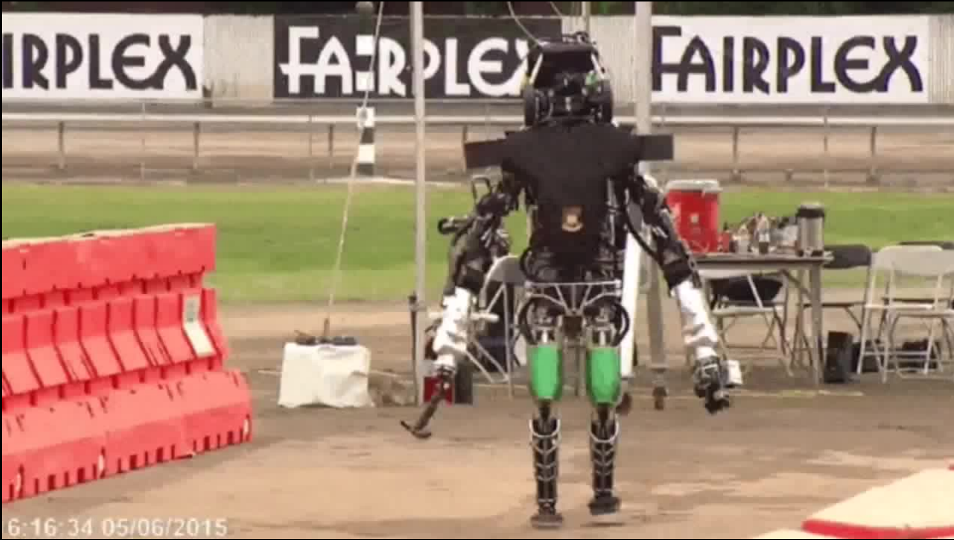
ROBOTICS
WITH
PLYMOUTH
UNIVERSITY

ROCO318
Mobile and Humanoid Robots
Bipedal Robots

Tony Belpaeme and Séverin Lemaignan

Centre for Neural Systems and Robotics
Plymouth University



IRPLEX
FAIRPLEX
6:16:34 05/06/2015

WHY WALKING ROBOTS?

Mobility

- Wheeled robots only function on prepared surfaces (roads, rails, ...)
- Legged robots can negotiate difficult terrain, which wheeled robots cannot reach

Understanding

- Understanding animal legged motion

WHY BIPEDAL WALKING?

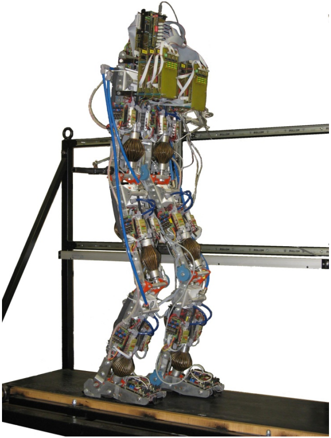
Because humanoid robots only have two legs

- Better mobility over rough terrain
- Active suspension that stabilises the load
- Overcoming obstacles, i.e. avoiding or moving over or under obstacles

Humans relate to humanoid robots much more readily

- Human–Robot Interaction

Lower hardware complexity



Source: Lucy, VUB

WHY BIPEDAL WALKING?

As a tool for understanding human gait disorders

Active prosthetics

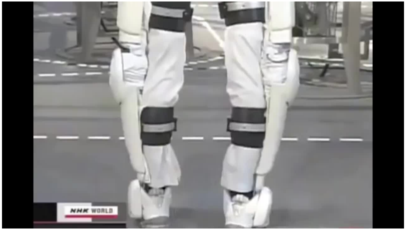
- E.g. Hugh Herr's prosthetic legs.

Exoskeletons

- E.g. Cyberdyne HAL

The human environment has been adapted profoundly to bipedal locomotion.

Just because it's cool!



CHARACTERIZING WALK

WHAT IS BIPEDAL WALKING?

A **gait cycle** starts with an initial contact of one foot with the ground, and ends at the next contact of the same foot with the ground.

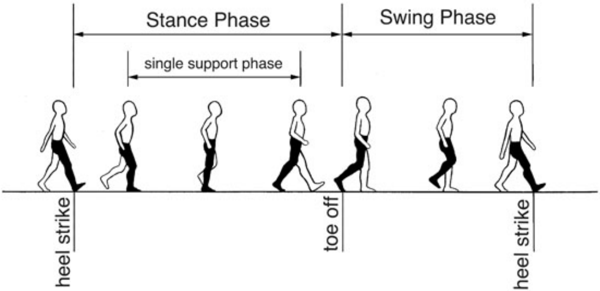
Walking: the two feet are simultaneously in contact with the ground at some point during the gait cycle.

Running: one foot at maximum touches the ground at any point in the gait cycle.

WHAT IS BIPEDAL WALKING?

Two phases to a gait cycle:

- **Stance phase** –when the foot is in contact with the ground (also called *support phase*);
- **Swing phase** –when the foot is lifted of the ground.



ACTIVE VS PASSIVE WALK

Active Walk

- An active walk is when the robot is internally powered. This is usually through electrical servos, pneumatics or hydraulics.
- Versatile
- Continuous high bandwidth control
- Fully actuated, resulting in high energy consumption
- BigDog (2, talk), Honda Asimo, Bioloid, ...

ACTIVE VS PASSIVE WALK

Passive Walk

- A passive walk is when a robot is not internally powered. This is usually driven by an initial push or through gravity (downhill)
- Fixed walking speed, no starting, stopping or turning
- No actuation (slope needed)
- No control
- Very low energy consumption
- Passive walker example, Nagoya U.

STATIC VS DYNAMIC WALK

Static Walk

- A static walk is when the robot is **always balanced**. If the robot was to stop at any point in the gait cycle then it would not fall over.
- Control is easy.
- Movements are slow.
- Aldebaran Nao (not always), LittleDog

STATIC VS DYNAMIC WALK

Dynamic Walk

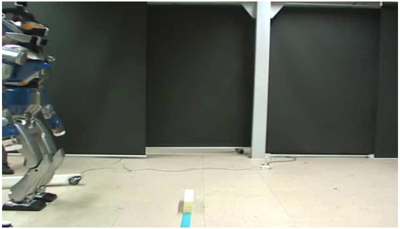
- A dynamic walk when the robot is **not always balanced**. If the robot was to stop at any point in the gait cycle then it would fall over. **Controlled "falling"**.
- Control is hard.
- Fast(er) motion.
- Honda Asimo, Cornell walkers, Delft walkers, Schaft robot

STATIC VS DYNAMIC WALK

Humanoid Robotics Project (HRP2) robot (Japan, 1997-)

