Cognitive Robotics

Motion (part 1)

Hans-Dieter Burkhard, Marija Brkic Bakaric
October 2014

Outline

Introduction

Kinematics of Poses

Kinematics of Drive Systems

Trajectories

Motion Planning

Motion Control

Motions of Legged Robots

Optimization/Learning of Motions

Biologically Inspired Motions

Motion

Motion:

Change of position(s) by certain actions/skills,

e.g. for locomotion or manipulation.

Great variety of natural and technical systems

Formal description by mechanics

(force, mass, displacement, velocity, acceleration)

Problems in Robotics:

How can motions be realized and controlled

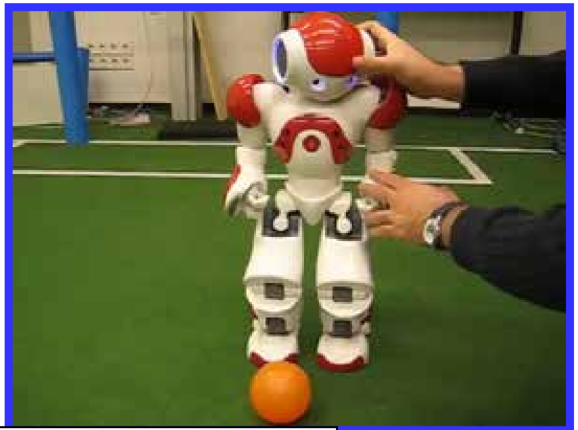
(hardware, software)

Cars

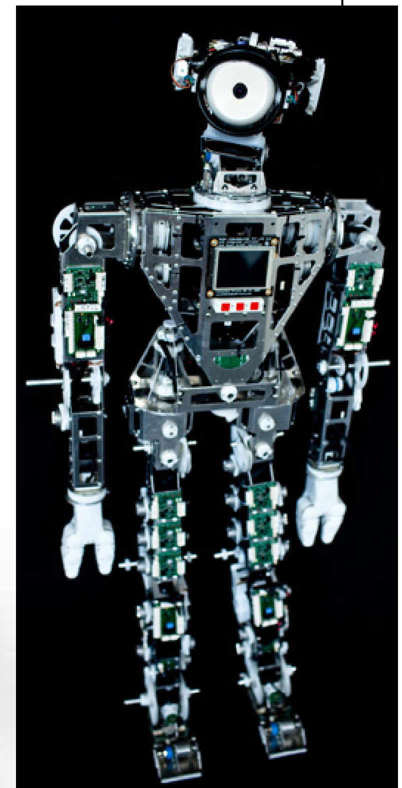
"MadeInGermany"
Autonomos Labs (R.Rojas, FU Berlin)
https://www.youtube.com/watch?v=nX-Ie6JSU5g

How many degrees of freedom?
Which poses can be reached?

Humanoid Robots



Nao (Aldebaran)



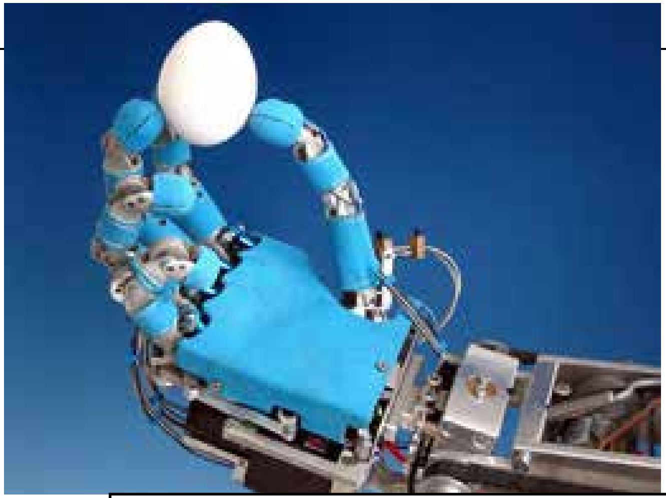
Myon
(Dr. Manfred Hild,
Neurorobotics Lab
Humboldt University)

How many degrees of freedom?
Which poses can be reached?

Flying Robots

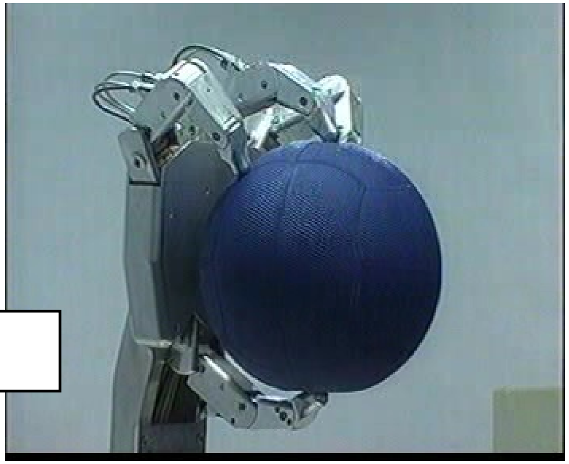
Cognitive Robotics Lab
Prof. Verena Hafner, HU

Manipulators



HU Berlin (Torsten Siedel)

Karlsruhe



Aachen

Manipulators

Surgical Robot DaVinci
Photo Nader Moussa
(WikiMedia)

Feeding Robot
Bestic AB (Stockholm)
Photo: Alice Öberg
(Sweden.se)

Manipulators

Photo: NASA (WikiMedia)

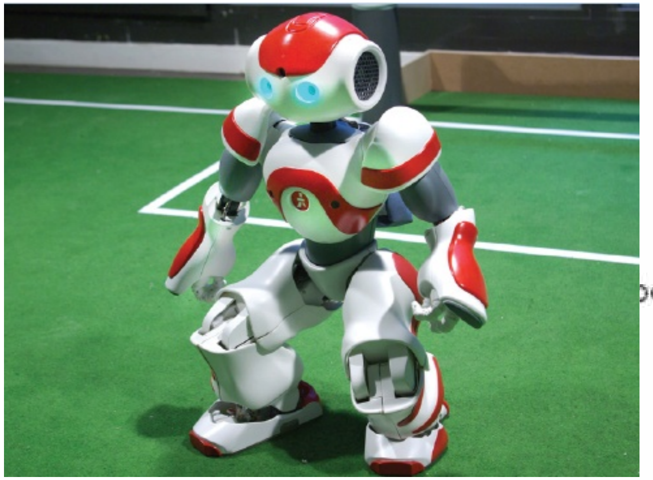
Strawberry Harvesting Robot by Robotic Harvesting LLC

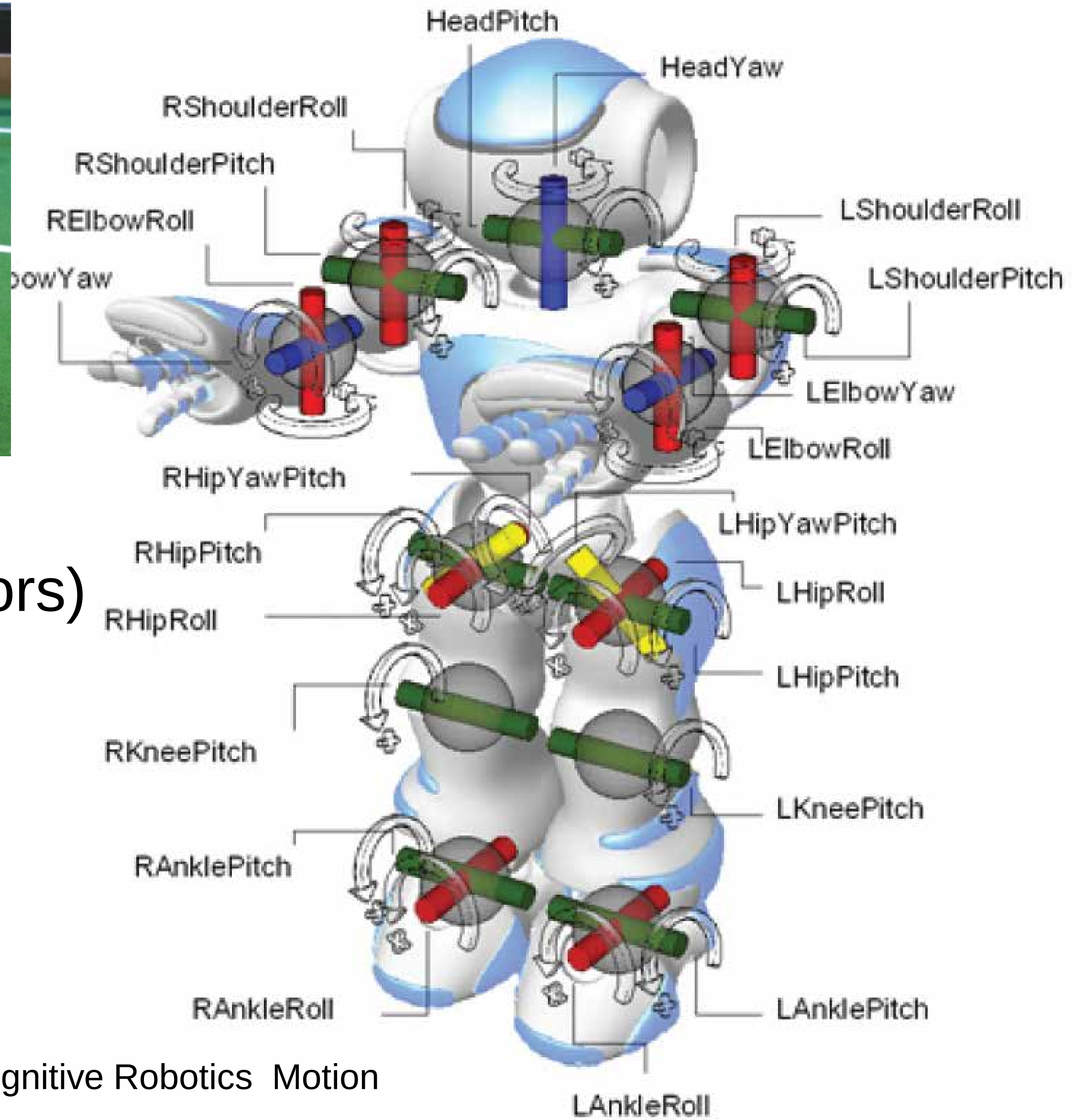
Joints

- Active: control with motors, pulleys, ...
- Problem: loading of gear axes
- Passive: Adaptation

Maintaining rest position by drives, gravity, friction, preload,…

Joints of Nao from Aldebaran

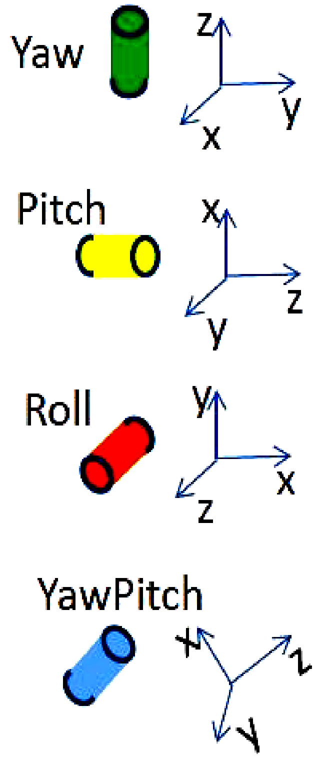


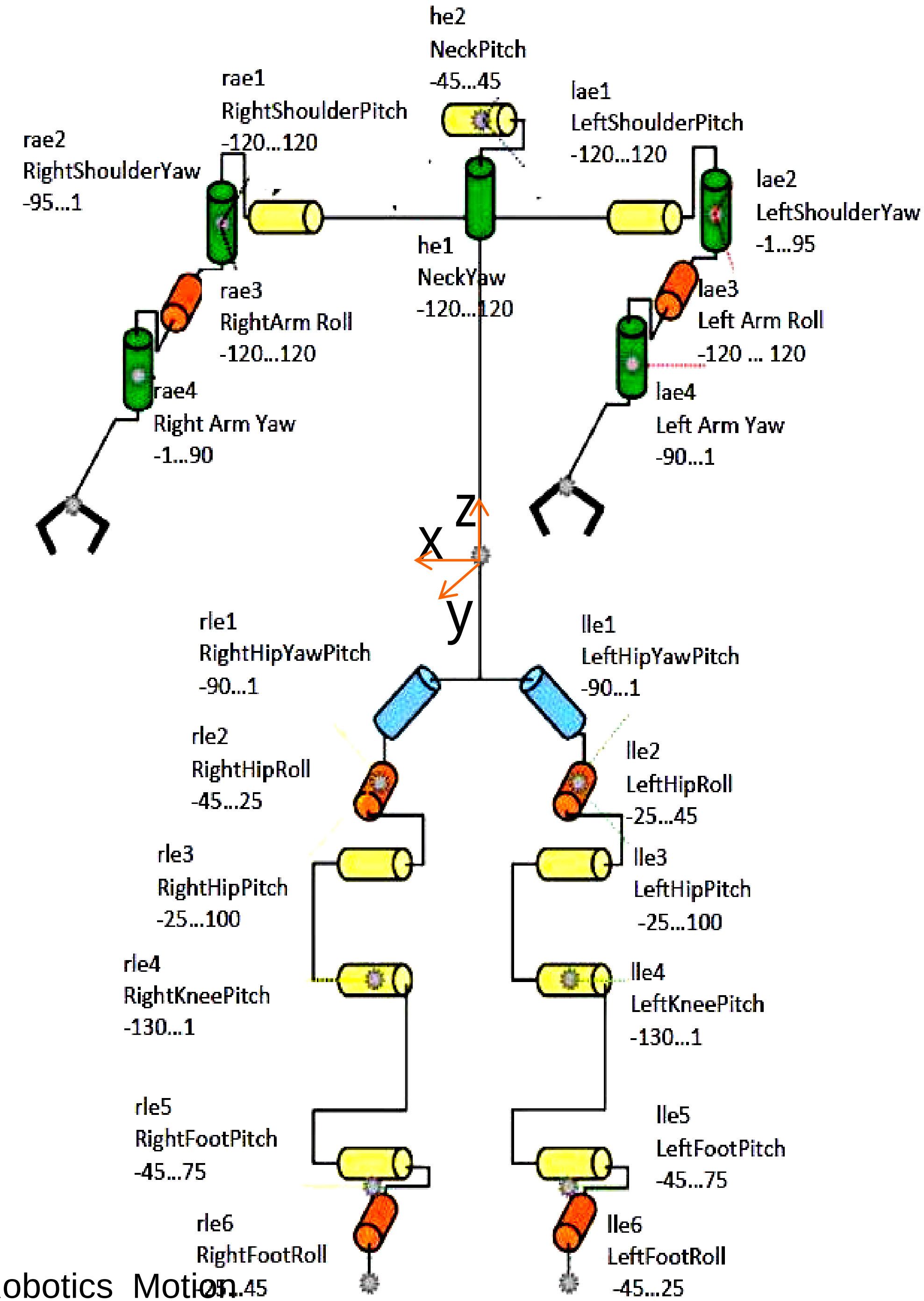


21 active DOF (motors)

- 2 head
- 4 per arm
- 5 per leg
- 1 hip

Nao in Simulation





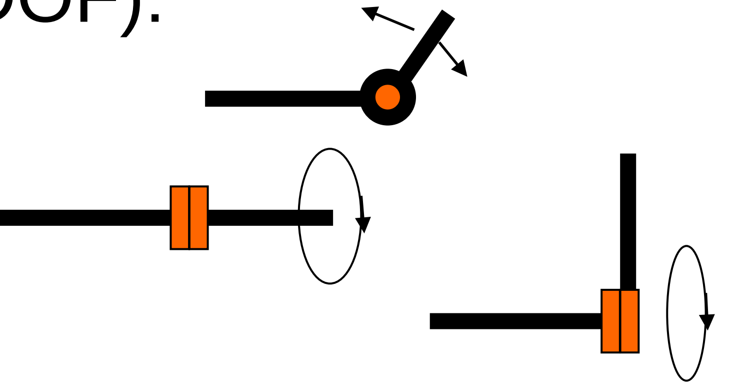
22 active DOF (motors):

- 2 head
- 4 per arm
- 5 per leg
- 2 hip

1 DOF Joints in Technique

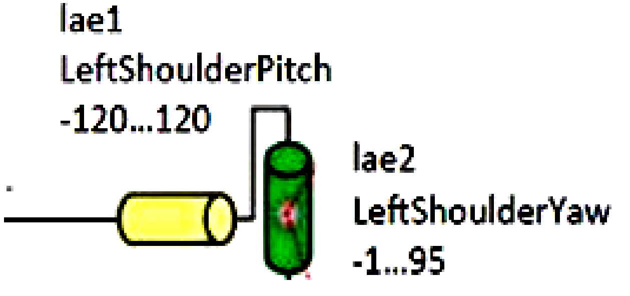
With 1 degree of freedom (DOF):

- rotation joint
- torsion joint
- revolver joint

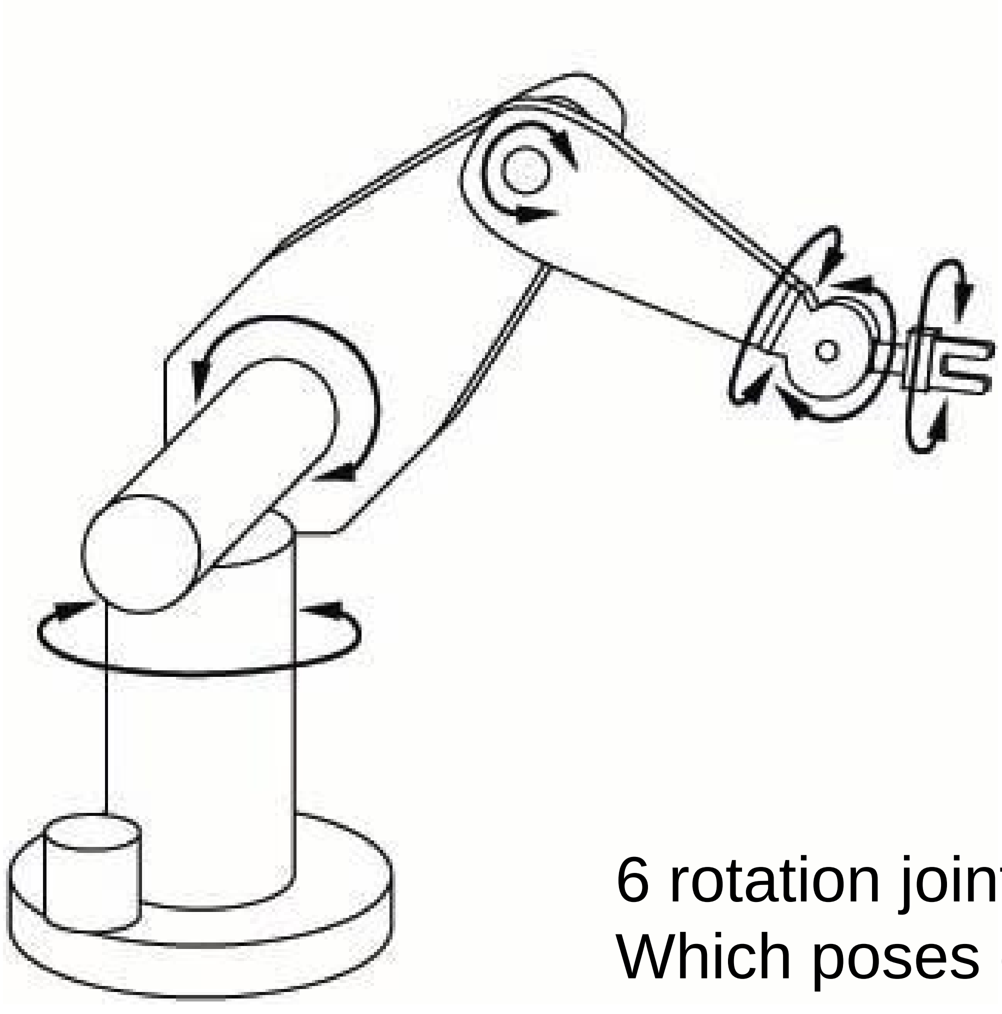


- Linear joint (Translation joint., prismatic joint)

Several DOF by combinations



Puma (Programmable Universal Manipulation Arm)



6 rotation joints (6 DOF)
Which poses can be reached?

