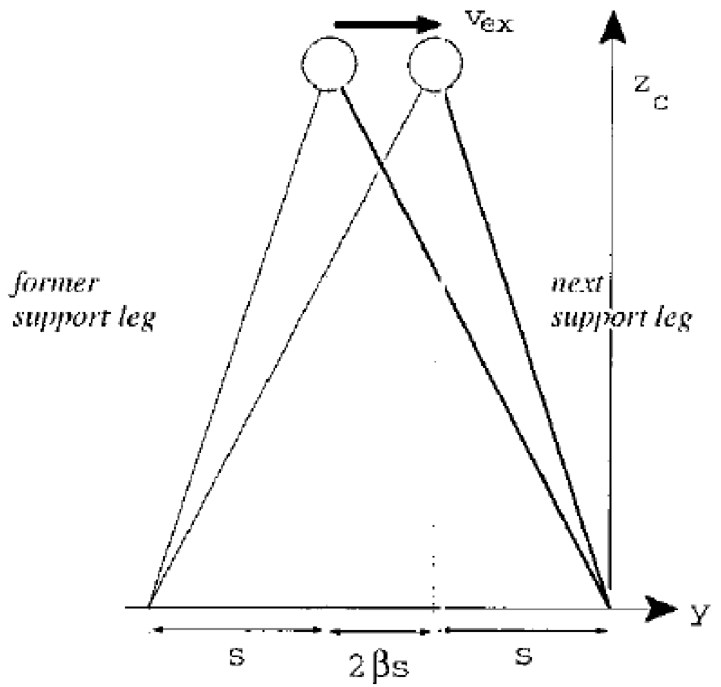


Figure 7: *For reliable support leg exchange, a double-leg support phase was inserted. The robot body travels the distance $2\beta s$ with both feet on the ground in the period between each single-leg support phase.*