Static walking

Dynamic walking



ROBOT WALK TYPOLOGY – SUMMARY

Passive/Active

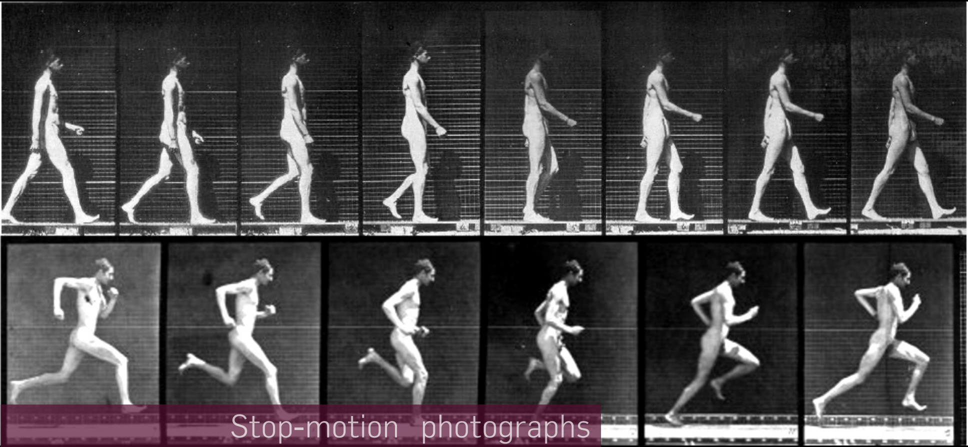
Active static
Always in balance,
with energy input

Active dynamic
Controlled falling,
with energy input

Static/ Dynamic

Passive static
Does not walk.
A brick.

Passive dynamic
Controlled falling,
no energy input to the system



Stop-motion photographs
by Eadweard Muybridge

HOW DO PEOPLE WALK?

Human Perspective

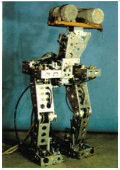
- Humans use a mixture of active/passive and static/dynamic walking.
- Walking requires generating energy (active) but the energy is also conserved where possible (passive).
- Humans can balance (static) although this requires more energy therefore they use an efficient unbalanced gait (dynamic).

Robot Perspective

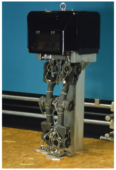
- Most robot research has focused on either an active static gait, an active dynamic gait or a passive dynamic gait.

SPEED OF ROBOTS

Walking person ±*5km/h*

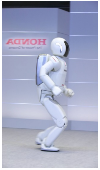


WL-5 ('70) *45s/step*

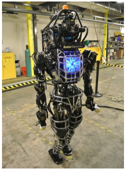


E2 ('91) *1.2km/h*

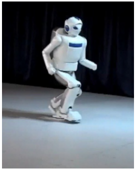
Nao *1.6km/h*



Asimo *6km/h*



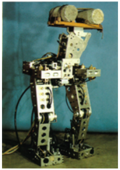
Atlas robot *7km/h*



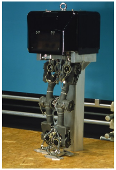
Partner *7km/h*

SPEED OF ROBOTS

Walking person $\pm$*5km/h*

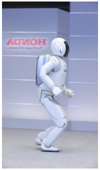


WL-5 ('70) *45s/step*

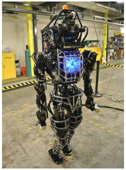


E2 ('91) *1.2km/h*

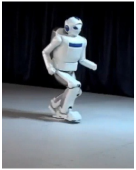
Nao *1.6km/h*



Asimo *6km/h*



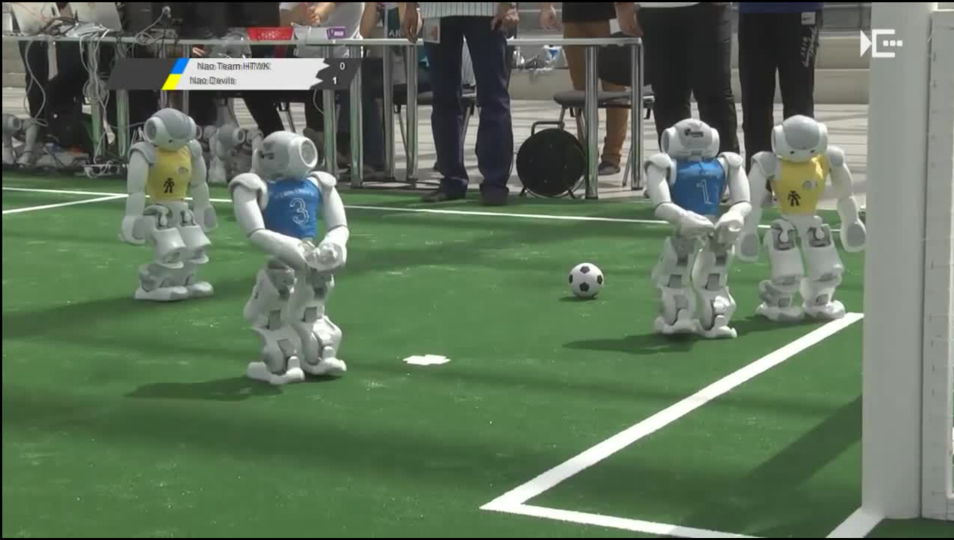
Atlas robot *7km/h*



Partner *7km/h*

Biped robot speed is often expressed as body lengths per second, to make comparison with different sized robots possible.