Degrees of Freedom (DOF)

DOF is the

- minimal number m of parameters $p_1,\ldots,p_m$ for complete description

equivalently:

- maximal number m of independent parameters $p_1,\ldots,p_m$

Degrees of Freedom (DOF)

DOF of poses

(= parameters for complete description in work space):

- point on plan p=(x,y) , 2 DOF (2 position)
- car on plane: p=(x,y,q), 3 DOF (2 position, 1 orientation)
- airplane: p=(x,y,z, f ,Y , q), 6 DOF (3 position, 3 orientation)

DOF of control parameters (in control/configuration space):

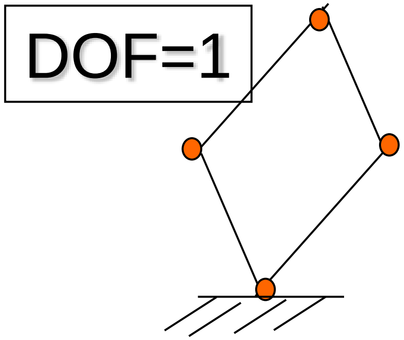
independently movable parts (joints, wheels/axes, …)

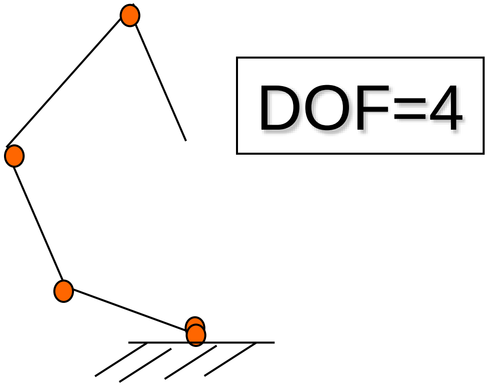
DOF of control may be *active* (actuated) or *passiv*

Degrees of Freedom (DOF)

Reachable poses depend on morphology and environment

Constraints $C(p_1,\ldots p_m) = 0$ for parameters may reduce DOF





Outline

Introduction

Kinematics of Poses

Kinematics of Drive Systems

Trajectories

Motion Planning

Motion Control

Motions of Legged Robots

Optimization/Learning of Motions

Biologically Inspired Motions

Kinematics of Poses

Kinematics (forward kinematics):

- What is the pose?

Inverse kinematics (reverse kinematics):

- How to set the pose?

Simplification in Kinematics:
Neglect mass and force

Work Space and Configuration Space

Work space: „Relevant“ environment of the robot or some part.

Pose $p=(p_1,\ldots,p_m)$:
Position/orientation of the robot or some part in work space
(e.g. the pose of an end effector, of a camera etc.).

m = DOF of the pose in Workspace

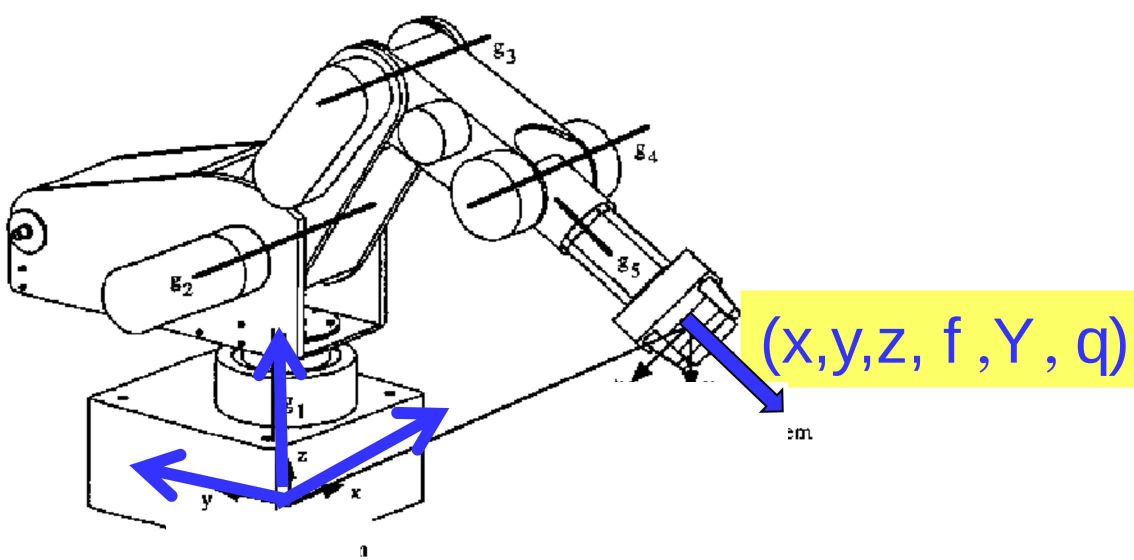
End effector of an industrial robot:

p=(x,y,z, f ,Y , q)

6 DOF:

Ø 3 position,

Ø 3 orientation



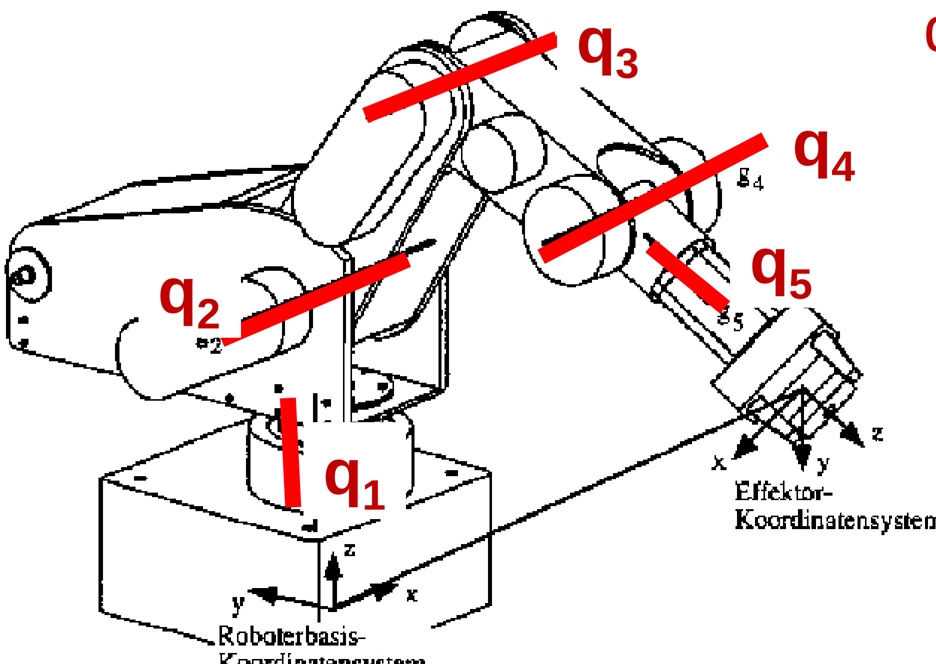
Work Space and Configuration Space

Configuration space:

Configuration $q=(q_1,\ldots,q_n)$: parameters of joints etc.

„generalized coordinates", „control parameters"