Control of Sagittal Motion (x-z) – Forward

From

$$\ddot{x} = \frac{g}{z_c}x + \frac{1}{mz_c}u_p. \quad (17)$$

Virtual Input $u_p = \frac{\tau_p D}{C_p} = 0$

$$\ddot{x} = \frac{g}{z_c}x \quad (31)$$

Ideal dynamics of equation

Control of Sagittal Motion (x-z) – Forward

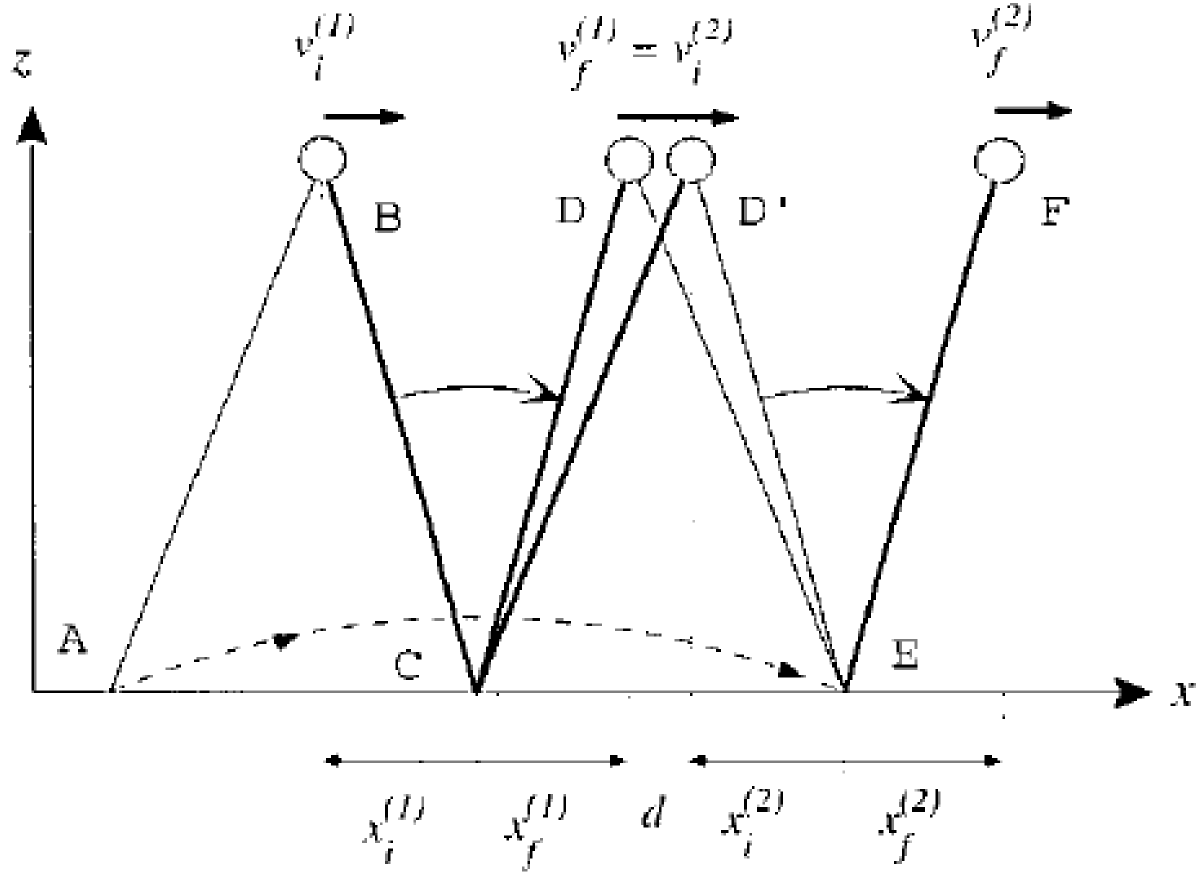
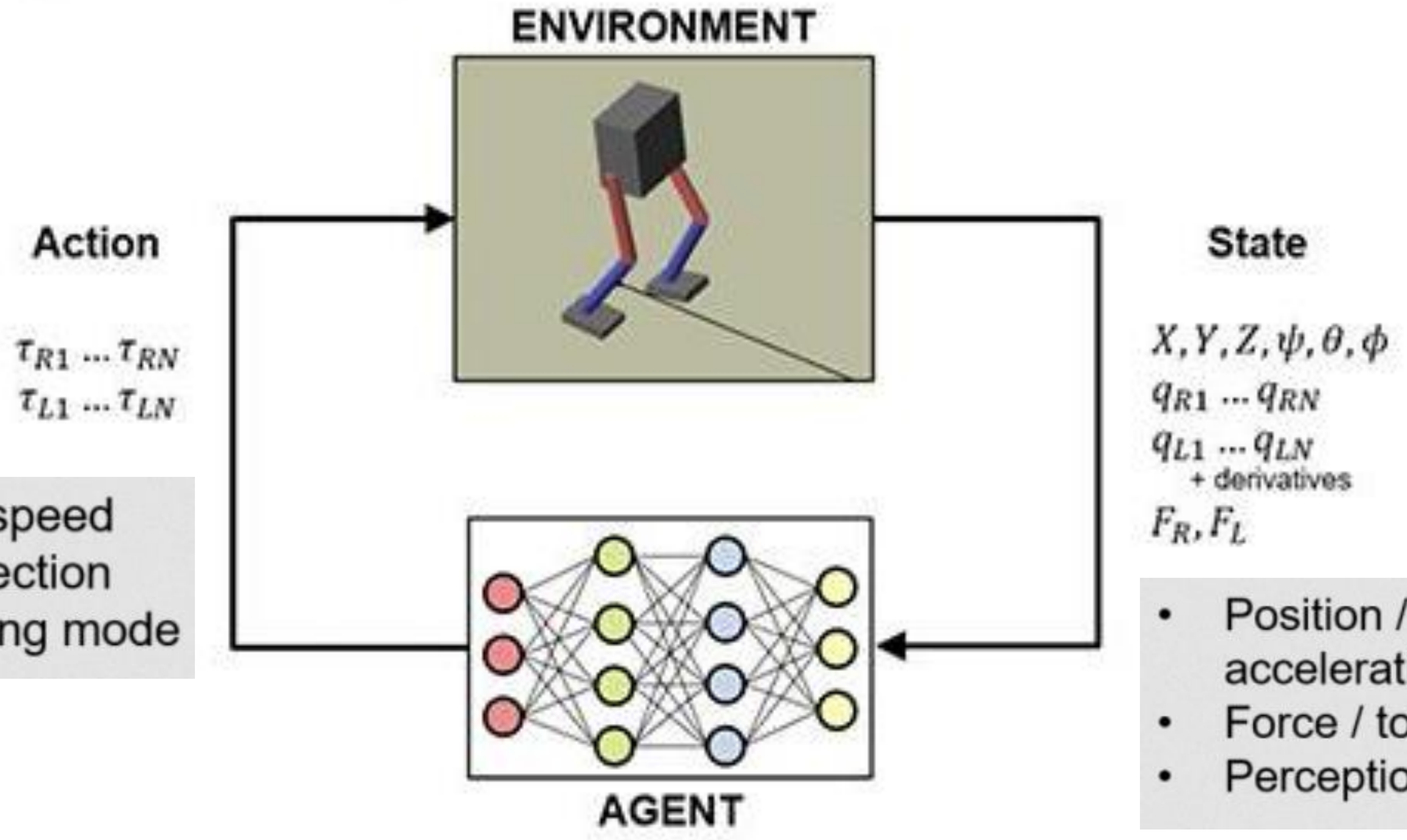