Usain Bolt WR 100m: 36.8km/h

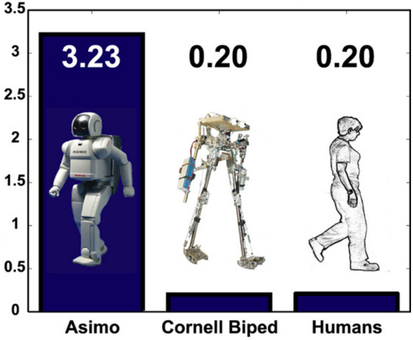

Nao Team HTWK 0
Nao Devils 1

ENERGY CONSUMPTION: COST OF TRANSPORT

$$\mathrm{COT} \stackrel{\mathrm{def}}{=} \frac{E}{m \cdot g \cdot d}$$

Dimension-less quantity

Energy E to move a system of mass m over a distance d

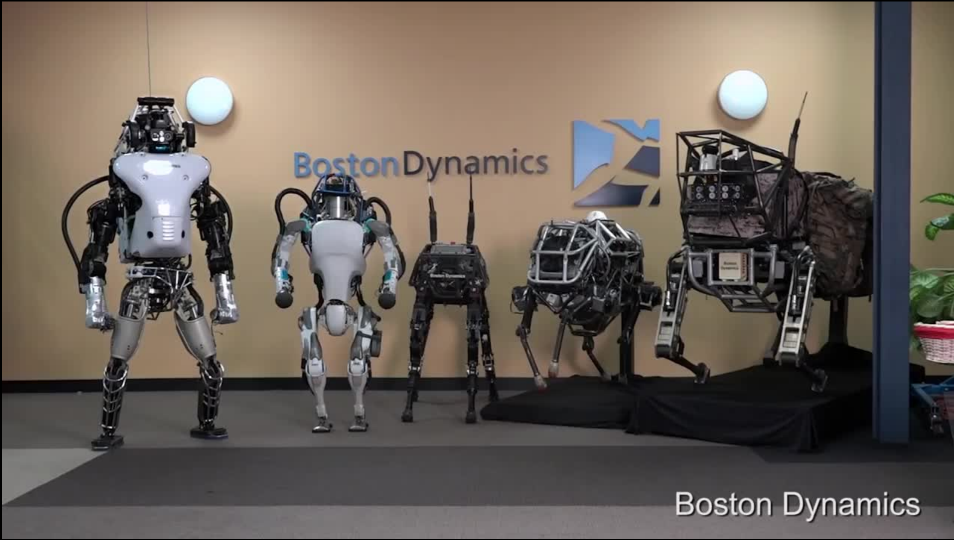


ACTUATION

ACTUATION

Hydraulics

- Pressurised liquid.
- Bigdog, Petman, Atlas


Boston Dynamics
Boston Dynamics

ACTUATION

Hydraulics

- Pressurised liquid.
- Bigdog, Petman, Atlas

Pneumatics

- Air powered: high energy, powerful.
- Inherently **compliant**.

ACTUATION

Hydraulics

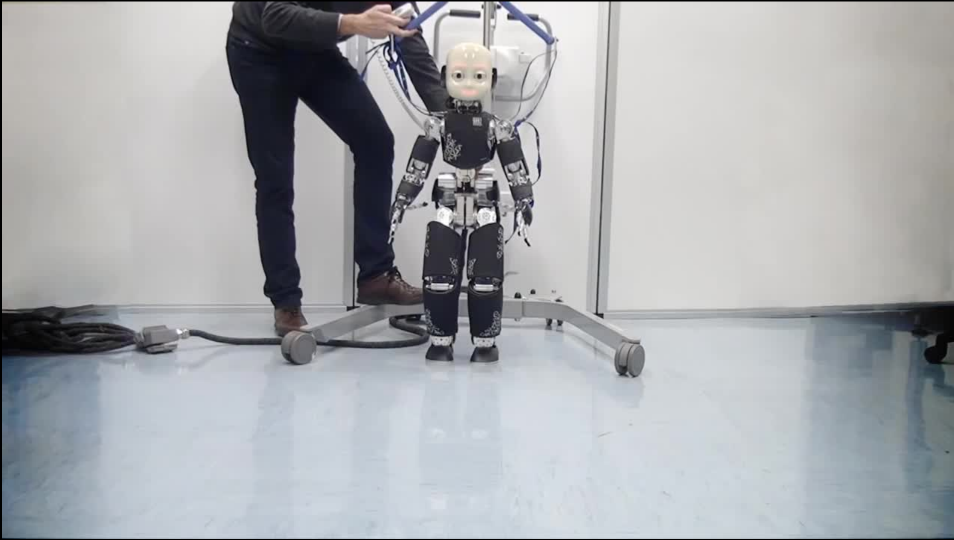
- Pressurised liquid.
- Bigdog, Petman, Atlas

Pneumatics

- Air powered: high energy, powerful.
- Inherently **compliant**.

DC motors

- Can be combined with spring to have compliant motors.
- iCub walking.



ACTUATION

Hydraulics

- Pressurised liquid.
- Bigdog, Petman, Atlas

Pneumatics

- Air powered: high energy, powerful.
- Inherently **compliant**.

DC motors

- Can be combined with spring to have compliant motors.
- iCub walking.

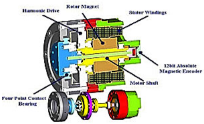
Servo motors

- DC motors with gears and positioning electronics

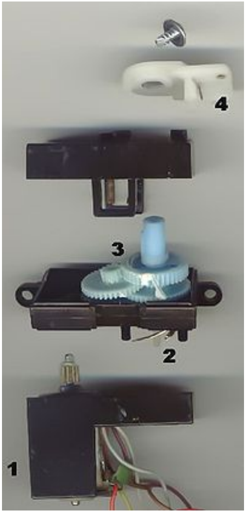
WHAT IS A SERVO MOTOR?

A servo motor is a combination of a DC motor, gearing and a position measurement.

Position measurement is used for negative feedback to decrease error between current position and demanded position.



iCub motor arrangement, can be used in servo mode



Source: RC servo motor (Wikipedia)

WHY SERVO MOTORS?

A large percentage of humanoid robots use servo motors

- Traditional method of powering a robot. The vast majority of industrial robots use motors to control the movement of the robots.
- Large knowledge base for servo motor control.
- Can be scaled for larger or smaller robots.
- Powered by electricity via a fixed or portable power supply.
- Accurate control of position.
- Large range of movement.
- Good power to weight ratio.
- Good speed characteristics.
- Cost effective solution.

HUMANOIDS AND SERVOS

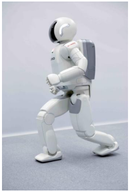
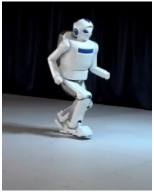
From industrial robots to humanoid robots

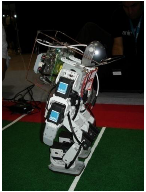
- Many of the first humanoid robots were developed from existing technology used in industrial robots.
- Car manufacturers use large numbers of industrial robots and were amongst the first companies to invest heavily in the development of new technologies to produce a humanoid robot.

HUMANOIDS AND SERVOS

Examples of humanoid robots using servo motors

- Honda Asimo
- Toyota humanoid
- Aldebaran Nao
- Bioloid
- Team Osaka football player




BIOLOID ROBOT

Bioloid Robot Specifications

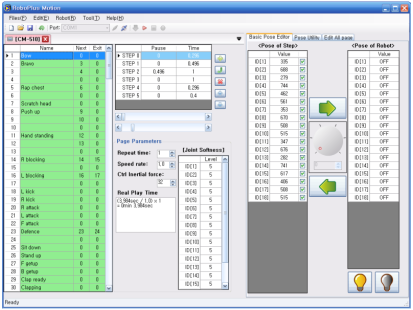
- Manufactured by Robotis in South Korea.
- Has 18 Degrees of Freedom (= motors).
- Designed to be reconfigured into different forms.
- Cost between £600 -£800.
- Robust.
- Easy to assemble.
- 40cm tall.
- Weighs 1.7Kg.
- Programmable.
- Safe.

BIOLOID ROBOT

Motors, brackets and comms module used for the Plymouth humanoids.

Can be programmed in two ways:

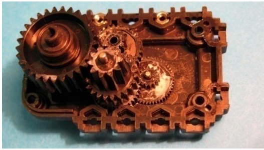
- Using an editor (RoboPlus)
- C (cross-compiled to Atmel processor assembly).