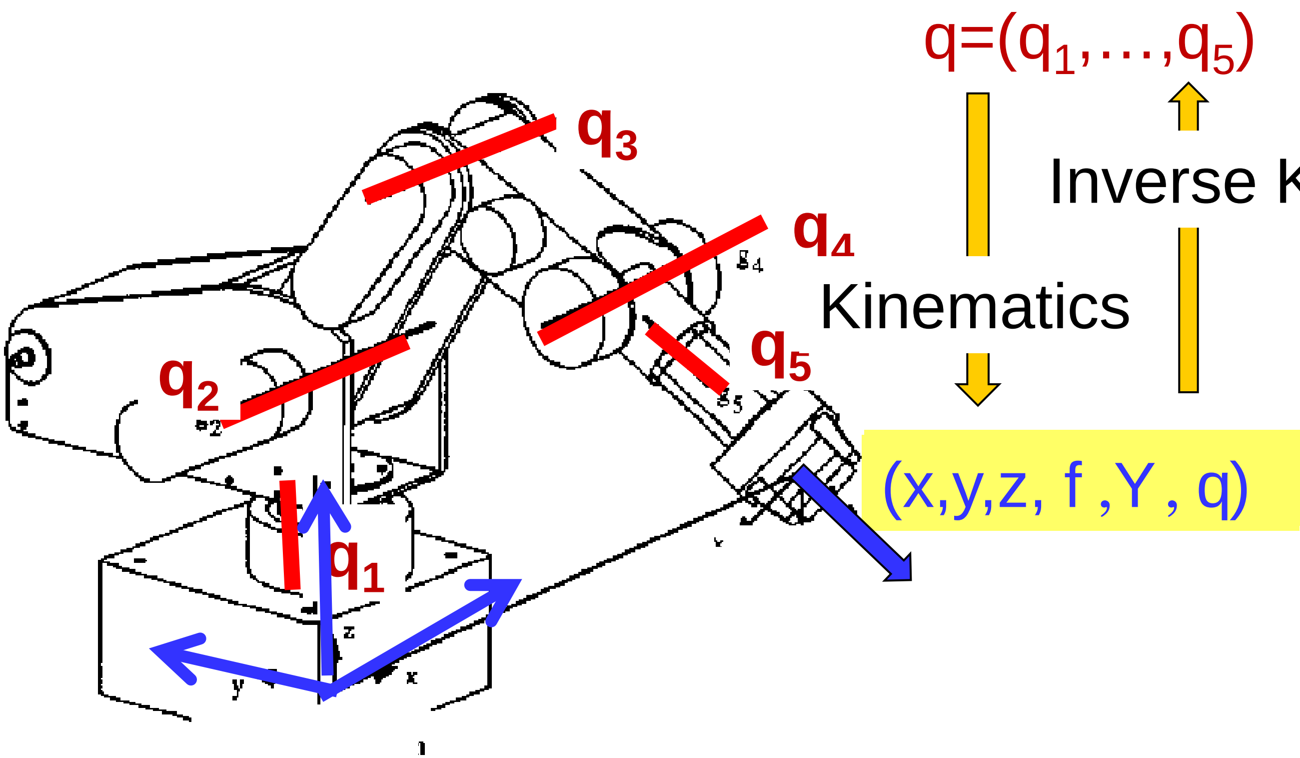
n = DOF in configuration space



$q=(q_1,\ldots,q_5)$

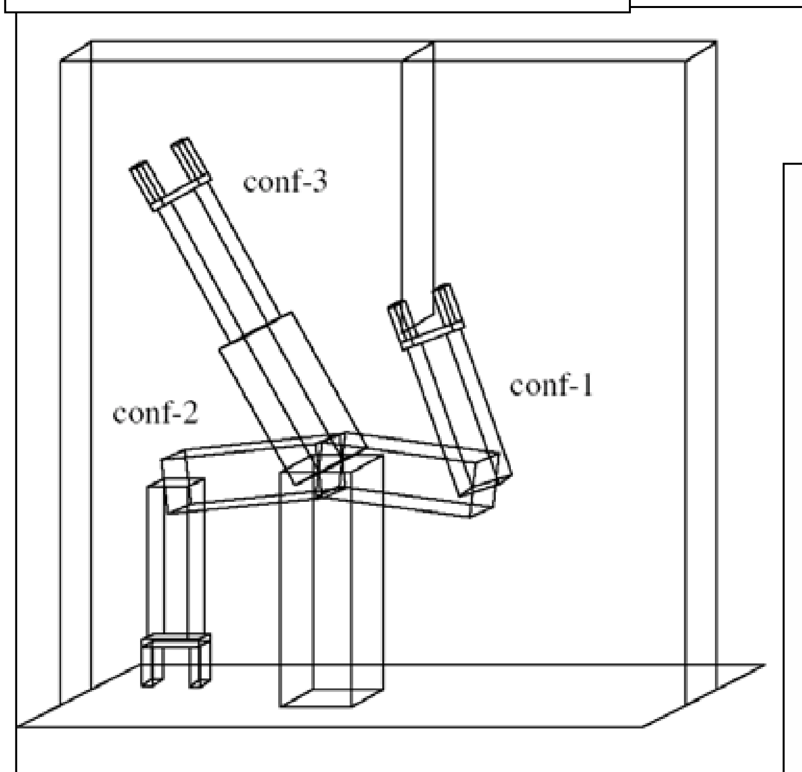
5 DOF

Work Space and Configuration Space

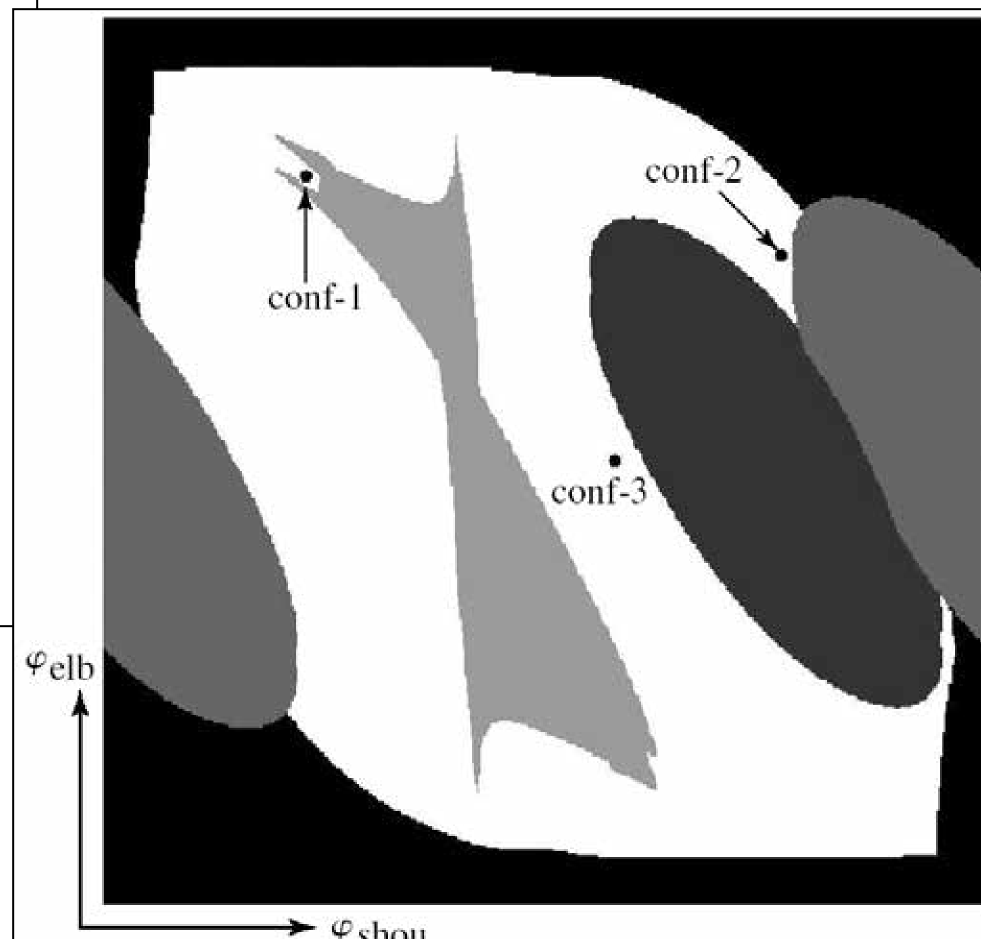


Work Space and Configuration Space

Work Space



Configuration Space



From Russell/Norvig: Artificial Intelligence

Work Space and Configuration Space

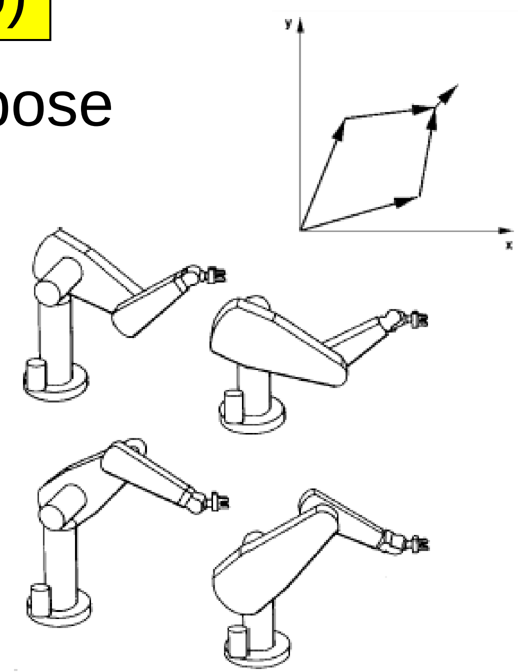
Kinematics: $p = f(q)$

Determine pose from configuration

– Configuration determines pose uniquely

Inverse Kinematics: $q = f^{-1}(p)$

Find a configuration for requested pose

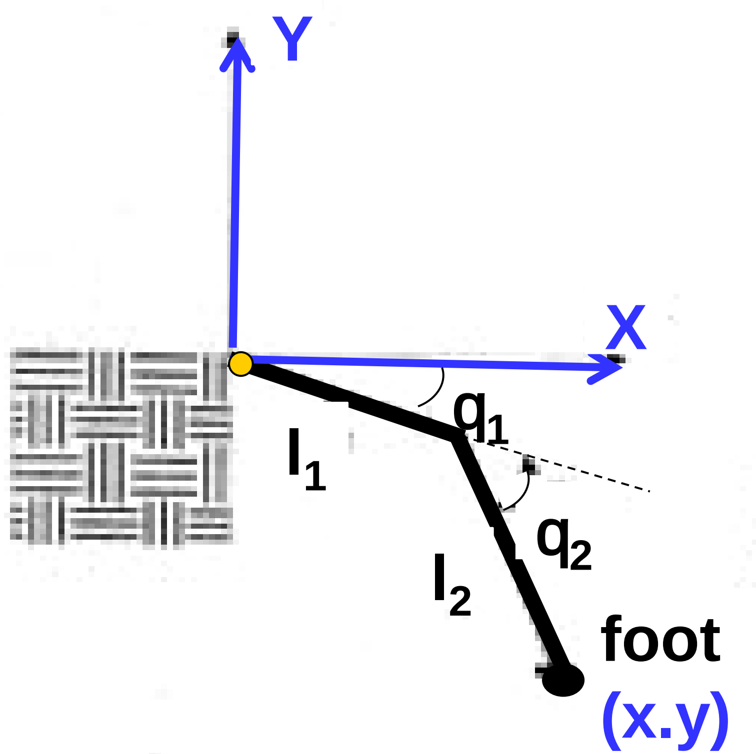


– Pose might be realized

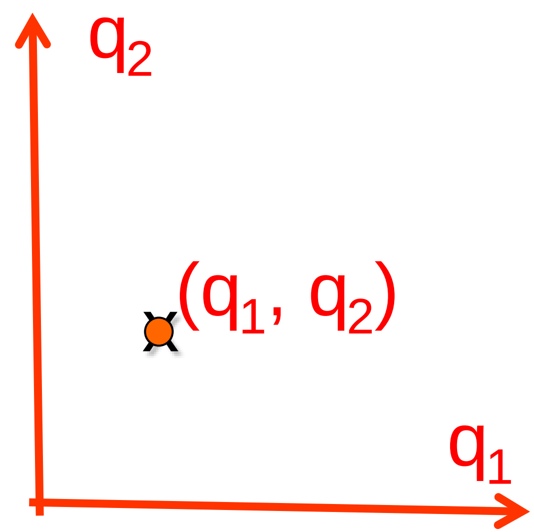
by different configurations

Example „Planar Leg“

Work space x,y



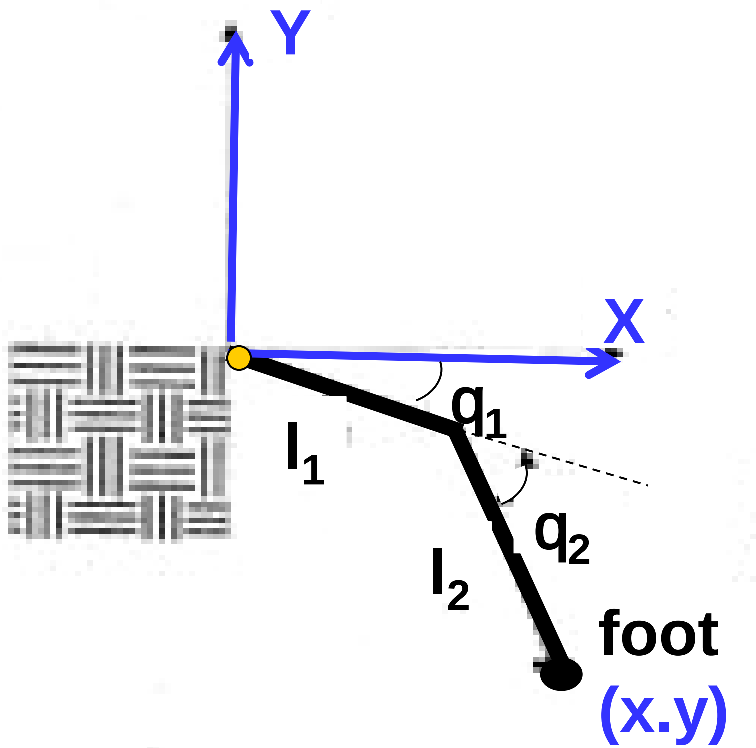
Configuration space q_1, q_2



Example „Planar Leg“

Kinematics:

- Rotation by Q_1
- Translation by l_1
- Rotation by Q_2
- Translation by l_2



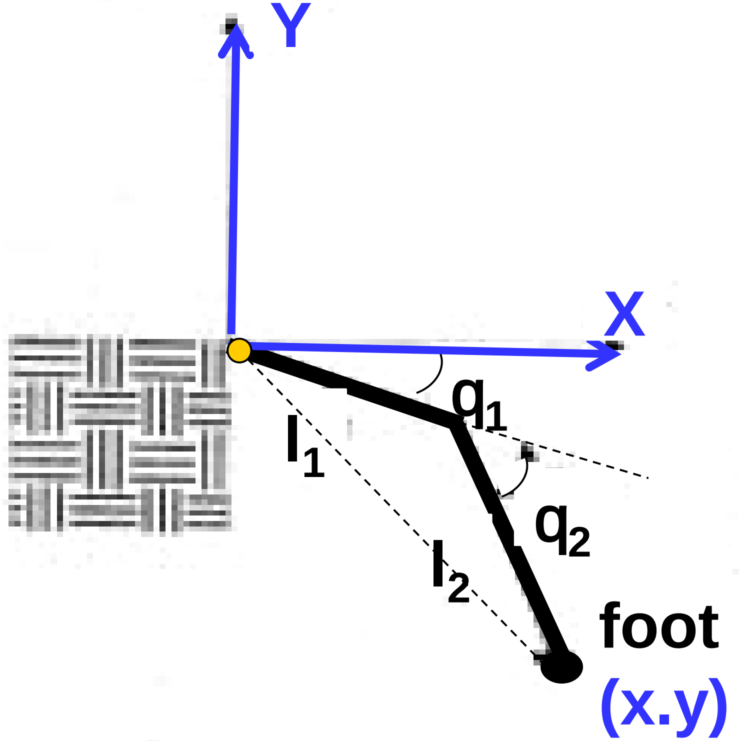
$$\begin{bmatrix} x \\ y \end{bmatrix} = l_1 \begin{bmatrix} \cos(\theta_1) \\ \sin(\theta_1) \end{bmatrix} + l_2 \begin{bmatrix} \cos(\theta_1 + \theta_2) \\ \sin(\theta_1 + \theta_2) \end{bmatrix}$$

Example „Planar Leg“

Inverse Kinematics:

(by cosine rule)

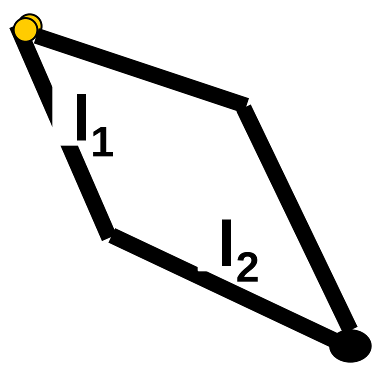
$$\cos(\theta_2) = \frac{x^2 + y^2 - l_1{}^2 - l_2{}^2}{2 l_1 l_2}$$



cos (Q_1) computable by the

formula for forward kinematics

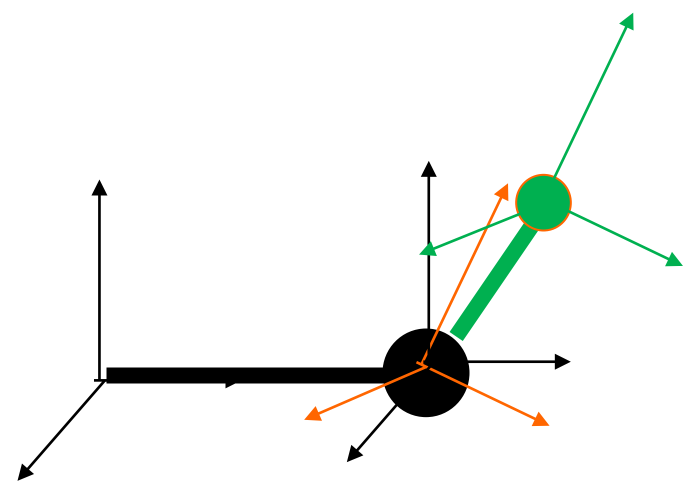
2 possible solutions:



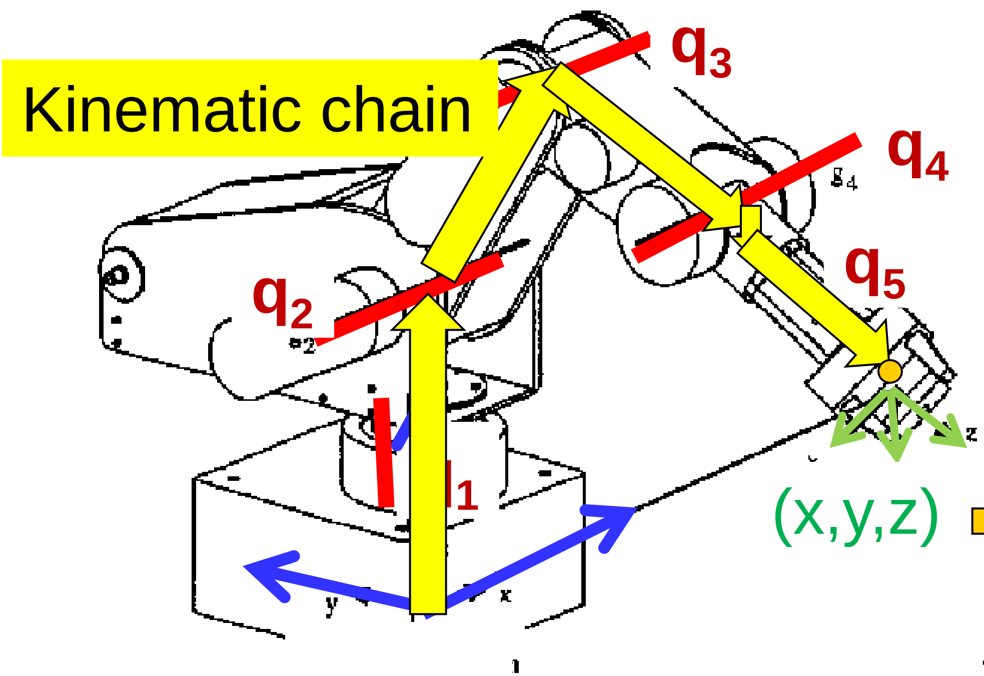
Kinematics: Calculate p = f(q)

Joints: Rotations

Limbs: Translations



Kinematics: Coordinate Transformation



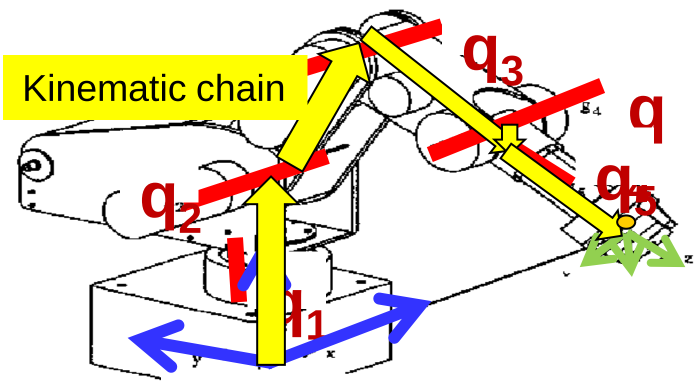
$q=(q_1,\ldots,q_5)$

Kinematics

(X,Y,Z)

From local coordinates to world coordinates

Kinematics: Coordinate Transformation



Coordinate transformation by a sequence of intrinsic rotations and translations along the **cinematic chain**.

The ordering must be preserved.

Homogenous Coordinates for 3D

4-dimensional vector

(x/w, y/w, z/w, w) with arbitrary $w \neq 0$ represents (x,y,z)

We will use (x, y ,z,1) i.e. w=1

The 4-dimensional matrix H

can describe Rotation R

followed by Translation T

$$H = \begin{bmatrix} \begin{bmatrix} & & \\ & R & \\ & & \end{bmatrix} & \begin{bmatrix} \\ T \\ \\ \end{bmatrix} \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix}$$

$$H \times \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \times \begin{bmatrix} R \times \begin{bmatrix} x \\ y \\ z \end{bmatrix} \\ 0 \end{bmatrix} + \begin{bmatrix} \begin{bmatrix} \\ T \\ \\ \end{bmatrix} \\ 1 \end{bmatrix}$$

Homogenous Coordinates for 3D

Sequence of transformations along cinematic chain can be described by matrix multiplications

$$X = M \times x$$

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = M \times \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$M = H_1 \times H_2 \times H_3 \times \ldots H_n$$

- **Kinematics**

  by computing X from $X = M \times x$ for given M, x

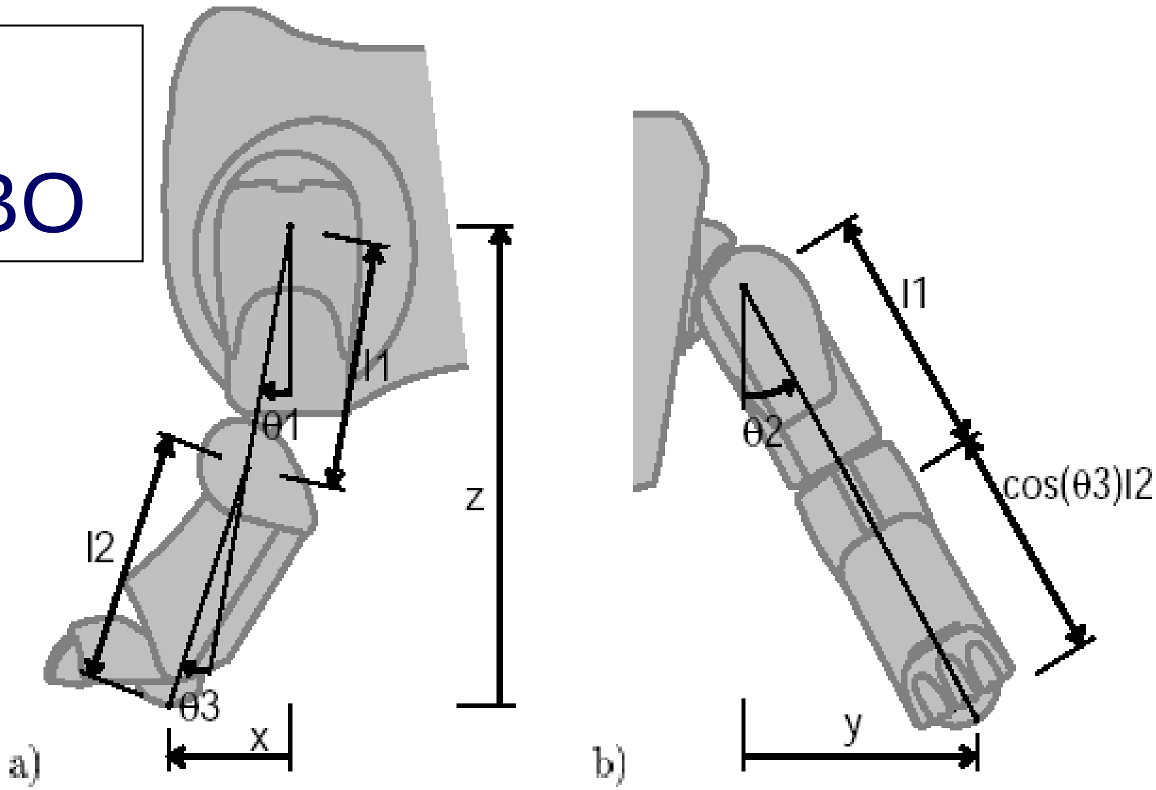
- **Inverse Kinematics**

  by finding $M = H_1 \times H_2 \times H_3 \times \ldots H_n$ for given X, x

  (but usually by other calculations resp. approximations)

Example: Kinematics AIBO

Diploma thesis
Uwe Düffert



World coordinates in the shoulder.