Figure 10: *Two successive steps in the sagittal plane are illustrated. The body travels from B to D in the single-leg support phase, then moves from D to D' in the double-leg support phase with constant speed $v_f^{(1)}$, and then travels D' to F in the second single-support phase. While the body moves from B to D, the tip of the swing leg travels from A to E (dashed curved line). By changing the position of E we can control the final body speed $v_f^{(2)}$ at the point F. Except for our inserted double-support phase, this is the same idea proposed by Miura and Shimoyama [7].*

4. Deep Reinforcement Learning



- Target speed and direction
- Operating mode

- Position / speed / acceleration
- Force / torque
- Perception sensors

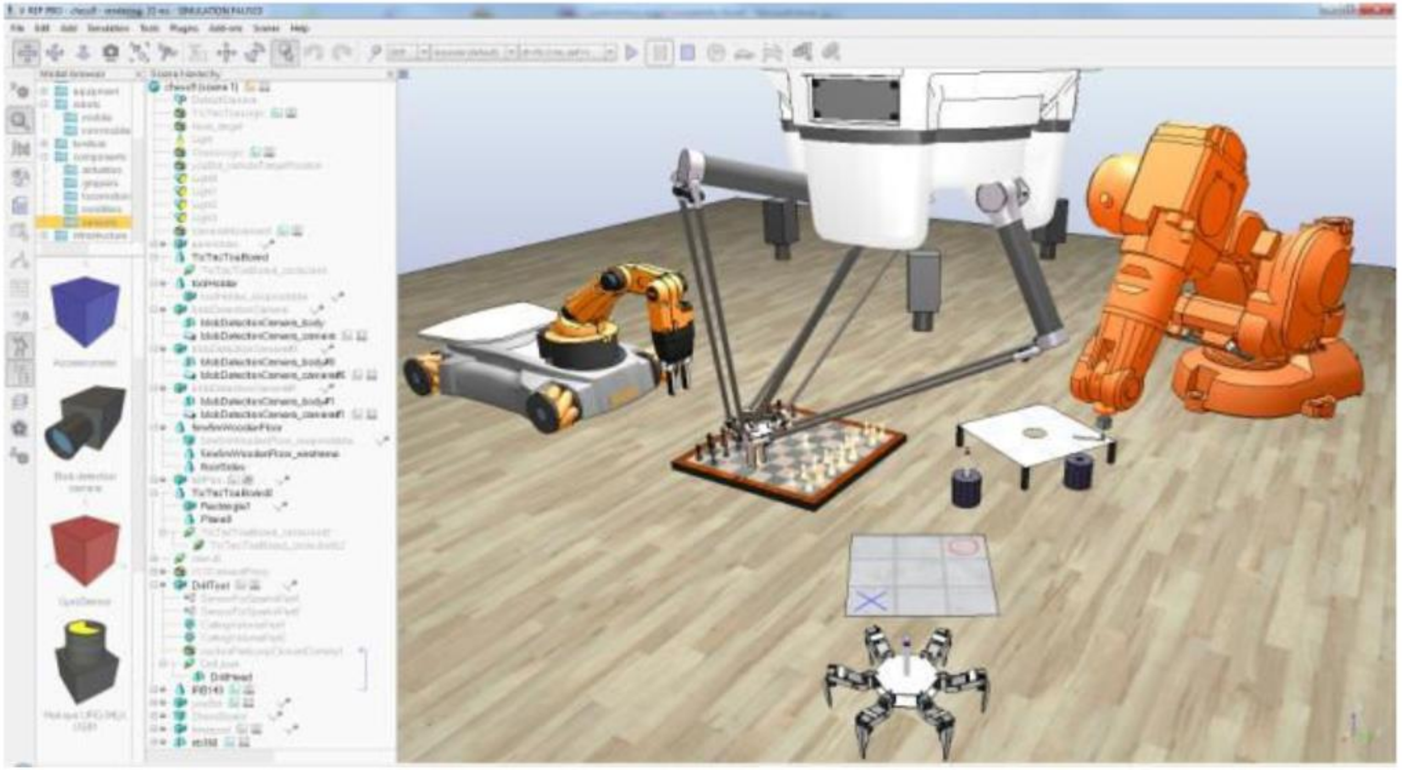
Chapter 3: Introduction to Robot Simulation Software