AX-12+ SERVO MOTOR

AX-12+ Specifications

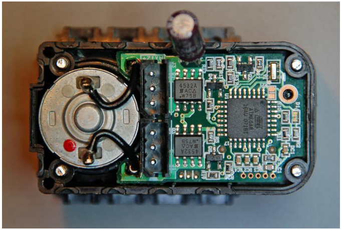
- Weight 55g
- Has a maximum speed of 50 RPM at 10V
- A maximum torque of 16.5 Kgf.cm or 1.62 Nm
- Has a gear ratio reduction of 1/254
- Operating angle of 300 degrees
- Uses plastic gears
- Programmable

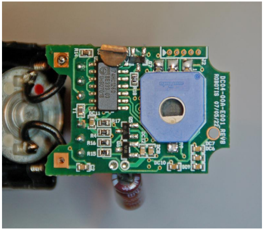


AX-12+ SERVO MOTOR

AX-12+ Servo Specifications

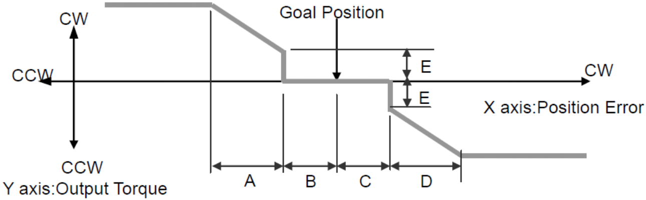
- AX-12 servo motor uses an AtmelMega8.
- Uses Serial RS232 communication at 128Hz or every 7.8ms.
- Calculate the required position of the motor then send the speed and position for the motor.





AX-12+ SERVO MOTOR

- Difficult to create exact controlled movements such as walking
- Takes a long time to manually enter the exact positions calculated for the robots
- Restricted by the performance of the AX-12 servo motors
- Compliance in the motors affects performance:



A : CCW Compliance Slope(Address0x1D)
B : CCW Compliance Margin(Address0x1B)
C : CW Compliance Margin(Address0x1A)
D : CW Compliance Slope (Address0x1C)
E : Punch(Address0x30,31)

KINEMATICS

HUMANOID KINEMATICS

Developing a robot gait: *What gait pattern results in a robust and fast walk?*

HUMANOID KINEMATICS

Developing a robot gait: ***What gait pattern results in a robust and fast walk?***

To answer to these questions the robot needs to be able to translate the movement in the Cartesian world coordinate system (x, y, z) to the angles of rotation for the motors (**joint space**).

- This is achieved by using inverse kinematics
- the joint space of the Bioloid robot has 18 dimensions

Forward kinematics is when given an angle of the rotation of the motors the position of the foot can be calculated in world coordinates.

HUMANOID KINEMATICS

Developing a robot gait: ***What gait pattern results in a robust and fast walk?***

To answer to these questions the robot needs to be able to translate the movement in the Cartesian world coordinate system (x, y, z) to the angles of rotation for the motors (**joint space**).

- This is achieved by using inverse kinematics
- the joint space of the Bioloid robot has 18 dimensions
- some dimensions are less relevant for static walking

Forward kinematics is when given an angle of the rotation of the motors the position of the foot can be calculated in world coordinates.

HUMANOID KINEMATICS

Developing a robot gait: ***What gait pattern results in a robust and fast walk?***

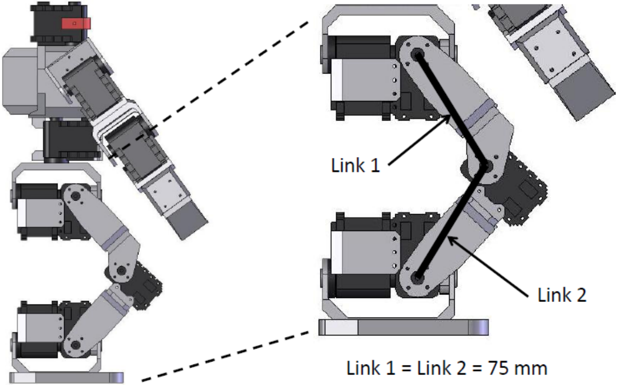
To answer to these questions the robot needs to be able to translate the movement in the Cartesian world coordinate system (x, y, z) to the angles of rotation for the motors (**joint space**).

- This is achieved by using inverse kinematics
- the joint space of the Bioloid robot has 18 dimensions
- some dimensions are less relevant for static walking
- however all are usually important for dynamic walking: **full body motion**

Forward kinematics is when given an angle of the rotation of the motors the position of the foot can be calculated in world coordinates.

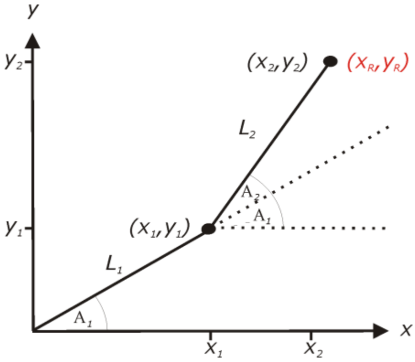
HUMANOID KINEMATICS

Simplified Bioloid kinematics:



BIOLOID FORWARD KINEMATICS

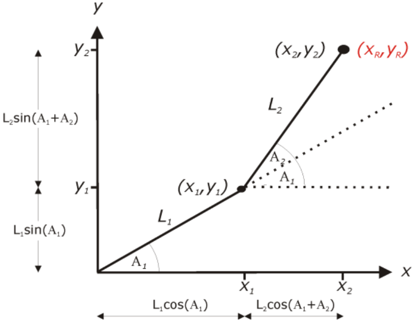
- Forward kinematics
- A_1 and A_2 are the angle of rotation, L_1 and L_2 are the length of the robot limbs
- To calculate the resultant position (x_R, y_R), first determine the X and Y component of each link and add these together.



$$\sin A = \frac{opposite}{hypotenuse}$$
$$\cos A = \frac{adjacent}{hypotenuse}$$

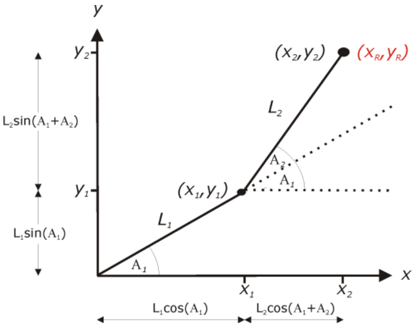
BIOLOID FORWARD KINEMATICS

$$x_R = L_1 \cos(A_1) + L_2 \cos(A_1 + A_2)$$
$$y_R = L_1 \sin(A_1) + L_2 \sin(A_1 + A_2)$$



BIOLOID FORWARD KINEMATICS

$$x_R = L_1 \cos(A_1) + L_2 \cos(A_1 + A_2)$$
$$y_R = L_1 \sin(A_1) + L_2 \sin(A_1 + A_2)$$

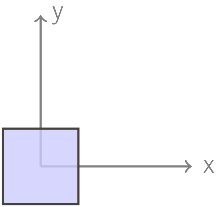


How does that generalize to more degrees of freedom (DoF)?