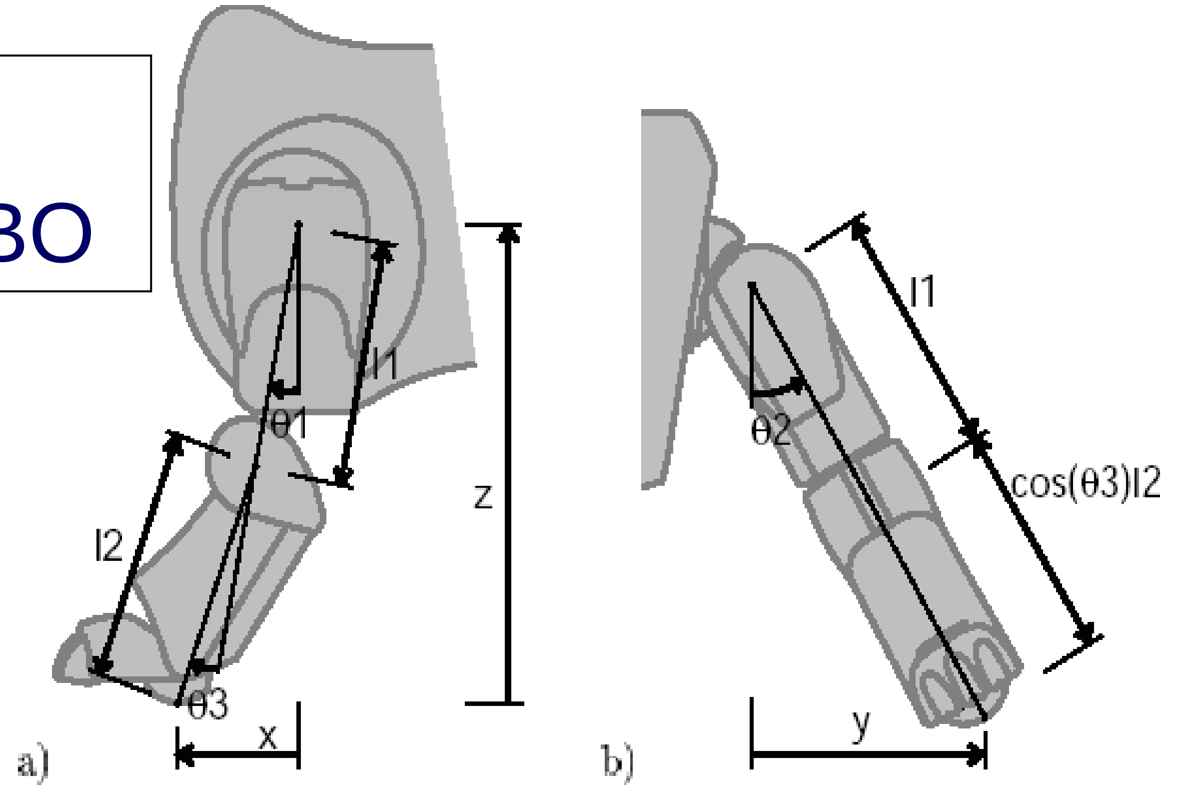
What are the coordinates (x,y,z) of the left forefoot?

Calculation:

by transformation of the foot coordinates to shoulder coordinates

Example: Kinematics AIBO

Diploma thesis
Uwe Düffert



Transformation of the foot coordinates to shoulder coordinates:

1. Translation lower leg: shift towards negative z axis (l_2).
2. Rotation knee: rotate clockwise around y-axis (q_3).
3. Translation upper leg: shift towards negative z axis (l_1).
4. Rotation shoulder 2: rotate counter-clockw. around x-axis (q_2).
5. Rotation shoulder 1: rotate clockwise around y-axis (q_1).

$$Rot(-q_1)\ Rot(q_2)\ Trans(l_1)Rot(-q_3)\ Trans(l_2)$$

Example: Kinematics AIBO

Rot($-q_1$) Rot(q_2) Trans(l_1) Rot($-q_3$) Trans(l_2)

$$\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \mathrm{Rot}_y(-\theta_1) \cdot \mathrm{Rot}_x(\theta_2) \cdot \mathrm{Trans}\begin{pmatrix} 0 \\ 0 \\ -l_1 \end{pmatrix} \cdot \mathrm{Rot}_y(-\theta_3) \cdot \mathrm{Trans}\begin{pmatrix} 0 \\ 0 \\ -l_2 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

$$= \begin{pmatrix} \cos(\theta_1) & 0 & -\sin(\theta_1) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(\theta_1) & 0 & \cos(\theta_1) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta_2) & -\sin(\theta_2) & 0 \\ 0 & \sin(\theta_2) & \cos(\theta_2) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -l_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos(\theta_3) & 0 & -\sin(\theta_3) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(\theta_3) & 0 & \cos(\theta_3) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -l_2 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

$$= \begin{pmatrix} l_2\cos(\theta_1)\sin(\theta_3) + l_2\sin(\theta_1)\cos(\theta_2)\cos(\theta_3) + l_1\sin(\theta_1)\cos(\theta_2) \\ l_1\sin(\theta_2) + l_2\sin(\theta_2)\cos(\theta_3) \\ l_2\sin(\theta_1)\sin(\theta_3) - l_2\cos(\theta_1)\cos(\theta_2)\cos(\theta_3) - l_1\cos(\theta_1)\cos(\theta_2) \\ 1 \end{pmatrix} .$$

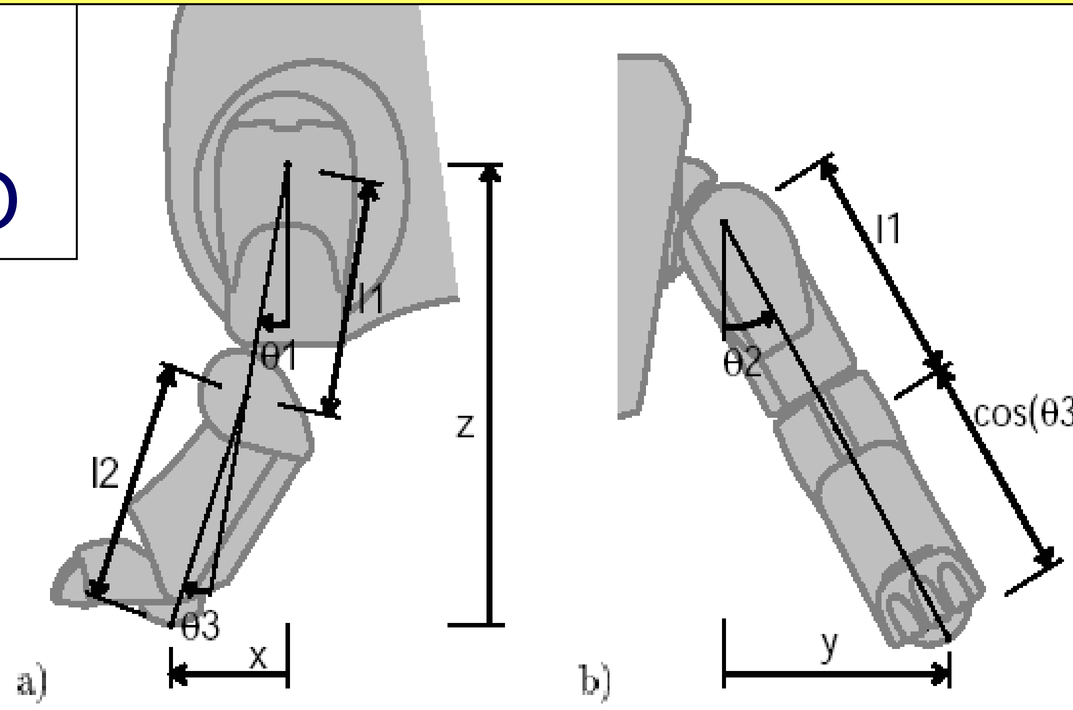
Example: Inverse Kinematics AIBO

Calculate q_1, q_2 ,q_3 for feet position (x,y,z)

q_3 (between l1,l2) by Cosine rule.

Preferably bending forward:

Positive solution.

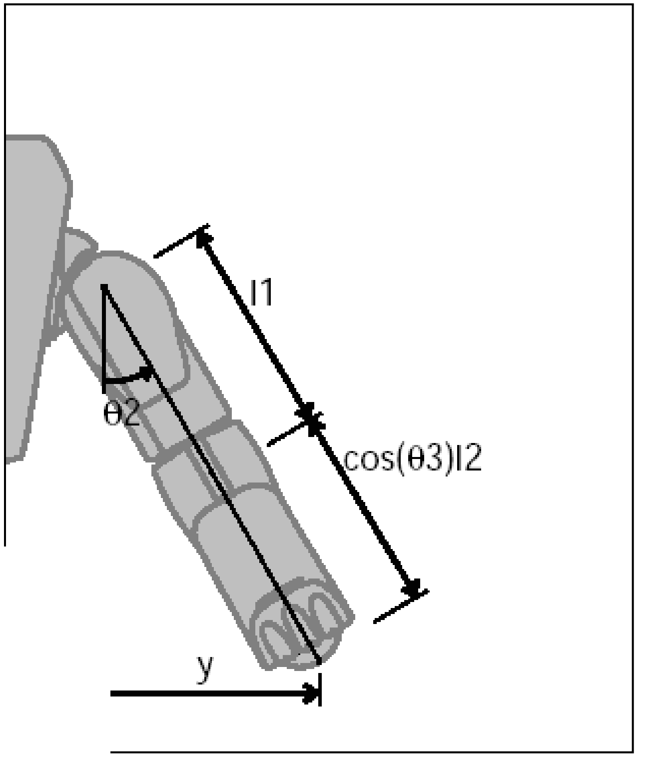


$$
\begin{aligned}
cos\left(\pi-\theta_3\right) &= \frac{l_1^2+l_2^2-(x^2+y^2+z^2)}{2\,l_1 l_2} \\
\theta_3 &= \pi \pm \arccos\left(\frac{l_1^2+l_2^2-(x^2+y^2+z^2)}{2\,l_1 l_2}\right) \\
&= \mp \arccos\left(\frac{(x^2+y^2+z^2)-l_1^2-l_2^2}{2\,l_1 l_2}\right) .
\end{aligned}
$$

Example: Inverse Kinematics AIBO

q_2 by definition of Sine,

where $| q_2 | <= p/2$ by anatomy

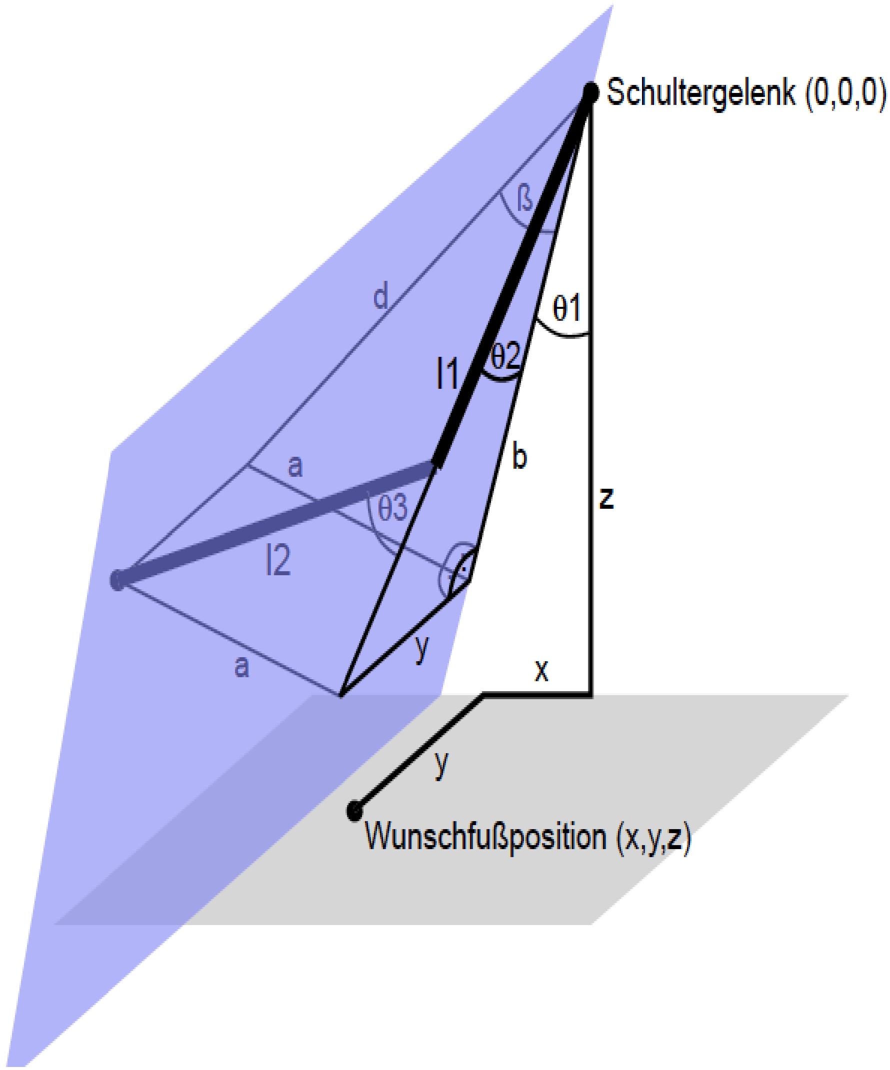


$$y = \sin(\theta_2) \cdot (l_1 + l_2 \cdot \cos(\theta_3))$$

$$\theta_2 = \arcsin\left(\frac{y}{l_1 + l_2 \cos(\theta_3)}\right)$$

Example: Inverse Kinematics AIBO

$$
\begin{aligned}
a &= l_2 \sin(\theta_3) \\
b &= (l_1 + l_2 \cos(\theta_3)) \cos(\theta_2) \\
d &= \sqrt{a^2 + b^2} \\
\beta &= \arctan(b, a)
\end{aligned}
$$



$$
\begin{aligned}
a &= d \cos(\beta) \\
b &= d \sin(\beta)
\end{aligned}
$$

Example: Inverse Kinematics AIBO

$$
\begin{aligned}
x &= l_2\cos(\theta_1)\sin(\theta_3) + l_2\sin(\theta_1)\cos(\theta_2)\cos(\theta_3) + l_1\sin(\theta_1)\cos(\theta_2) \\
&= a\cos(\theta_1) + b\sin(\theta_1) \\
&= d\cos(\theta_1)\cos(\beta) + d\sin(\theta_1)\sin(\beta) \\
&= d\cos(\theta_1 + \beta) \\
z &= d\sin(\theta_1 + \beta) \qquad .
\end{aligned}
$$

$$
\begin{aligned}
\theta_1 + \beta &= \arctan(z, x) \\
\theta_1 &= \arctan(z, x) - \beta
\end{aligned}
$$

Example: Inverse Kinematics AIBO

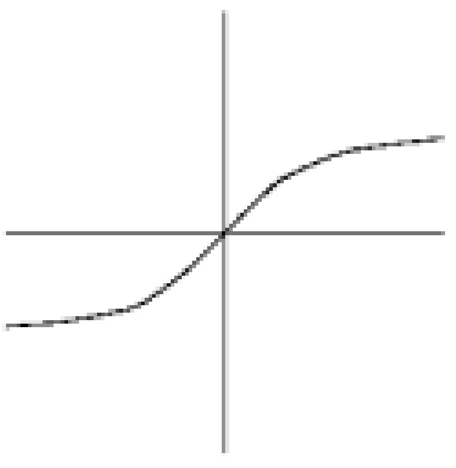
Calculate q_1, q_2 ,q_3

for feet position (x,y,z)

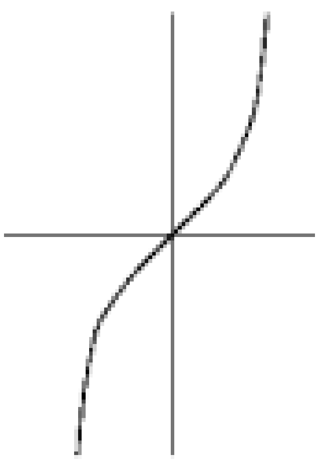
$$
\begin{aligned}
\theta_3 &= \arccos\left(\frac{x^2+y^2+z^2-l_1^2-l_2^2}{2\,l_1 l_2}\right)\\
\theta_2 &= \arcsin\left(\frac{y}{l_1+l_2\cos(\theta_3)}\right)\\
\theta_1 &= \arctan(z,x)-\arctan\Big(\big(l_1+l_2\cos(\theta_3)\big)\cos(\theta_2),\; l_2\sin(\theta_3)\Big)
\end{aligned}
$$

Special Benefits in Calculations

- Rotations in a plane (around joint axis)
- Select "simple" solutions
- Select "simple" relationships
- Use arctan (better: atan2) instead of arcsin or arccos
  (because of large error propagation near -1/+1)



arctan



arcsin

Outline

Introduction

Kinematics of Poses

Kinematics of Drive Systems

Trajectories

Motion Planning

Motion Control

Motions of Legged Robots

Optimization/Learning of Motions

Biologically Inspired Motions

Kinematics of Drive Systems

Kinematics (forward kinematics):

- Where does it move to?

Inverse kinematics (reverse kinematics):

- How can it get there?

Simplification:

Neglect mass and force

Kinematics of Drive Systems

- Driven wheels or chains
- Further wheels as stabilizers or for odometry
- Controlable wheels

Idealizing assumptions:

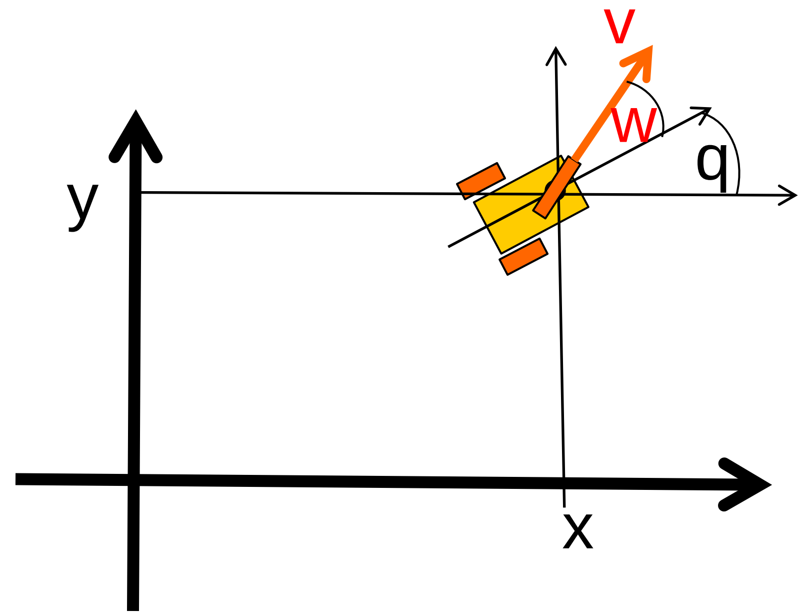
- Wheels run straight (perpendicular to the axis)
- Forward movement per complete rotation: 2pr for radius r
- Forward movement per rotation about w: wr for radius r

Drives for Vehicles on a Plane

Work space:

Pose (x,y, q) with 3 DOF

$q = 0$ in x-direction



$V(t) = (V_x(t), V_y(t))$ and $w(t)$ are *control parameters* for motion.