Many Types of Software for Robot Simulation

1. **CoppeliaSim (http://www.coppeliarobotics.com/)**
2. **Gazebo (http://gazebosim.org/)**
3. Webots (https://cyberbotics.com/)
4. MATLAB Simulink (Licenced software)

Introduction to CoppeliaSim

- Formerly known as VREP (Virtual Robot Experimentation Platform)
- A versatile, scalable, yet powerful general-purpose robot simulation framework
- Cheaper option to test robots than hardware development



Can Choose Various Programming Techniques

1. Embedded script
2. Add-ons
3. Plug-ins
4. Remote API client (can be embedded with other codes like C/C++, Python, Java, Matlab)
5. ROS nodes

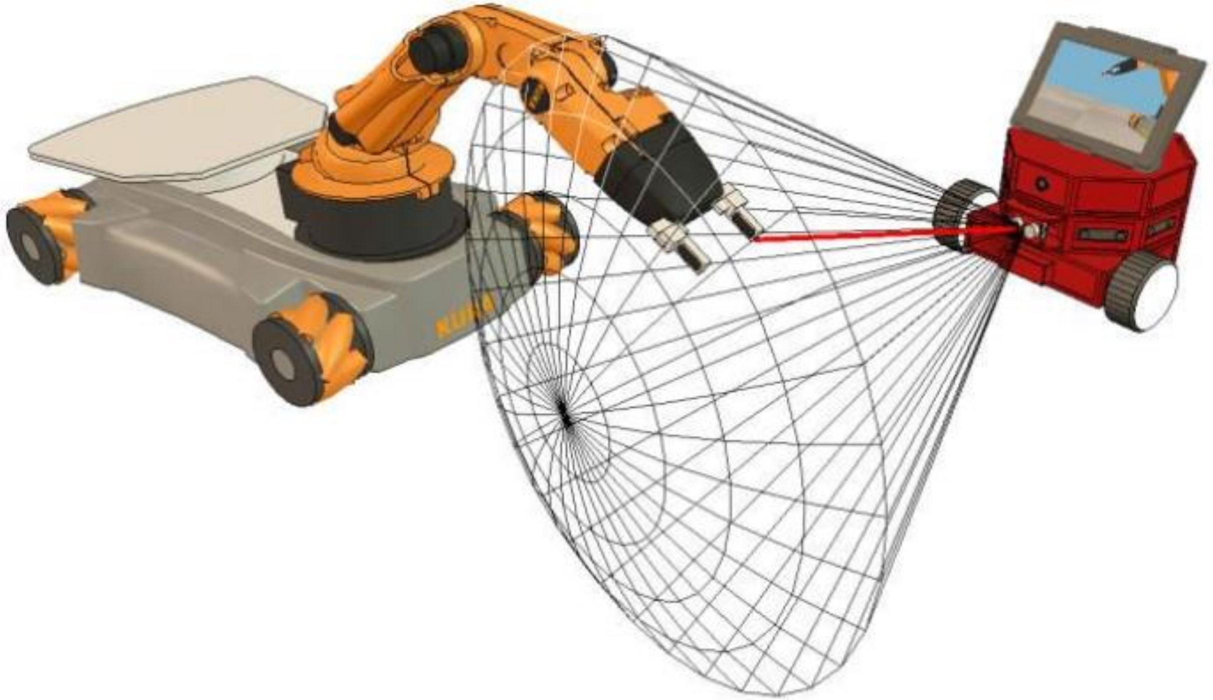
	Embedded script	Add-on	Plug-in	Remote API client	ROS node
API mechanism	Regular API	Regular API	Regular API	Remote API	ROS
Supported programming language	Lua	Lua	C/C++	C/C++, Python, Java, Matlab, Urbi	Depends on ROS support
Available API functions	>280 functions	>270 functions	>400 functions	>100 functions	>150 services, publisher and subscriber types
API is user-extendable	Yes, with custom Lua functions	Yes, with custom Lua functions	Yes, V-REP is open source	Yes, remote API is open source	Yes, ROS interface is open source
Control entity is external and can be native	No	No	No	Yes	Yes
Simulation models are fully portable and scalable	Yes	No	No	No	No
Communication lag	None	None	None	Yes	Yes
Synchronous operation is supported*	Yes, inherent. No delays	Yes, inherent. No delays	Yes, inherent. No delays	Yes. Slow due to comm. lag	Yes. Slow due to comm. lag
Asynchronous operation is supported*	Yes	No	Yes	Yes	Yes
Can start, stop, pause and step a simulation	Stop, pause	Start, stop, pause	Start, stop, pause, step	Start, stop, pause, step	Start, stop, pause, step