COMPOSITION OF TRANSFORMATIONS

Computation of the forward kinematics can be seen as a composition of transformations:

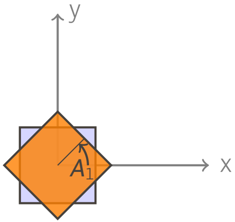
$\rightarrow$ to go from $(0, 0)$ to (x_R, y_R):



COMPOSITION OF TRANSFORMATIONS

Computation of the forward kinematics can be seen as a composition of transformations:

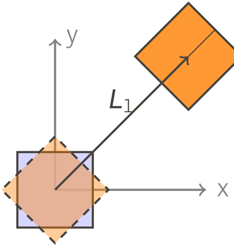
$\rightarrow$ to go from $(0,0)$ to (x_R, y_R): first rotation by A_1 around z



COMPOSITION OF TRANSFORMATIONS

Computation of the forward kinematics can be seen as a composition of transformations:

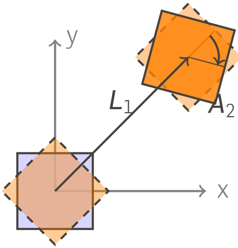
$\rightarrow$ to go from $(0,0)$ to (x_R, y_R): first rotation by A_1 around z , then translation by L_1 along x



COMPOSITION OF TRANSFORMATIONS

Computation of the forward kinematics can be seen as a composition of transformations:

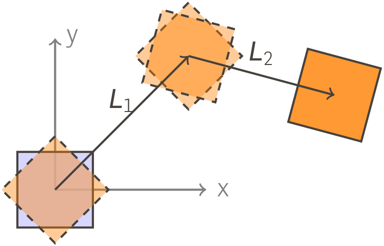
$\rightarrow$ to go from $(0, 0)$ to (x_R, y_R): first rotation by A_1 around z , then translation by L_1 along x, then rotation by A_2 around z



COMPOSITION OF TRANSFORMATIONS

Computation of the forward kinematics can be seen as a composition of transformations:

$\rightarrow$ to go from $(0, 0)$ to (x_R, y_R): first rotation by A_1 around z , then translation by L_1 along x, then rotation by A_2 around z, then translation by L_2 along x



To compose transformations is to multiply their matrices

MATRIX FORM

2D transformation matrices:

$$R(\alpha) = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad T\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} 1 & 0 & x \\ 0 & 1 & y \\ 0 & 0 & 1 \end{bmatrix}$$

MATRIX FORM

2D transformation matrices:

$$R(\alpha) = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad T\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} 1 & 0 & x \\ 0 & 1 & y \\ 0 & 0 & 1 \end{bmatrix}$$

In the case of a planar kinematics (i.e., 2D kinematics), the resulting forward transformation *FK* is:

$$FK(A_i, L_i) = \prod R_i(A_i) \cdot T_i(L_i)$$

MATRIX FORM

2D transformation matrices:

$$R(\alpha) = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad T\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} 1 & 0 & x \\ 0 & 1 & y \\ 0 & 0 & 1 \end{bmatrix}$$

In the case of a planar kinematics (i.e., 2D kinematics), the resulting forward transformation *FK* is:

Joint matrix, noted Z_i in the general, 3D case

$$FK(A_i, L_i) = \prod R_i(A_i) \cdot T_i(L_i)$$

Link matrix, noted X_i in the general, 3D case

MATRIX FORM

2D transformation matrices:

$$R(\alpha) = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad T(\begin{bmatrix} x \\ y \end{bmatrix}) = \begin{bmatrix} 1 & 0 & x \\ 0 & 1 & y \\ 0 & 0 & 1 \end{bmatrix}$$

In the case of a planar kinematics (i.e., 2D kinematics), the resulting forward transformation *FK* is:

$$\begin{aligned} FK(A_i, L_i) &= \prod R_i(A_i) \cdot T_i(L_i) \\ &= R(A_1) \cdot T(L_1) \cdot R(A_2) \cdot T(L_2) \\ &= \begin{bmatrix} \cos A_1 & -\sin A_1 & 0 \\ \sin A_1 & \cos A_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & L_1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos A_2 & -\sin A_2 & 0 \\ \sin A_2 & \cos A_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & L_2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \cos(A_1 + A_2) & -\sin(A_1 + A_2) & L_1 \cos(A_1) + L_2 \cos(A_1 + A_2) \\ \sin(A_1 + A_2) & \cos(A_1 + A_2) & L_1 \sin(A_1) + L_2 \sin(A_1 + A_2) \\ 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

MATRIX FORM

2D transformation matrices:

$$R(\alpha) = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad T(\begin{bmatrix} x \\ y \end{bmatrix}) = \begin{bmatrix} 1 & 0 & x \\ 0 & 1 & y \\ 0 & 0 & 1 \end{bmatrix}$$

In the case of a planar kinematics (i.e., 2D kinematics), the resulting forward transformation *FK* is:

$$\begin{aligned} FK(A_i, L_i) &= \prod R_i(A_i) \cdot T_i(L_i) \\ &= R(A_1) \cdot T(L_1) \cdot R(A_2) \cdot T(L_2) \\ &= \begin{bmatrix} \cos A_1 & -\sin A_1 & 0 \\ \sin A_1 & \cos A_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & L_1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos A_2 & -\sin A_2 & 0 \\ \sin A_2 & \cos A_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & L_2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \cos(A_1 + A_2) & -\sin(A_1 + A_2) & L_1 \cos(A_1) + L_2 \cos(A_1 + A_2) \\ \sin(A_1 + A_2) & \cos(A_1 + A_2) & L_1 \sin(A_1) + L_2 \sin(A_1 + A_2) \\ 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

MATRIX FORM (2)

Homogenous coordinates: we need them to use transformation matrices.

What is the position of the origin $\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$?

MATRIX FORM (2)

Homogenous coordinates: we need them to use transformation matrices.

What is the position of the origin $\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$?