They depend on position and speeds of driving wheels.

Kinematics/Inverse Kinematics

Kinematics: Calculate motion from control.

Change from pose (0,0,0) to (x(t),y(t),q(t))
by speed $V(t)= (V_x(t), V_y(t))$ in direction w(t)

$$x(t) = \int_0^t V_x(t)\, dt = \int_0^t V(t) \cos [q(t)]\, dt$$

$$y(t) = \int_0^t V_y(t)\, dt = \int_0^t V(t) \sin [q(t)]\, dt$$

$$q(t) = \int_0^t w(t)\, dt$$

Inverse Kinematics:
Which control ***V*** and w is needed for desired motion?
Options depend on kind of drive.

Drives for Vehicles on a Plane

Configuration space:

Options for control:

- Speeds of the driving wheels
- Directions of the wheels / axes

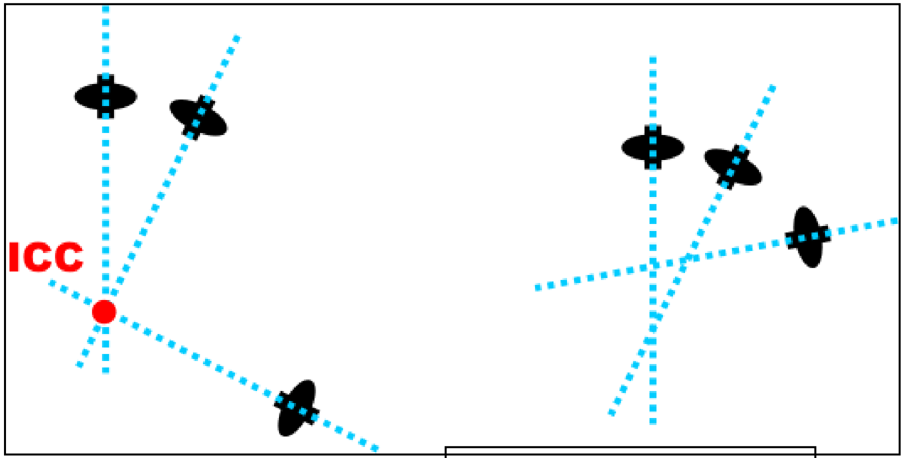
Limitations by constraints

e.g.

- connections between wheels
- Dependency between direction and speed of wheels

ICC = instantaneous center of curvature

ICC defined as intersection point of all axes



Images from Borenstein et.al.: Where am I?

Constraints for smooth motion:

- ICC exists
- Consistent speed of driving wheels

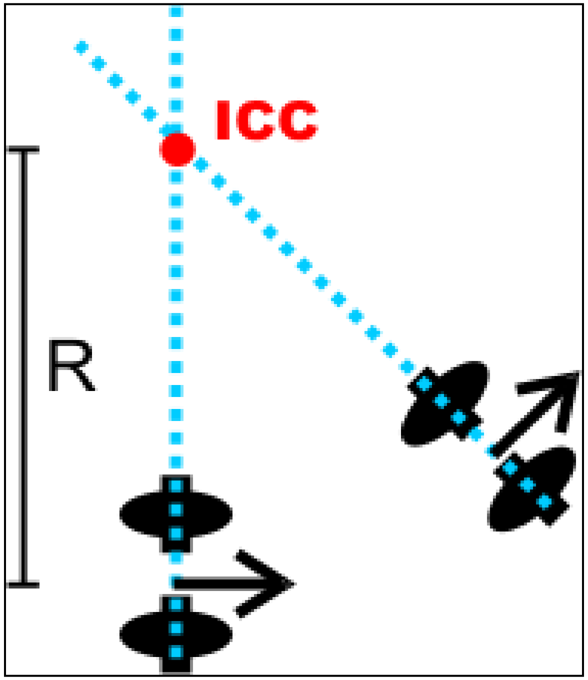
Otherwise:

- Robot loses traction
- Robot slides, unpredictable motion

ICC = instantaneous center of curvature

Robot moves on a circle around ICC.

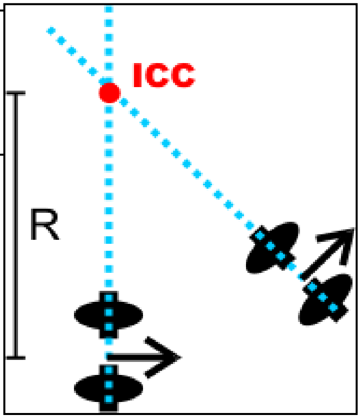
(Straight move for parallel axes: ICC infinitely far.)



ICC can be changed by

- steering of axes/wheels
- different speeds of driving wheels

Kinematics by ICC



Position of ICC for robot at pose (x,y, q) :

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

Pose of Robot after time dt

while robot rotates wdt around ICC:

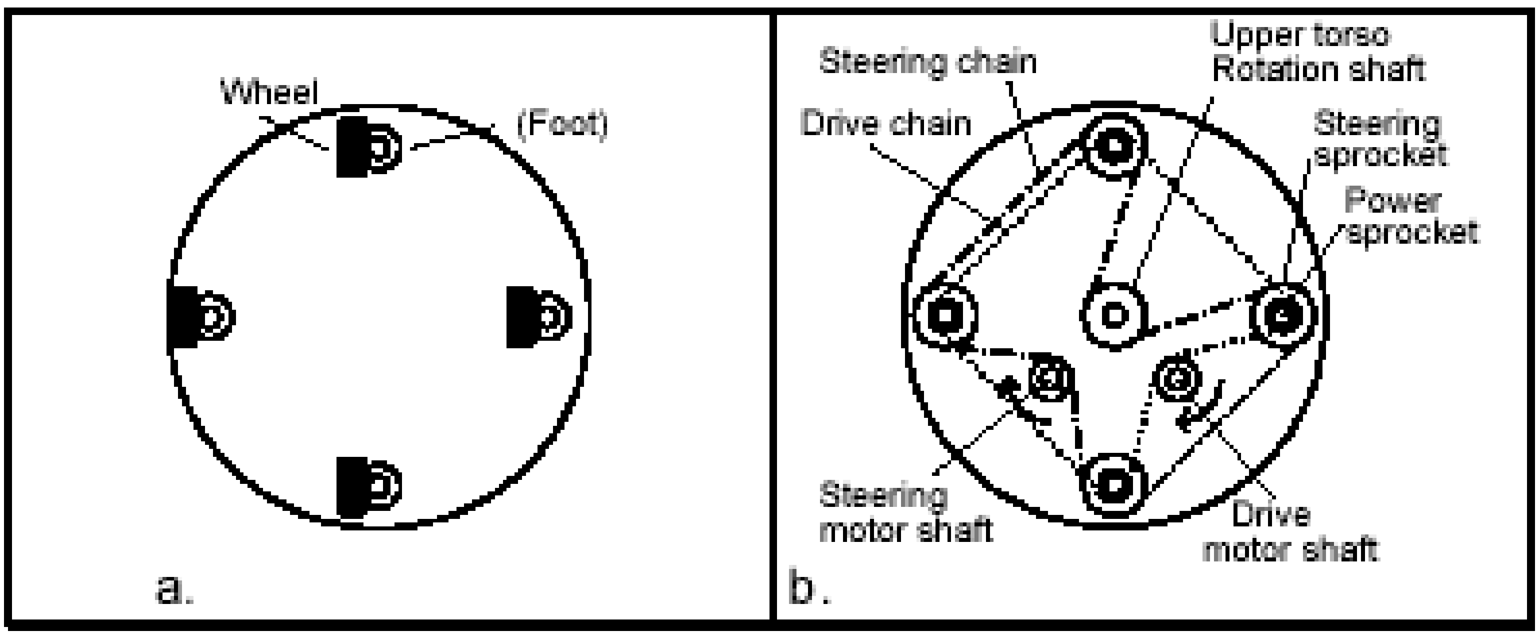
$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} cos(\omega\delta t) & -sin(\omega\delta t) & 0 \\ sin(\omega\delta t) & cos(\omega\delta t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x - ICC_x \\ y - ICC_y \\ \theta \end{bmatrix} + \begin{bmatrix} ICC_x \\ ICC_y \\ \omega\delta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

Synchrodrive

All wheels in same steerable direction w with identical speed.

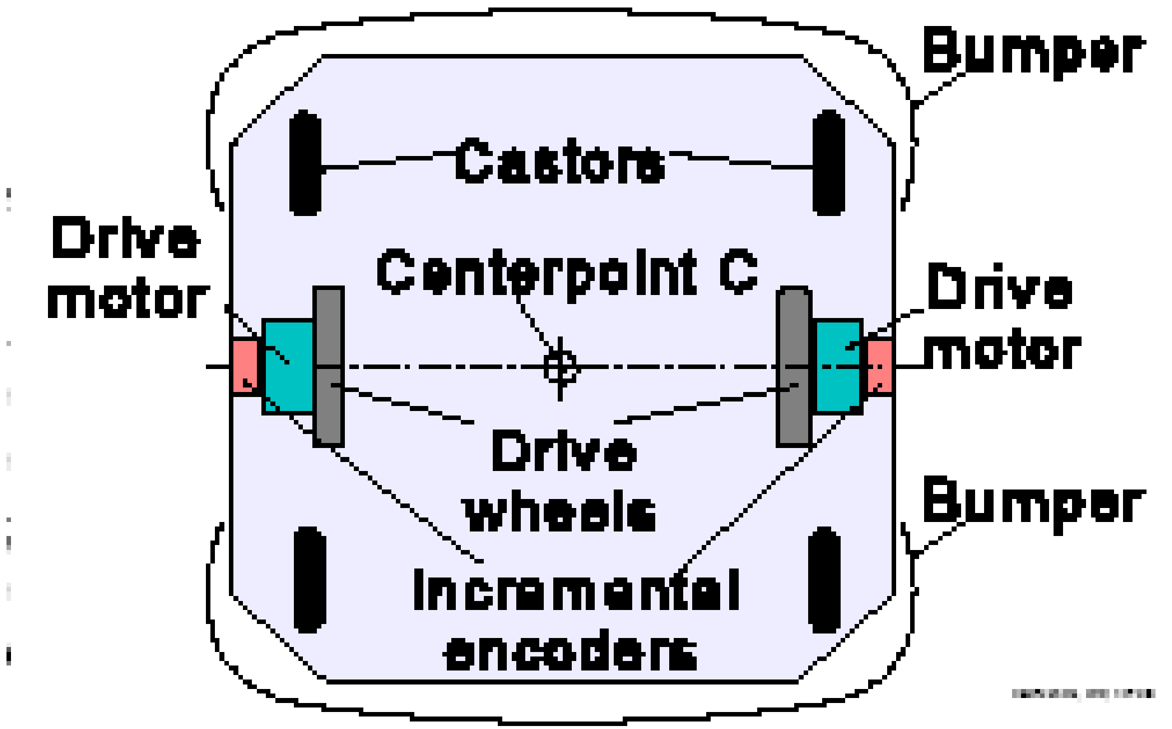
ICC infintely far perpendicular to direction w

Control:
Speed v and direction w
of wheel(s)



Differential Drive

Driving wheels on 1 axis with different speeds



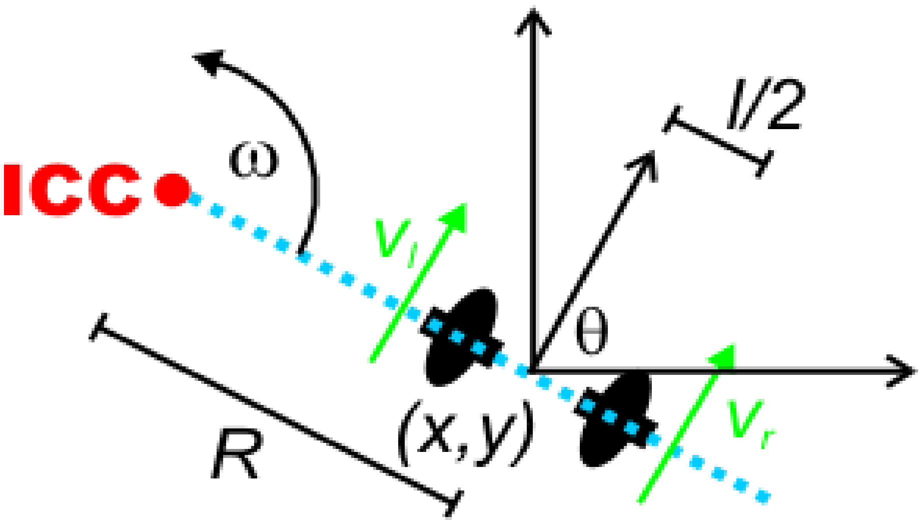
Differential Drive

ICC on the axis, position depends on v_l ,v_r

$v_l = v_r$ moves straight on

$v_l = -v_r$ turns around

Control:
Speeds v_l and v_r



$$\omega(R + l/2) = v_r$$

$$\omega(R - l/2) = v_l$$

$$R = \frac{l}{2}\frac{(v_l + v_r)}{(v_r - v_l)} \qquad \omega = \frac{v_r - v_l}{l}$$

Differential Drive: Kinematics

Change from pose (0,0,0) to (x(t),y(t),q(t))
by speeds v_l and v_r of left and right wheel

$$x(t) = \frac{1}{2}\int_0^t [v_r(t) + v_l(t)] cos[\theta(t)] dt$$

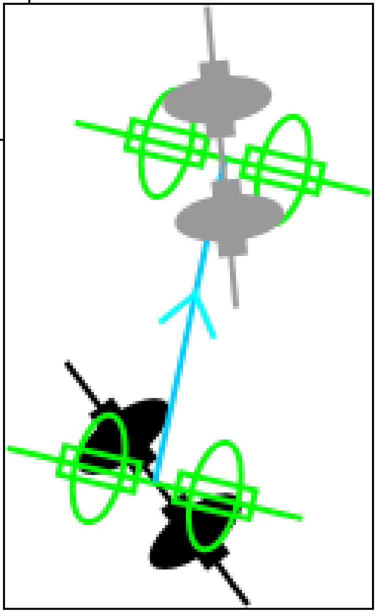
$$y(t) = \frac{1}{2}\int_0^t [v_r(t) + v_l(t)] sin[\theta(t)] dt$$

$$\Theta(t) = \frac{1}{l}\int_0^t [v_r(t) - v_l(t)]) dt$$

Differential Drive: Inverse Kinematics

Which controls $v_l(t), v_r(t)$ result indesired motion?

$$
\begin{aligned}
x(t) &= \frac{1}{2}\int_0^t [v_r(t) + v_l(t)] cos[\theta(t)] dt \\
y(t) &= \frac{1}{2}\int_0^t [v_r(t) + v_l(t)] sin[\theta(t)] dt \\
\Theta(t) &= \frac{1}{l}\int_0^t [v_r(t) - v_l(t)]) dt
\end{aligned}
$$



Many different solutions to arrive at a given target.

No motion in direction of the axis (towards ICC).

Differential Drive: Inverse Kinematics

$$x(t) = \frac{1}{2}\int_0^t [v_r(t) + v_l(t)] cos[\theta(t)]dt$$

$$y(t) = \frac{1}{2}\int_0^t [v_r(t) + v_l(t)] sin[\theta(t)]dt$$

$$\Theta(t) = \frac{1}{l}\int_0^t [v_r(t) - v_l(t)])dt$$

Special cases:

$v = v_l = -v_r$:

$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x \\ y \\ \theta + 2v\delta t/l \end{pmatrix}$$

Turn on place

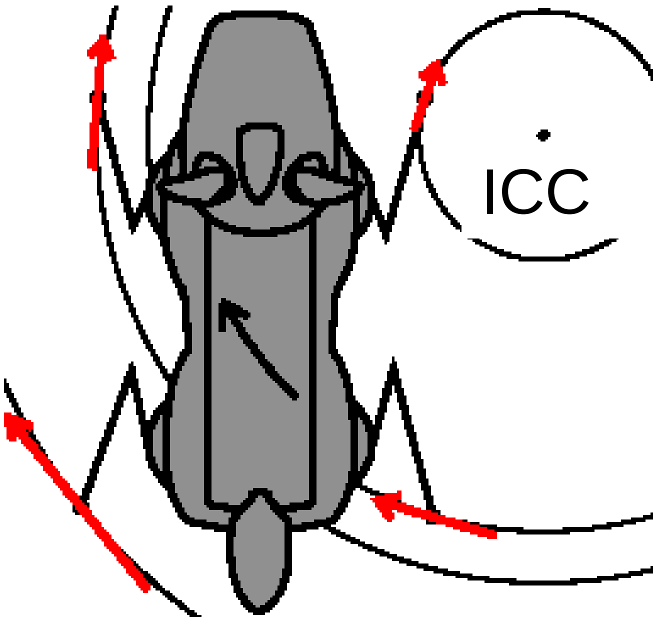
$v = v_l = v_r$:

$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x + v\cos(\theta)\delta t \\ y + v\sin(\theta)\delta t \\ \theta \end{pmatrix}$$

Forward motion

AIBO: „Wheel model“ (Differential Drive)

Curved motion by different speeds of legs.