* *Synchronous* in the sense that each simulation pass runs synchronously with the control entity

Simulation Functionality

1. Scene Objects within the simulation scene; E.g.

- Joints
- Shapes
- Proximity sensors
- Vision sensors
- Force sensors
- Graphs
- Cameras
- Lights
- Paths

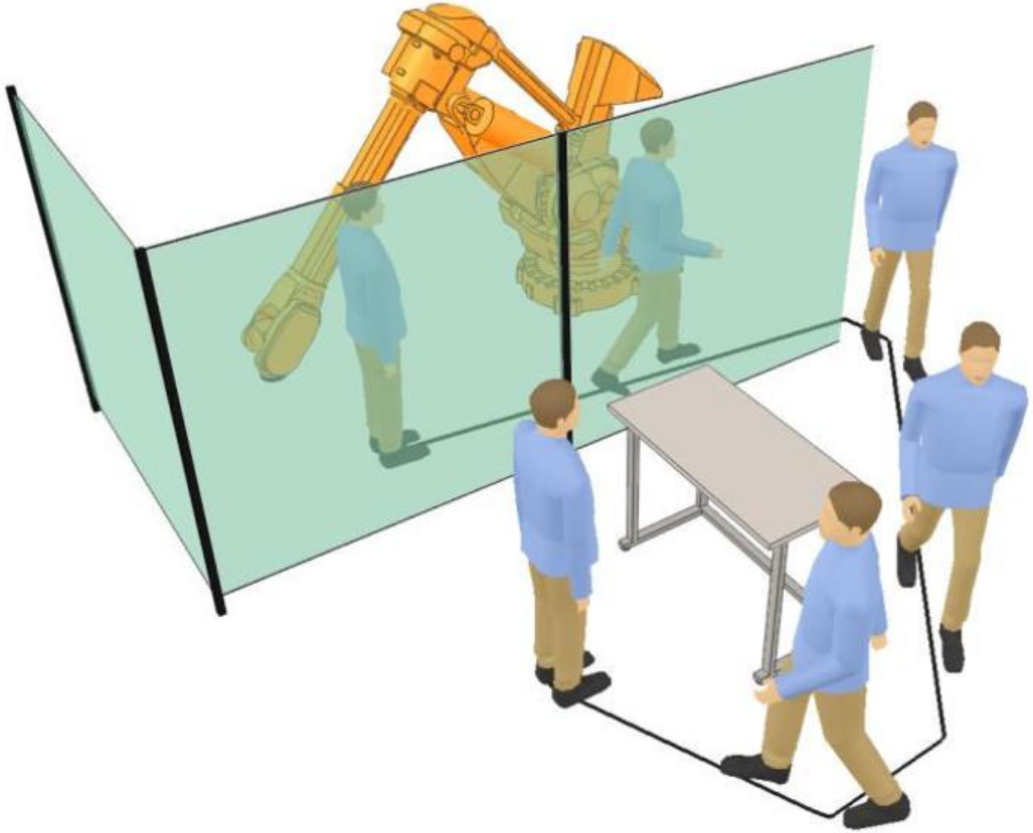


Proximity sensor in V-REP

Simulation Functionality

2. Calculation Modules to support scene objects; E.g.

- Kinematics Module
- Dynamics Module
- Collision Detection Module
- Mesh-mesh Distance Calculation Module
- Path/motion Planning Module



Path planning human model in V-REP

Let’s Try Out CoppeliaSim!

- **Install CoppeliaSim_EDU** .exe file from LumiNUS; you can download from this link as well: http://www.coppeliarobotics.com/downloads.html
- Once installed, open the programme
- We will test the following simulated robots:
  1. **7-DOF** arm manipulator
  2. **ABB IRB** (an arm manipulator)
  3. **ASTI** humanoid robot
  4. **NAO** humanoid robot (mentioned in the previous module!)
  5. Other robot types that you wish to play around with

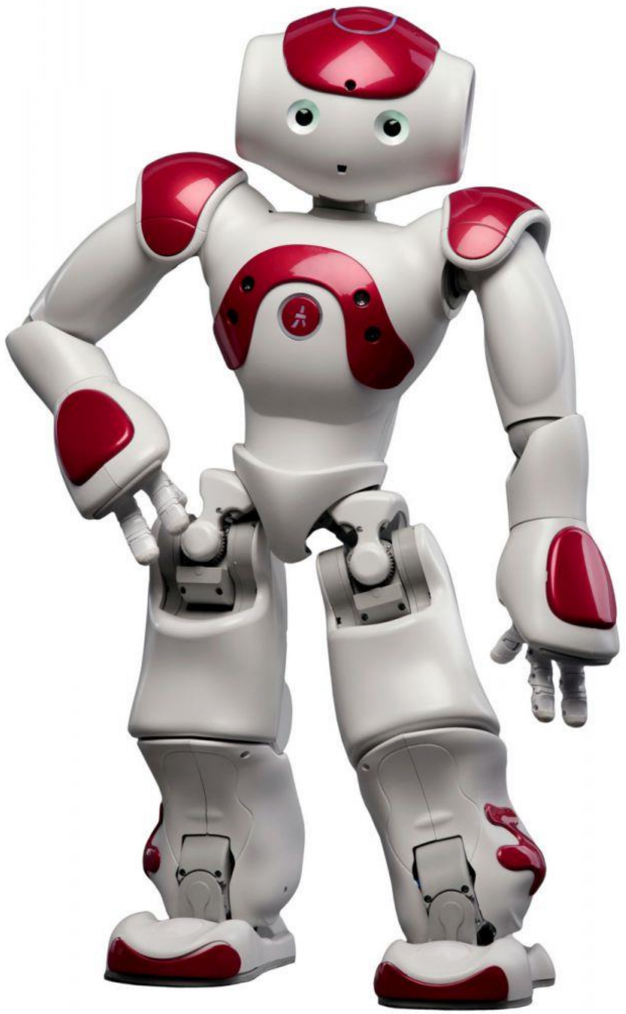
Chapter 4: Summary & Conclusions

Summary & Conclusions

- The most common locomotion methods for robots are **wheeled** and **legged** types
- The 3 main factors to consider when choosing the most suitable locomotion method are: (a) **Nature of terrain**, (b) **Nature of task**, and (c) **Physical considerations**
- Understanding mobility models for bipedal robots, in particular walking movements, is challenging because of the **common stability issues that bipedal robots faced**
- These mobility models include: (a) **PID**, (b) **Virtual Model Control**, (c) **3D-LIPM,** and (d) **Reinforcement Learning**

Thank you!

Questions?



Reflections Time!