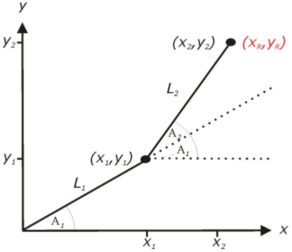
$$
\begin{aligned}
FK(A_i, L_i) \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} &= \begin{bmatrix} \cos(A_1 + A_2) & -\sin(A_1 + A_2) & L_1 \cos(A_1) + L_2 \cos(A_1 + A_2) \\ \sin(A_1 + A_2) & \cos(A_1 + A_2) & L_1 \sin(A_1) + L_2 \sin(A_1 + A_2) \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} L_1 \cos(A_1) + L_2 \cos(A_1 + A_2) \\ L_1 \sin(A_1) + L_2 \sin(A_1 + A_2) \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} y_R \\ y_R \\ 1 \end{bmatrix}
\end{aligned}
$$

INVERSE KINEMATICS

$$A_2 = \cos^{-1}\left(\frac{x_R^2 + y_R^2 - L_1^2 - L_2^2}{2L_1L_2}\right)$$

$$A_1 = \sin^{-1}\left(\frac{y_r}{\sqrt{x_R^2 + y_R^2}}\right) - \sin^{-1}\left(\frac{L_2 \sin(A_2)}{\sqrt{x_R^2 + y_R^2}}\right)$$

This is a geometric solution, more can be found in Mathematics required for legged robotic motion



GAIT

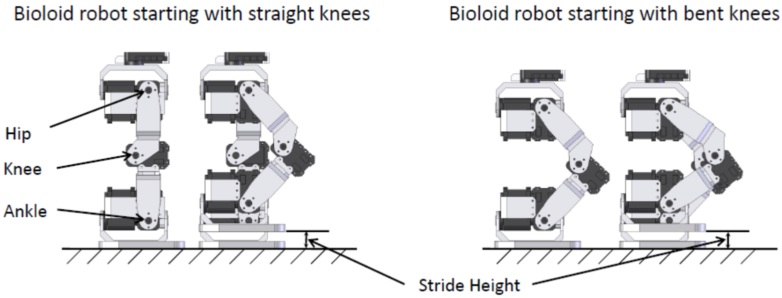
SERVO-BASED HUMANOIDS

What do servo powered humanoid robots have in common?

- They never straighten their legs!
- Most servo powered robots either walk or run with their legs slightly bent
- Produces an non-human like gait
- Same for either statically of dynamically balanced robots
- Results from the physical restrictions of using servo motors

BIOLOID GAIT ANALYSIS

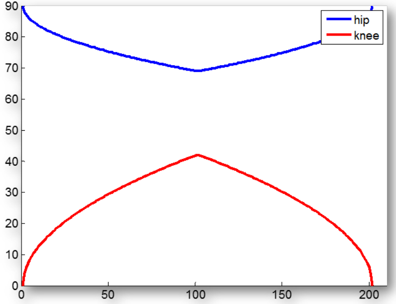
Starting with slightly bent knees means the knee servo motor is used more effectively resulting in the foot being lifted a greater distance and in a more controlled manner.



EFFORT TO LIFT FOOT 1CM?

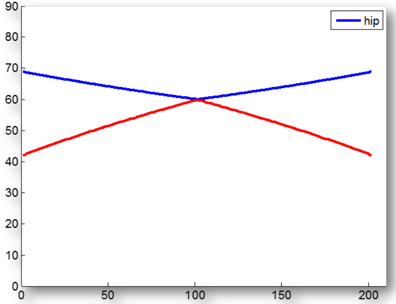
Stretched leg

- Hip moves 21.0°, knee moves 42.0°



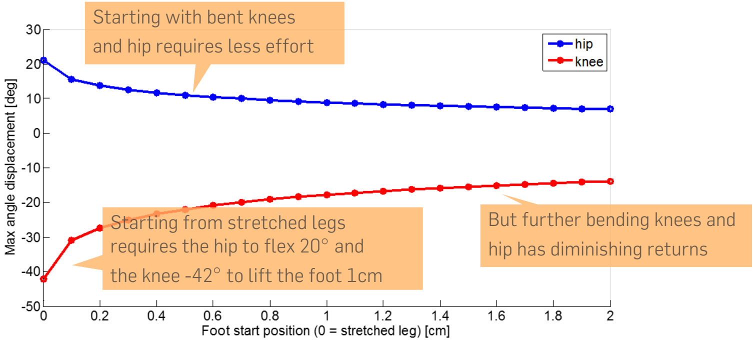
Bent knees

- Hip moves 8.9°, knee moves 17.8°



→ Much less angle change required for motors to lift foot same amount.

EFFORT TO LIFT FOOT 1CM?



BIOLOID GAIT ANALYSIS

What are the implications for the Bioloid robot gait?

How often are the servo motors asked to move?

- Usually at 64Hz or every 15.6ms.

Therefore what is the maximum angle they can rotate in that time.

- At 50 RPM = $300^\circ/s \rightarrow 4.68^\circ$ in 15.6 ms

As the height the foot is raised and the forward movement of the foot is increased the servo motors reach the limit of their speed.

Above the maximum speed the gait will not be able to reach its calculated position. This could result in instability.

BIOLOID GAIT ANALYSIS

Using a simple inverse kinematic solution for the Bioloid leg servos, the angle of rotation for a step height and length can be calculated.

Given the update rate of the servo motors it is possible to calculate what is the maximum height and distance the foot can be moved. This gives a maximum possible theoretical speed for the robot to move.

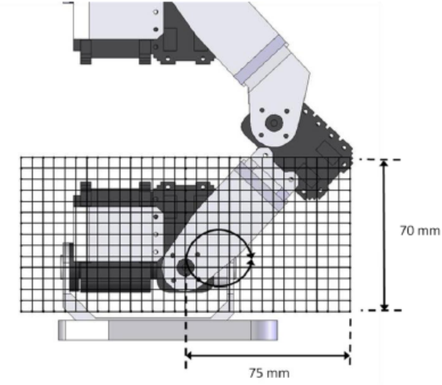
To improve the gait different motors speeds can be assessed.

The gait generation pattern can be optimised for the motors.

- Currently the Bioloid uses the equation of an ellipse to generate the gait pattern to follow.

BIOLOID GAIT ANALYSIS

In order to reduce the computational cost to calculate the equation of an ellipse every 15.6ms, the solution is pre-calculated over a grid (75mm x 70mm) and stored in a look-up table.

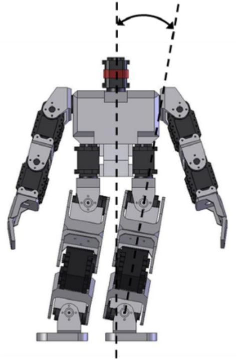


BIOLOID DYNAMIC GAIT GENERATOR

- There is also a Bioloid dynamic gait generator, which takes 10 parameters to generate a series of robot poses.

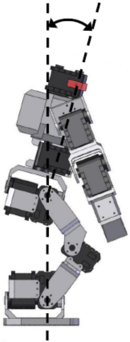
GAIT GENERATOR – PARAMETERS (1)

- Swing – Hip movement (Sideways)



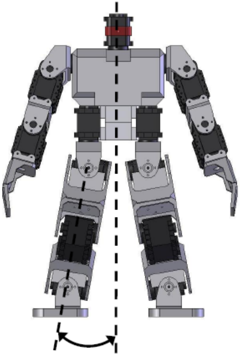
GAIT GENERATOR – PARAMETERS (2)

- Tilt – Hip (used to balance the robot)



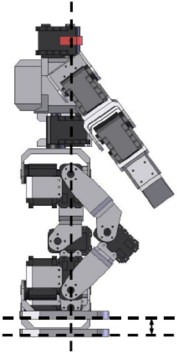
GAIT GENERATOR – PARAMETERS (3)

- Camber (Splaying legs)



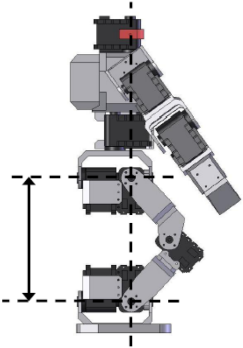
GAIT GENERATOR – PARAMETERS (4)

- Stride height (Foot lift)



GAIT GENERATOR – PARAMETERS (5)

- Y-Offset (Starting position for knees)



THEORIES OF WALKING

SUPPORT POLYGON

The **support polygon** is a horizontal region over which the center of mass must lie to achieve static stability.