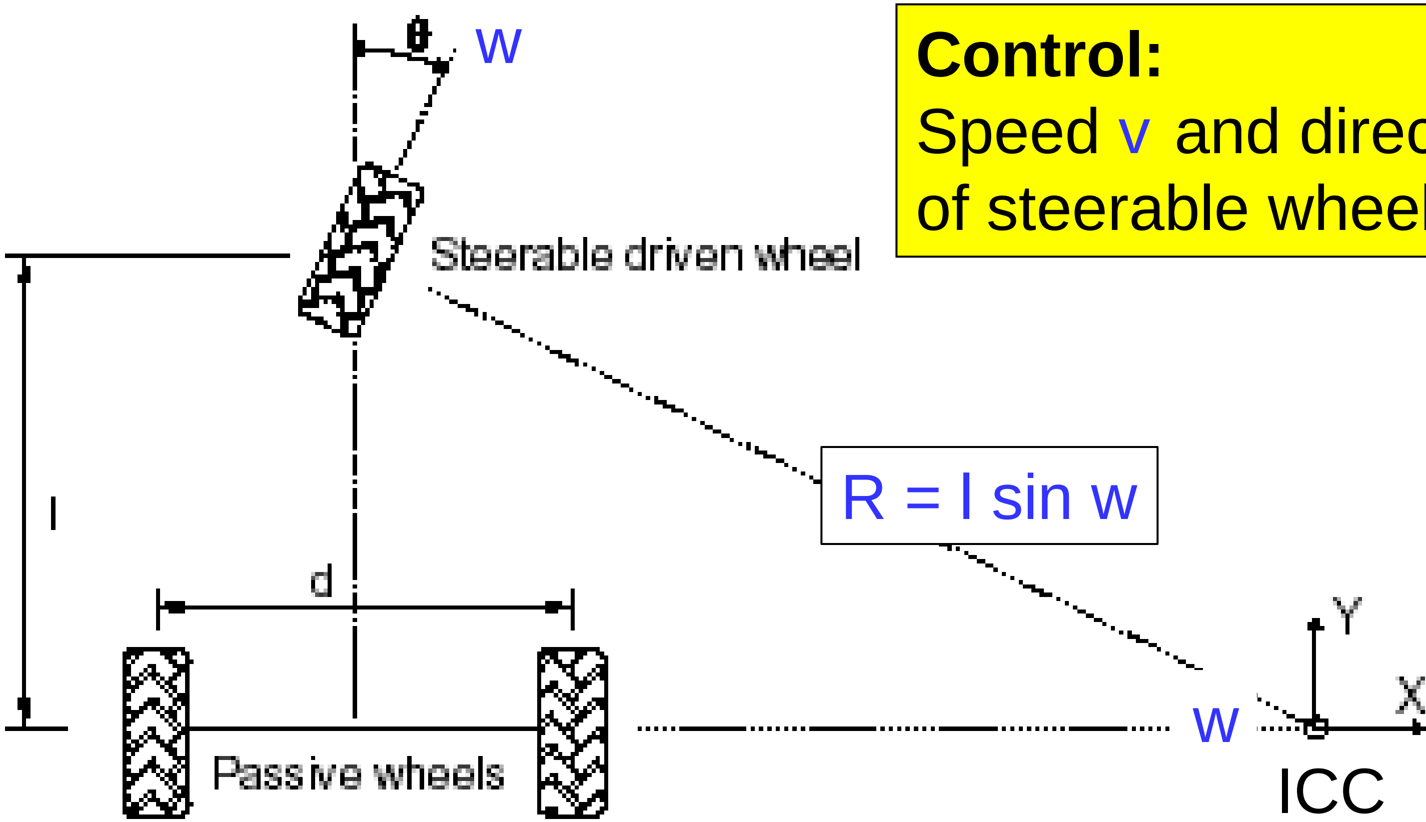
Controlled Wheels

One (or more connected) steerable wheels, other wheels passiv:

Bicycle, Tricycle, Wagon etc.

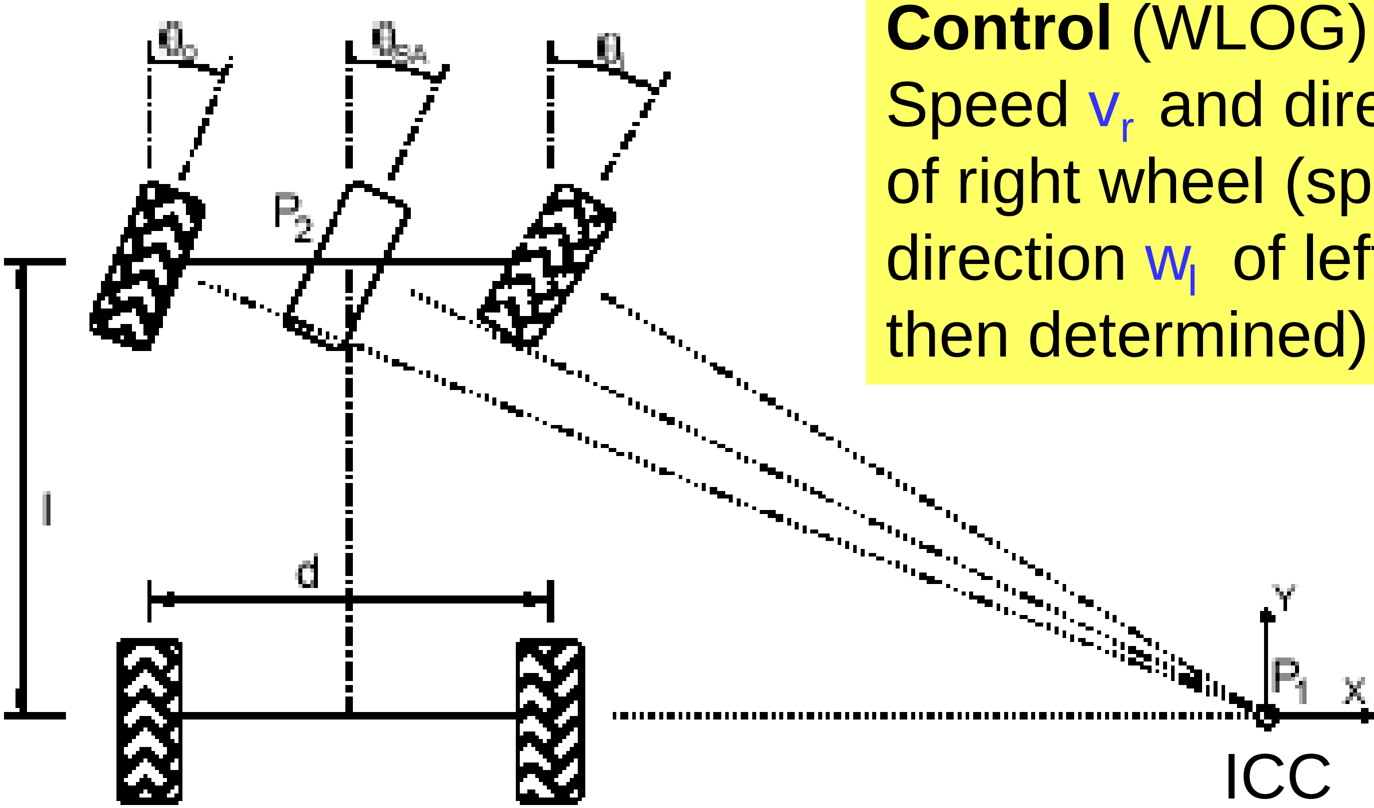
ICC on the axis of passive wheel(s), position depends on w

Control:
Speed v and direction w
of steerable wheel(s)



Ackermann-Drive: Automobile

Front wheels are individually steerable



Control (WLOG):
Speed v_r and direction w_r of right wheel (speed v_l and direction w_l of left wheel is then determined)

Modell with ICC like for tricycle by a phantom wheel at P_2

Characteristica of Drives:

Mostly 2 control parameters
for 3 spatial DOF
ØNonholonomic drives

Rotation on place:

- Differential drive
- Tricycle, Ackerman only for $w = 90^0$ (with stability problems)

Differential drive:

- Uneven terrain and sliding results in direction errors for.

Tricycle, Ackerman:

- Complicated maneuvers (parking!)

Ackerman:

- Improved stability by separated (and slanted) front wheels

Degrees of Freedom (DOF) - continued

DOF is in both work space resp. configuration space the

- minimal number of parameters for complete description

equivalently:

- maximal number of independent parameters

Work space: *effective DOF*

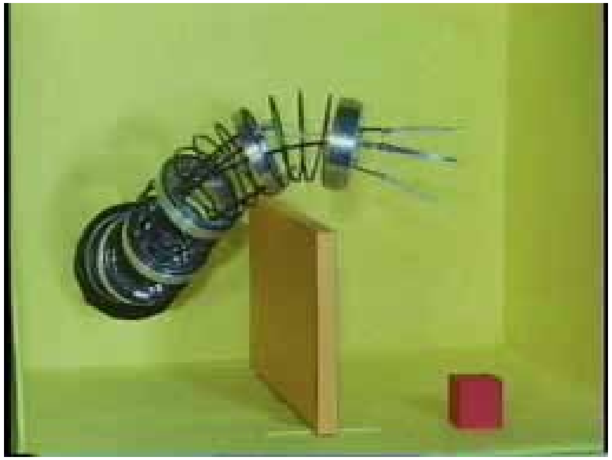
Configuration space: *controlable DOF*

Degrees of Freedom (DOF) - continued

Number of effective DOF

i.g. different from number of controlable DOF.

- All poses in work space may be reachable even in case of effective DOF > controlable DOF (e.g. differential drive)

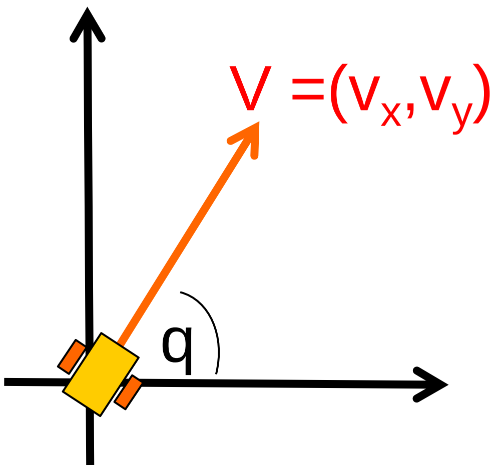


- effective DOF < controlable DOF is useful in case of obstacles

Nonholonomic Drive Systems

Nonholonomic Constraints $C(p_1,\ldots p_n, \dot{p}_1,\ldots \dot{p}_n,t) = 0$
impose dependencies of paramaters *and their derivatives*.

Holonomic Constraints
$C(p_1,\ldots,p_n,t) = 0$
impose dependencies
of parameters.

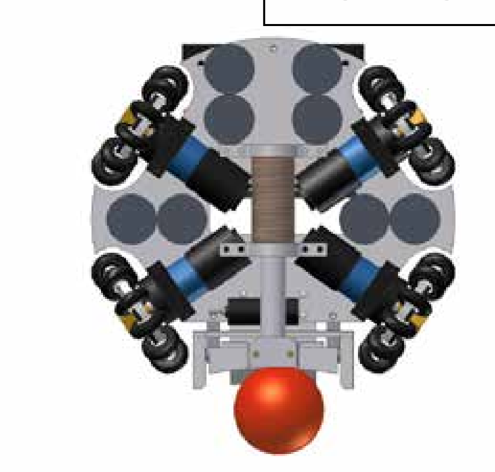


Nonholonomic Constraint:
$tan q = v_y / v_x$ i.e. $v_x sin q - v_y cos q = 0$
$cos q = 0$ for $q = p/2$ implies $v_x = 0$:
No motion in direction of axis (e.g. for differential drive)

Holonomic Drive Systems

Most drive systems are nonholonomic
and have only 2 controllable parameters

Fufighters
FU Berlin



Holonomic drives:

- Omnidirectional drive
  (Control by separate motors of wheels)

- Synchrodrive for rotationally symmetric vehicles
  (2 spatial DOF)
- Synchrodrive with additional body rotation

Outline

Introduction

Kinematics of Poses

Kinematics of Drive Systems

Trajectories

Motion Planning

Motion Control

Motions of Legged Robots

Optimization/Learning of Motions

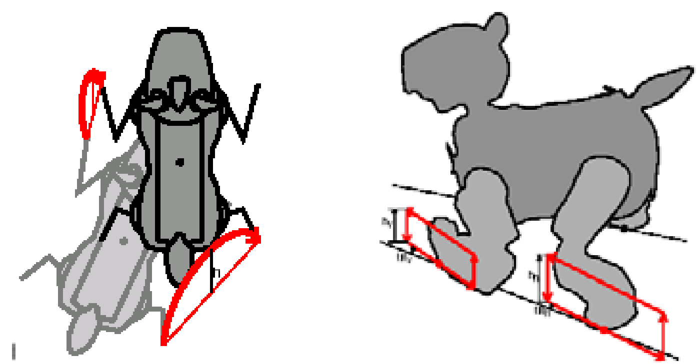
Biologically Inspired Motions

Trajectories

Trajectory in work space/configuration space:

Sequence of spatial parameters (positions/poses of the robot or its parts) or of control parameters at different times, e.g.

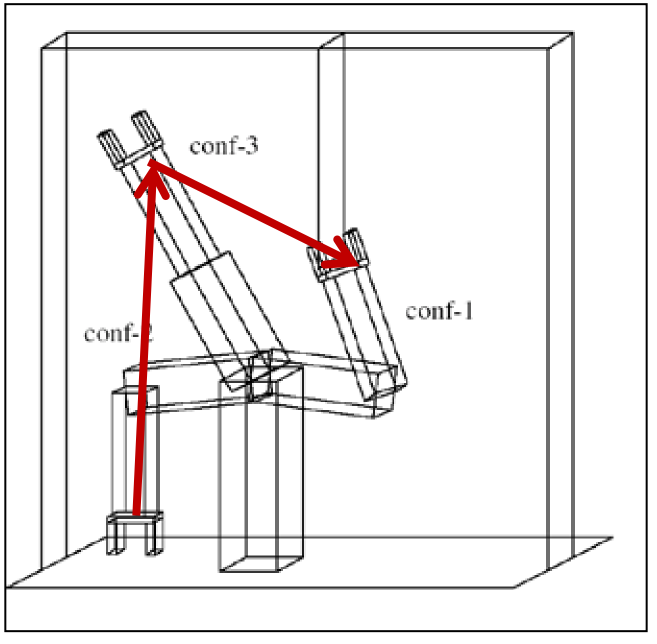
- trajectory of CoM (center of mass)
- trajectory of feets
- trajectory of limb angles

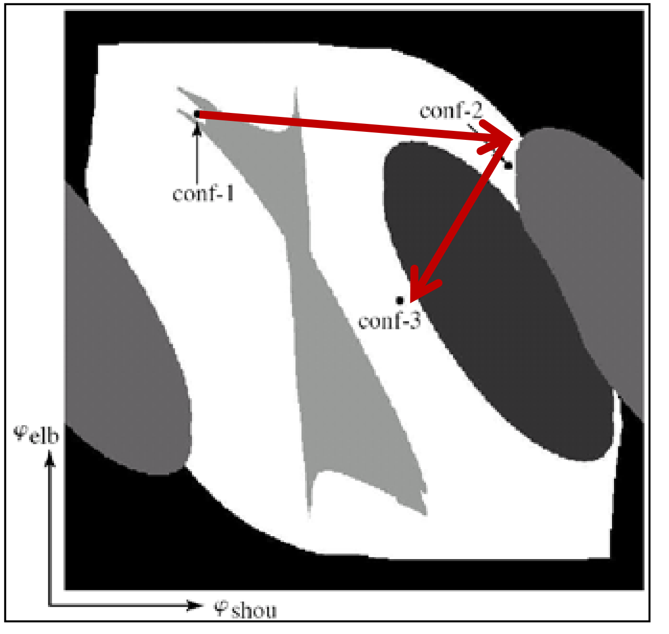


Diploma Thesis U. Düffert

Trajectories

Set of poses p(t) and corresponding configurations q(t):





Motion planning: Find realistic (and optimal) trajectories.
The trajectories in the figures are not realistic.

Trajectories of Keyframes

Sequence of Keyframes:

Characteristic poses during a motion (“like in a comic”).

Originally used in animated movies.

Transition times define speed to reach next pose.

Poses between keyframes must be interpolated.

Keyframe

Complete set of joint angles to be set in given time

Time 1000

HeadPitch HeadYaw 0

RShoulderPitch LShoulderPitch 120

RShoulder RollLShoulderRoll 0

RElbowRoll 90

LElbowRoll -90

RElbowYaw 90

LElbowYaw -90

RHipYawPitch LHipYawPitch 0

RHipPitch LHipPitch -31

RHipRoll LHipRoll 0

RKneePitch LKneePitch 63

RAnklePitch LAnklePitch -31

….

Motion Skill: Sequence of Keyframes

FILE walk_forward-flemming-nika.txt
in …/keyframes

300 0 -21 -62 32 -69 -59 0 -8

300 -5 -21 -62 46 -69 -59 0 0

300 0 -21 -62 60 -69 -59 0 8 -10 -0 12 -11 0 8 12 -0 -3 -11 -110 -32 69 59

300 0 -21 -75 60 -69 -59 0 8 6 -36 27 -11 0 8 12 -15 7 -11 -97 -32 69 59

300 0 -21 -86 60 -69 -59 0 8 42 -69 13 -11 0 8 12 -30 23 -11 -86 -32 69 59

300 0 -21 -110 60 -69 -59 0 8 12 -0 -9 -11 0 8 -10 -0 12 -14 -62 -32 69 59

300 -5 -21 -110 46 -69 -59 0 0 18 -0 -9 -4 0 0 -10 -0 17 -5 -62 -46 69 59

300 0 -21 -110 32 -69 -59 0 -8 12 -0 -3 11 0 -8 -10 -0 12 11 -62 -60 69 59

300 0 -21 -97 32 -69 -59 0 -8 12 -15 7 11 0 -8 6 -36 27 11 -75 -60 69 59

300 0 -21 -84 32 -69 -59 0 -8 12 -30 23 11 0 -8 42 -69 13 11 -84 -60 69 59

Each line starts with the transition time followed by the target angles of joints in a predefined order.
RoboNewbie:
Keyframe sequences are “played” by class keyframeMotion.

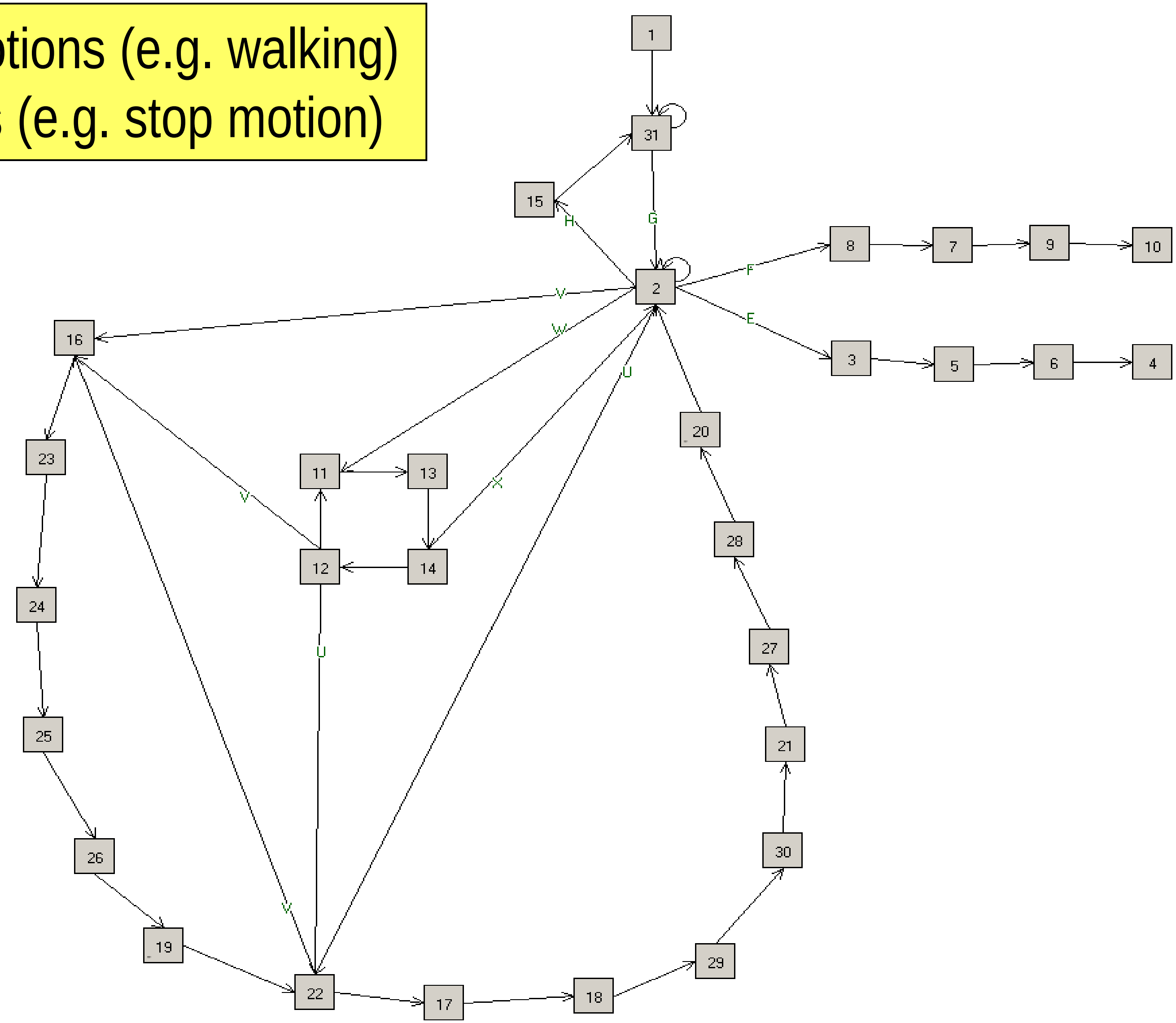
Order of Joints in RoboNewbie Keyframes

NeckYaw = 0
NeckPitch = 1
LeftShoulderPitch =2
LeftShoulderYaw = 3
LeftArmRoll = 4
LeftArmYaw = 5
LeftHipYawPitch = 6
LeftHipRoll = 7
LeftHipPitch = 8
LeftKneePitch = 9
LeftFootPitch = 10

LeftFootRoll = 11
RightHipYawPitch = 12
RightHipRoll = 13
RightHipPitch = 14
RightKneePitch = 15
RightFootPitch = 16
RightFootRoll = 17
RightShoulderPitch = 18
RightShoulderYaw = 19
RightArmRoll = 20
RightArmYaw = 21

Keyframes: MotionNets

Cycles: Repeated motions (e.g. walking)
Conditional Branches (e.g. stop motion)



Motion Editor could be extended for Motion Nets

Burkhard

Keyframes

Simple implementation

Simple design (especially with “teaching”)

But motions can not adapt

Best suited for short sequences (stand-up, kick)

Usage of Trajectories for Motion Planning

Find a trajectory (path) of the robot or a part of the robot
in work space or configuration space
which satisfies certain conditions, e.g.

- Motion from start to destination while avoiding obstacles
- Motion of a limb while maintaining stability
- Motion of a manipulator to grasp an object

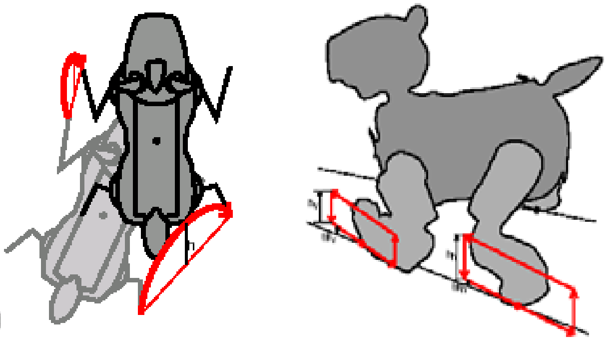
Side conditions may be
time, energy, smoothness, stability, safety…

Appropriate trajectories can be found e.g.
by physical models or by Machine Learning

Usage of Trajectories for Motion Control

Control the actuators (joint, limbs,…) such that
the robot or a part of the robot follows a given trajectory.

Inverse kinematics
can be used to find the appropriate control parameters.



Shift CoM following a (straight) trajectory
implies trajectories of feet, e.g. semi-ellipses or parallelograms.
Related joint controls by inverse kinematics.

Outline

Introduction

Kinematics of Poses

Kinematics of Drive Systems

Trajectories

Motion Planning

Motion Control

Motions of Legged Robots

Optimization/Learning of Motions

Biologically Inspired Motions

Planning

… is a broad field in AI with many different methods.

Planning can be used for motions and for more complex behaviors (different time horizons).

Here: Some useful methods for motion planning

Later: Behavior planning

Motion Planning vs. Control

Robot can

- plan motions (and more complex behavior) before execution
- execution is then performed by appropriate control

Robot Control can be performed as

- Open loop control: (“blind” control)
  Preplanned motions performed without sensor feedback.
- Closed loop control:
  Sensor feedback is used for adaptation of intended motions.

Some planning methods lead directly to controls
(e.g. potential fields).

Motion Planning vs. Control

Alternative for Planning:

Online motion control
by immediate reactions to sensor measurements
(e.g. for maintaining balance)

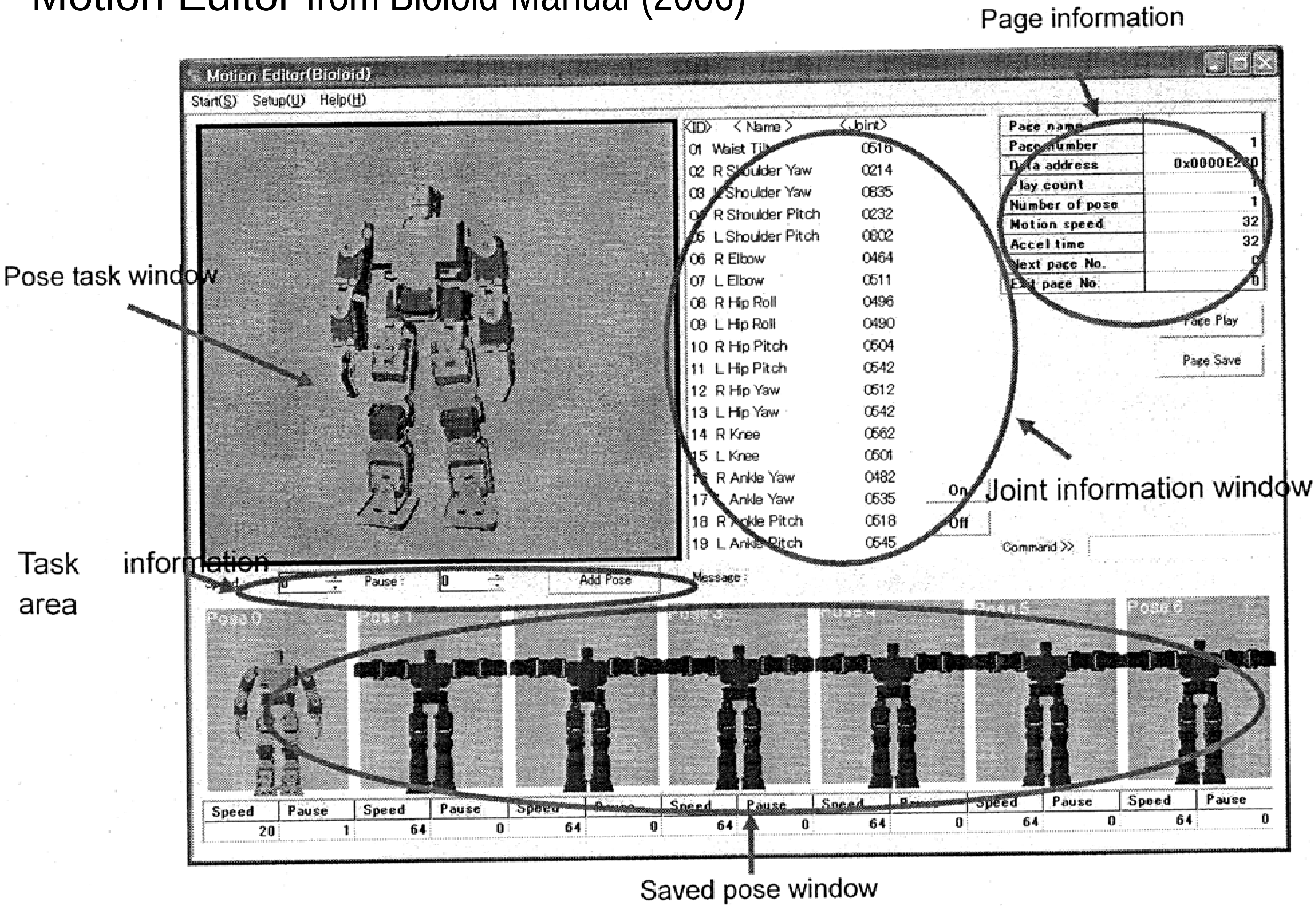
- sensor actor coupling
- behavioral robotics
- emergence principle

Ø later more

Teaching

- Set „characteristic poses“ of a motion by hand (at real robot) or by motion editor
- Protocol joint angles of each such pose as *keyframe* resulting in a sequence of keyframes
- Optimize (transition times, smoothing, …) e.g. by machine learning

Motion Editor from Bioloid Manual (2006)



Motion-Capturing

Imitate demonstrated motions

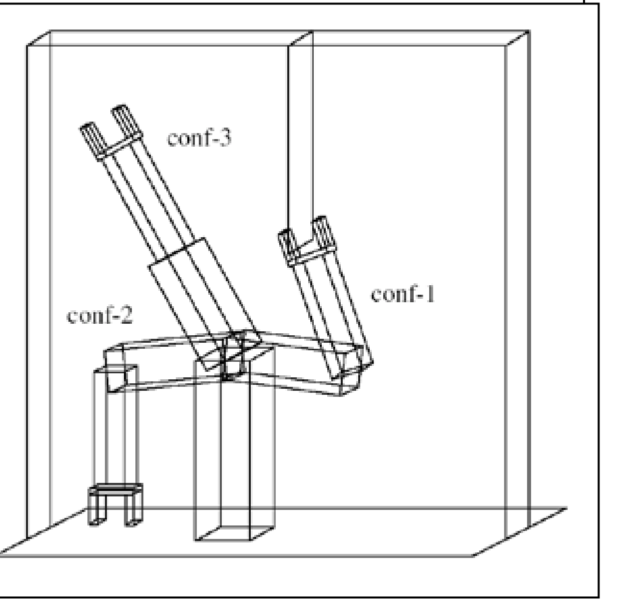
From IROS 2007

Markers at important points

Record motions (3D Motion Tracker)

Implement related control (e.g. by analyzing the motion)

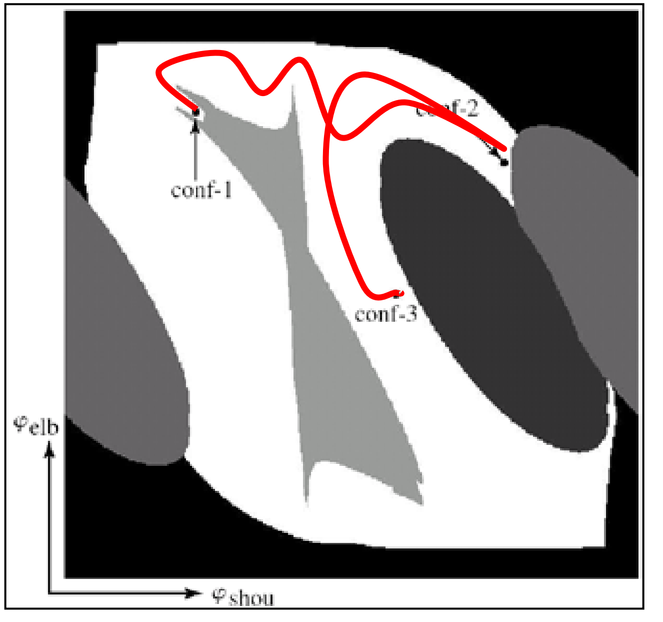
Motion Planning



Optimality of a trajectory may concern

- Length of path
- Time
- Smoothness
- Stability
- Safety
- Energy consumption
- Esthetics
- …

Planning can be performed
in work space or configuration space
using path planning algorithms (e.g. A*)



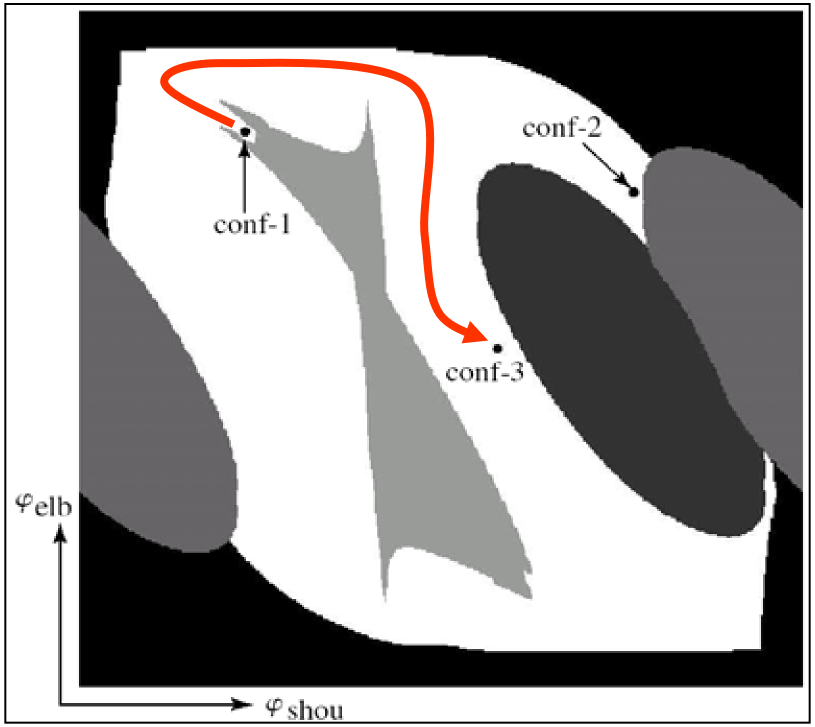
Images
Russell/Norvig: Artificial Intelligence