SUPPORT POLYGON

The **support polygon** is a horizontal region over which the center of mass must lie to achieve static stability.

For example, for an object resting on a horizontal surface (e.g. a table), the support polygon is the **convex hull of its ”footprint”** on the table.

SUPPORT POLYGON

The **support polygon** is a horizontal region over which the center of mass must lie to achieve static stability.

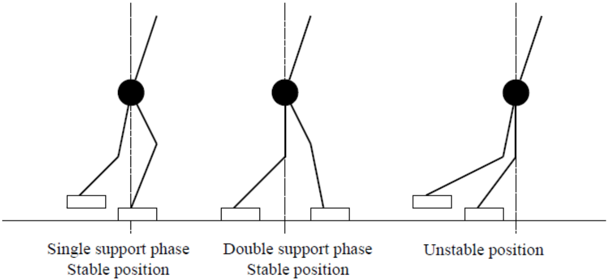
For example, for an object resting on a horizontal surface (e.g. a table), the support polygon is the **convex hull of its ”footprint”** on the table.

For bipedal robots, the support polygon is **the convex hull of the contact points with the ground**.

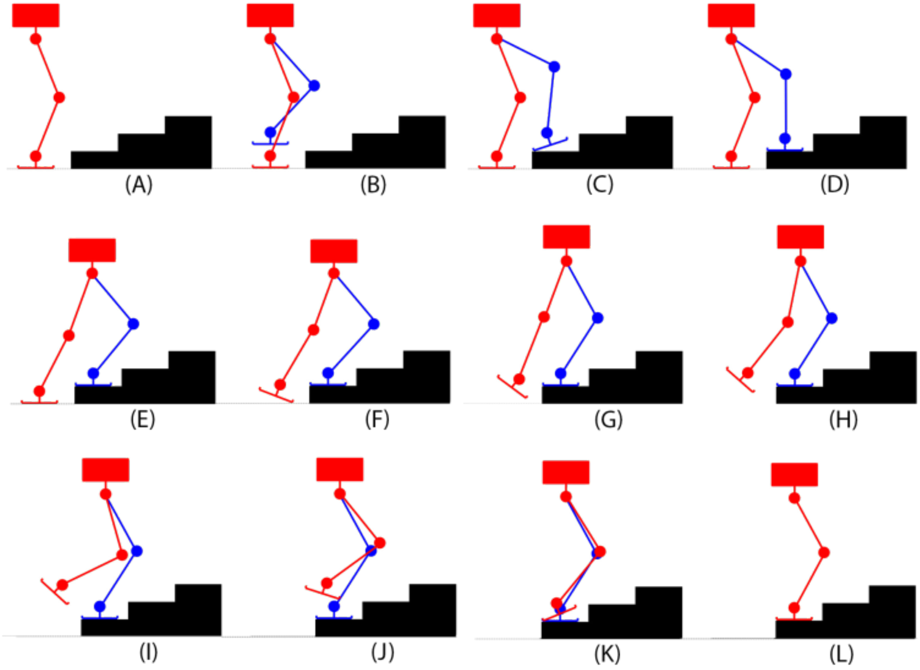
- Might be a single foot!
- Changes over time

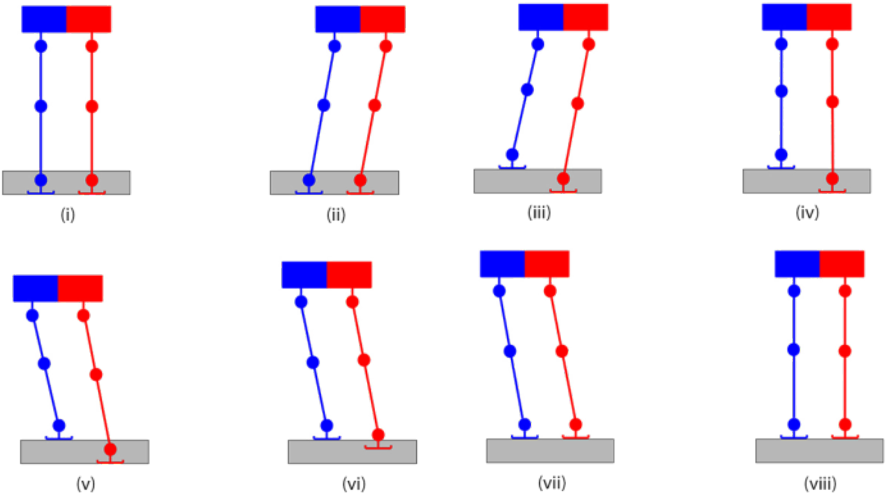
STATIC WALKING

Based on keeping **Centre of Gravity** (COG) **over the support polygon**



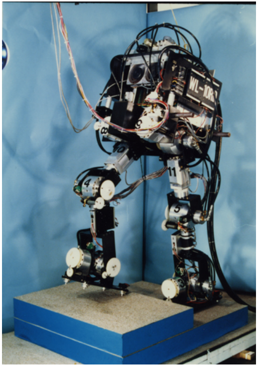
Source: Zaldivar Navarro, 'A Biped Robot Design'


(A)
(B)
(C)
(D)
(E)
(F)
(G)
(H)
(I)
(J)
(K)
(L)


(i)
(ii)
(iii)
(iv)
(v)
(vi)
(vii)
(viii)

DYNAMIC WALKING

Based on keeping **Zero Moment Point** (ZMP) **over the support polygon**



WL-10RD. First dynamic walker using the ZMP scheme (1985).

Honda Asimo, uses ZMP scheme for walking as deduced from papers and patents(2000-).

REQUIREMENTS OF ZMP BIPEDS

The fundamental requirements for ZMP-based walking

- At least six fully actuated joints for each leg: hip (3), knee (1), foot (2)
- Joint are position controlled (as opposed to speed)
- Feet are equipped with force sensors, used to measure ZMP.

FORCE AND MOMENT ON A BODY

The resultant force of inertia and gravity forces is:

$$\vec{F}^{gi} = m \cdot \vec{g} - m \cdot \vec{a_G}$$

m is total mass of robot, g is gravitational acceleration ($9.81 m/s^2$), a_G is acceleration of centre of mass

The moment (aka torque) in any point X is:

$$\vec{M}_X^{gi} = \vec{XG} \times m \cdot \vec{g} - \vec{XG} \times m \cdot \vec{a_G} - \dot{\vec{H}}_G$$

$\vec{XG}$ is the vector between point X and centre of mass G and $\dot{H}_G$ is rate of angular momentum at G (amount of rotation)

FORCE AND MOMENT INDUCED BY CONTACT

Euler's laws of motion:

$$\vec{F} = m \cdot \vec{a}_G$$
$$\vec{M}_X = \vec{XG} \times (m \cdot \vec{a}_G) + I \cdot \ddot{\vec{\alpha}}_G$$

FORCE AND MOMENT INDUCED BY CONTACT

Euler's laws of motion:

$$\vec{F} = m \cdot \vec{a}_G$$
$$\vec{M}_X = \vec{XG} \times (m \cdot \vec{a}_G) + I \cdot \ddot{\vec{\alpha}}_G$$

Resultant of the contact forces:

$$\vec{F}^c + m \cdot \vec{g} = m \cdot \vec{a}_G$$

Moment caused by contact forces:

$$\vec{M}_X^c + \vec{XG} \times m \cdot \vec{g} = \dot{\vec{H}}_G + \vec{XG} \times m \cdot \vec{a}_G$$

DYNAMIC BALANCE

- When forces and moments are opposite the body is dynamically balanced:

$$\vec{F}^{gi} + \vec{F}^{c} = 0$$
$$\vec{M}^{c}_{X} + \vec{M}^{gi}_{X} = 0$$

Read more here (if you dare)...

DYNAMIC BALANCE

- When forces and moments are opposite the body is dynamically balanced:

$$\vec{F}^{gi} + \vec{F}^{c} = 0$$
$$\vec{M}^{c}_{X} + \vec{M}^{gi}_{X} = 0$$

- Zero Moment Point (ZMP):

$$\vec{PZ} = \frac{\vec{n} \times \vec{M}^{gi}_{P}}{\vec{F}^{gi} \cdot \vec{n}}$$

n is normal vector (a vector perpendicular to a surface of length 1) to the support surface. ***Z*** is the Zero Moment Point, ***P*** is the resultant point of contact of the foot.

Read more here (if you dare)...

ZMP: THE PROBLEM

In theory, it works. In practice, it doesn't always.

Requires a precise knowledge of where the Centre of Gravity is, what the forces are acting on the point of contact, and what the rate of angular momentum ($\dot{H}_G$) of the robot is.

- In practice, this can be done for simple systems (inverted pendulum walkers).
- Is impossible for real, complex robots.

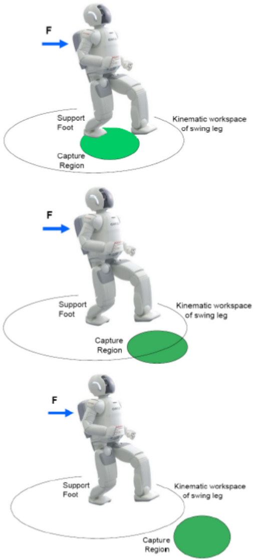
CAPTURE STEPS

A promising way forward is to use “capture steps”.

- A capture step places the foot in the “capture region” making the robot come to a standstill.
- If a robot is **pushed** it places one foot in the direction of the fall.

If a Capture Point is situated within the convex hull of the foot support area (the Base of Support), the robot is able to recover from the push without having to take a step. Otherwise, it must take a step.

CAPTURE STEPS



- The Capture Region intersects the Base of Support, no step is needed.
- The Capture Region falls outside the Base of Support, a step is needed to prevent falling over.
- Same as before, but the Capture Region falls outside the reachable space. More than one step is needed to regain balance.


regiocom
GERMAN OPEN

CAPTURE STEPS EXAMPLES

Fast optimisation of capture steps using a dynamic simulated model

- Video and paper

Walking using capture step

- Video and paper

THE HARDEST PROBLEM IN ROBOTICS?

WHY IS WALKING HARD FOR ROBOTS?

Number of possible gaits is enormous

- Finding a gait that works (a “robust” gate) is hard.
- One set of parameter settings will typically work for only one type of surface under one set of conditions.

Fun example: genetic algorithms to find the right parameters

WHY IS WALKING HARD FOR ROBOTS?

Number of possible gaits is enormous

- Finding a gait that works (a “robust” gate) is hard.
- One set of parameter settings will typically work for only one type of surface under one set of conditions.

Fun example: genetic algorithms to find the right parameters

Floor reaction control is very hard Human feet sense and change to the structure of the surface. Robots don’t yet.

WHY IS WALKING HARD FOR ROBOTS?

Number of possible gaits is enormous

- Finding a gait that works (a "robust" gate) is hard.
- One set of parameter settings will typically work for only one type of surface under one set of conditions.

Fun example: genetic algorithms to find the right parameters

Floor reaction control is very hard Human feet sense and change to the structure of the surface. Robots don't yet.

Sensing for walking challenging Gyroscopes, accelerometers, cameras, … we do not yet understand how to use their readings to make robust gaits.

WHY IS WALKING HARD FOR ROBOTS?

Number of possible gaits is enormous

- Finding a gait that works (a "robust" gate) is hard.
- One set of parameter settings will typically work for only one type of surface under one set of conditions.

Fun example: genetic algorithms to find the right parameters

Floor reaction control is very hard Human feet sense and change to the structure of the surface. Robots don't yet.

Sensing for walking challenging Gyroscopes, accelerometers, cameras, ... we do not yet understand how to use their readings to make robust gaits.

Power requirements are high

WHY IS WALKING HARD FOR ROBOTS?

Number of possible gaits is enormous

- Finding a gait that works (a "robust" gate) is hard.
- One set of parameter settings will typically work for only one type of surface under one set of conditions.

Fun example: genetic algorithms to find the right parameters

Floor reaction control is very hard Human feet sense and change to the structure of the surface. Robots don't yet.

Sensing for walking challenging Gyroscopes, accelerometers, cameras, ... we do not yet understand how to use their readings to make robust gaits.

Power requirements are high

Robots do not walk in isolation...

POSSIBLE WAYS FORWARD

Closed-loop walking

- Sensors (pressure sensors, IMU, camera, ...) in a fast loop report back on the state of robot, can be used to respond to changes.
- Problem 1: measuring on a moving robot is hard.
- Problem 2: even if you know what the robot is doing (e.g. falling), what to do?

POSSIBLE WAYS FORWARD

Letting the robot learn how to walk

- Difficult to do in the “real world”
- Learning or evolving in a physics simulator, and then transferring to the real world
- Problem: the “reality gap” is too big for unstable robot. Can work for 2+-legged robots (example), but has not been possible yet for bipeds

That's all, folks!

Questions:
Portland Square B316 or **severin.lemaignan@plymouth.ac.uk**

Slides:
github.com/severin-lemaignan/module-mobile-and-humanoid-robots