Planning in Configuration Space

Special regions in the configuration space for

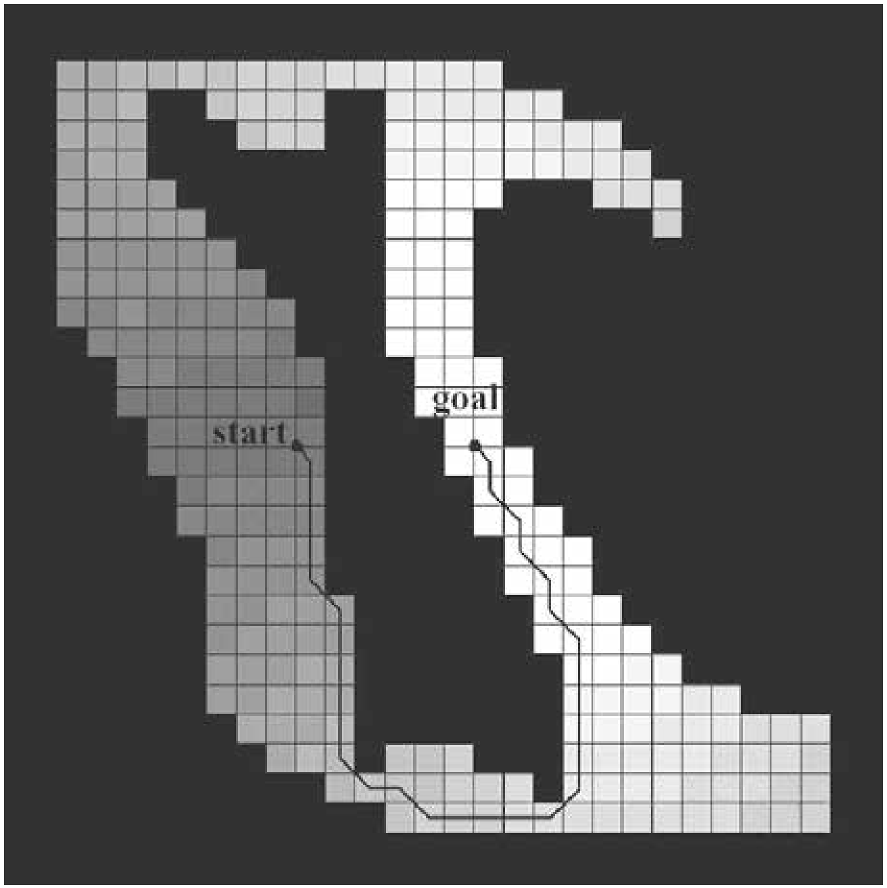
- obstacles in work space (gray),
- geometry of the robot (black)

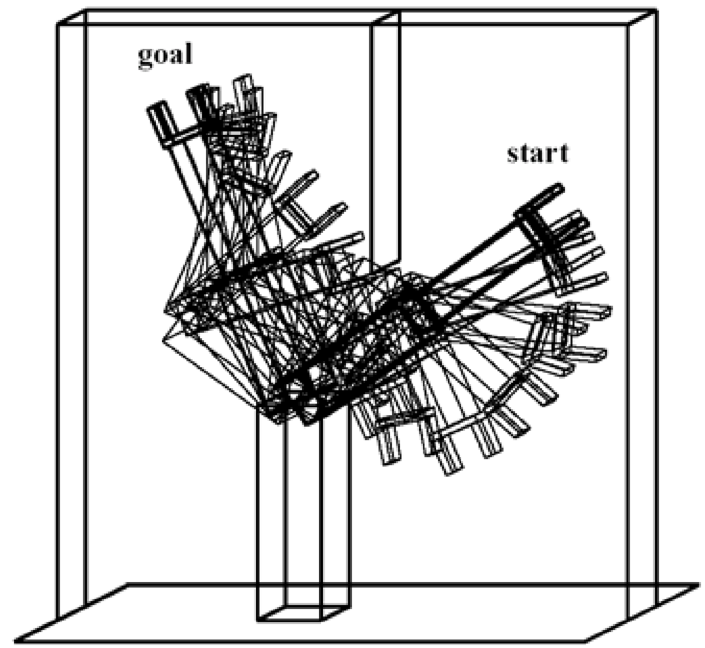
Results of planning in configuration space can be directly used as control for motion in work space.



Grid Based Search in Configuration Space

e.g. using graph search methods like A*



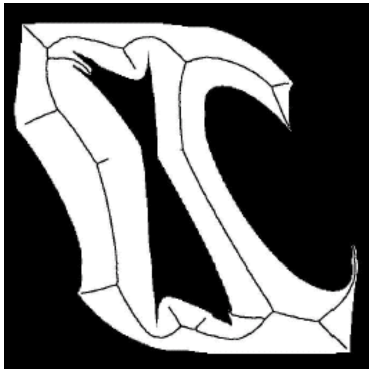


Skeleton Based Search in Configuration Space

Skeleton: Connects certain points.

Search for path on skeleton.

- as Voronoi-Graph:
  points with equal minimal
  distances to obstacles



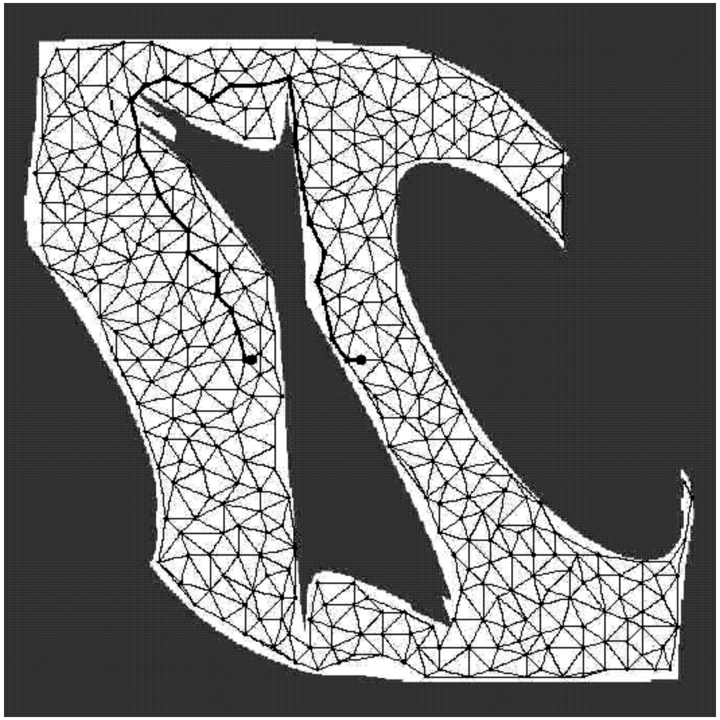
- as Visibility Graph:
  Nodes at corners of obstacles
  Arcs between mutually observable nodes

Problems:
• complex algorithms
• results often in detours

Random Point Search in Configuration Space

Graph search through random points in free space.

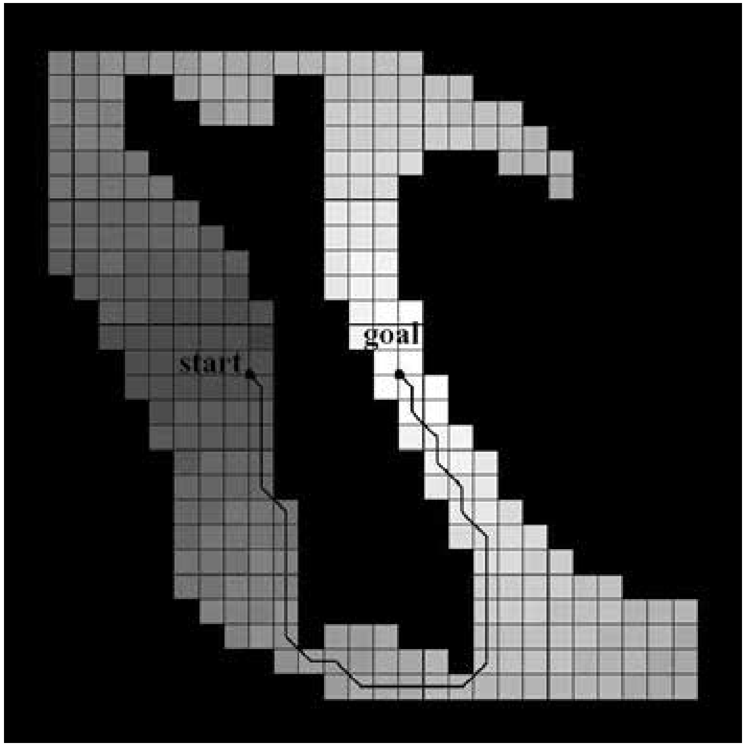
Ranking of preferable areas by differently distributed points.



Potential Field in Configuration Space

„Potentials at the field“:

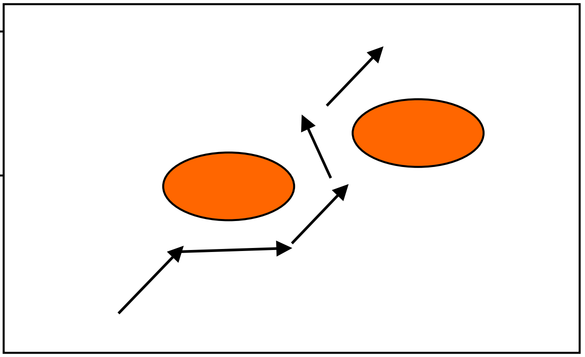
- Target attracts
- Obstacles repel



Can be combined with other search methods.

Can be used as control:
Robot follows attractions in the potential field.

Potential field



Control in a point (x_0,y_0)

as direction vector $[\mathbf{F_x}(x_0,y_0), \mathbf{F_y}(x_0,y_0)]$

of vector field $\mathbf{F}(x, y) = [\mathbf{F_x}(x,y), \mathbf{F_y}(x,y)]$

Special case:

Vector field $\mathbf{F}(x, y)$ is gradient of a potential field $\mathbf{U}(x, y)$

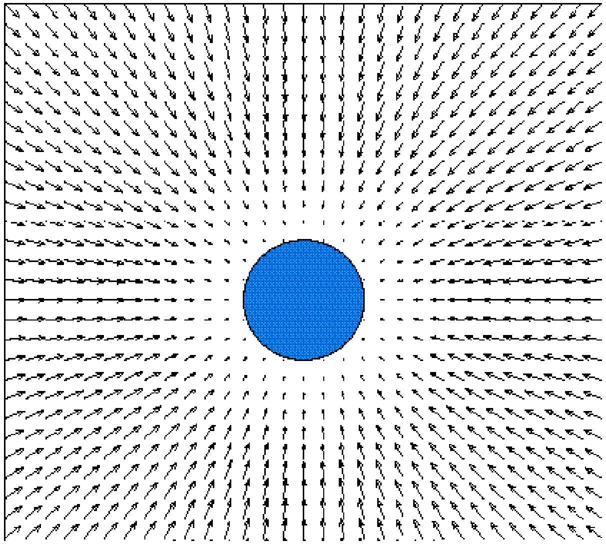
$$\mathbf{F}(x, y) = [\, d\mathbf{U}(x, y) / dx \,,\, d\mathbf{U}(x, y) / dy \,]$$

For application:

- Potential determined by environment/from sensory information
- Motion follows the gradient

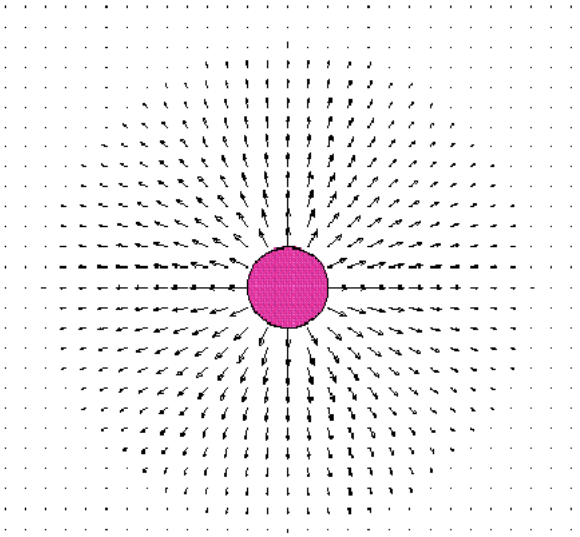
Potential field

target:
attracting field



e.g.
$\mathbf{U}_{goal}(p) = a\ dist(p,goal)^2$

obstacles:
repelling fields



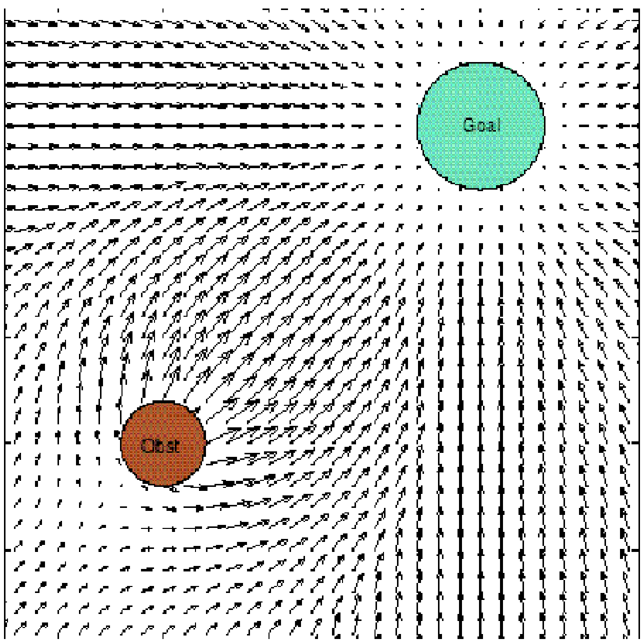
e.g.
$\mathbf{U}_{obstacle}(p) = b\ dist(p,obstacle)^{-1}$

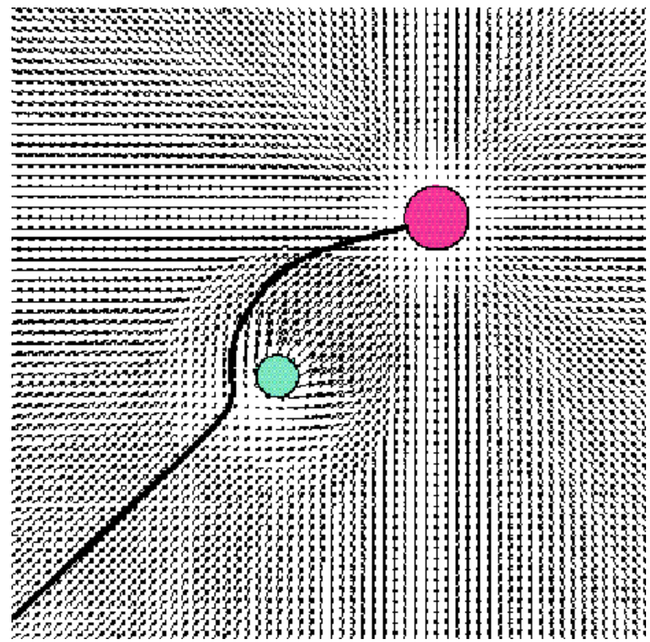
Potential field

Potential field by superposition (addition):

$$\mathbf{U}(p) = \mathbf{U}_{goal}(p) + S\,\mathbf{U}_{obstacle}(p)$$

$$\mathbf{F} = - [\, d\mathbf{U} / dx \,,\, d\mathbf{U} / dy \,]$$





Potential field

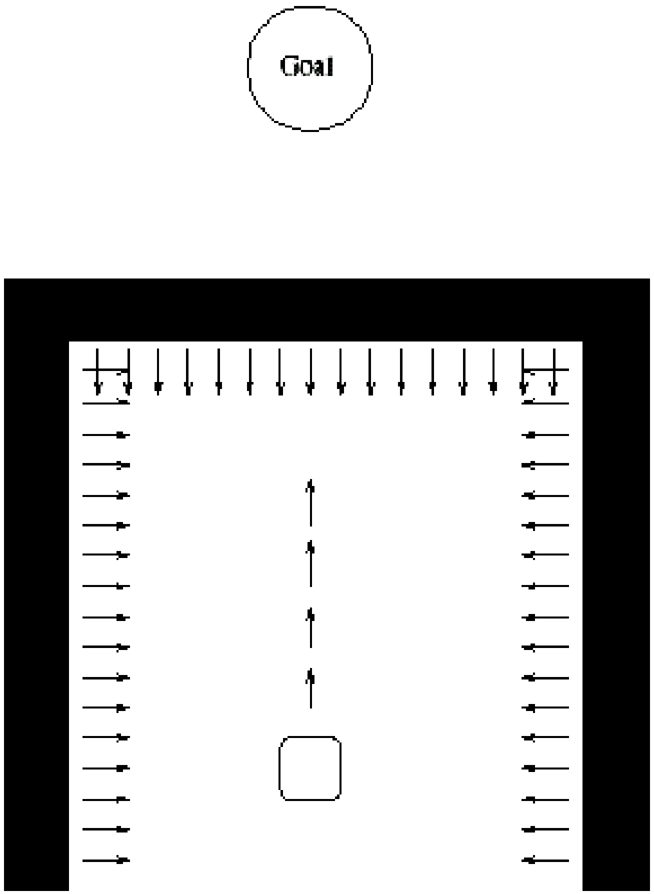
Benefits:

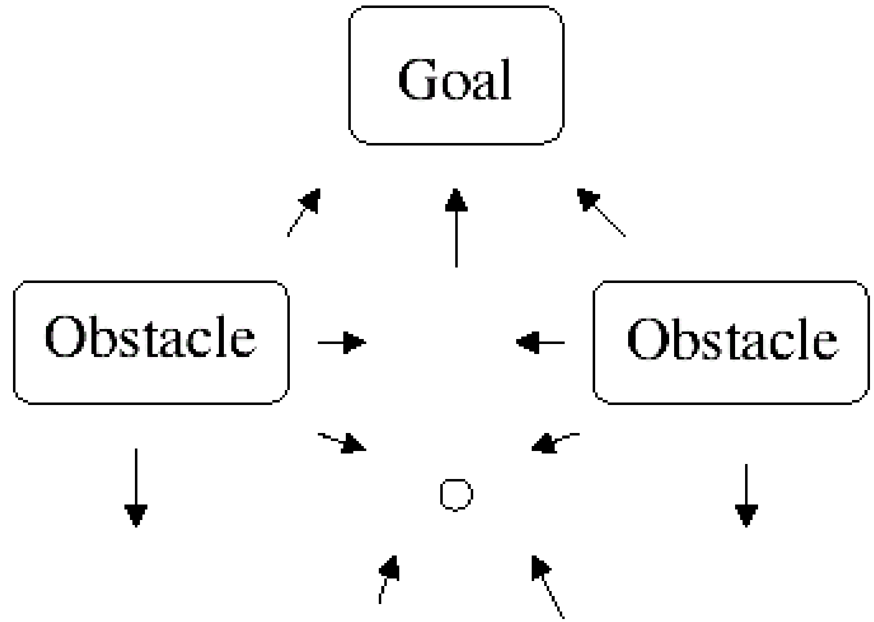
- direct usage for control
- local evaluation

Problems:

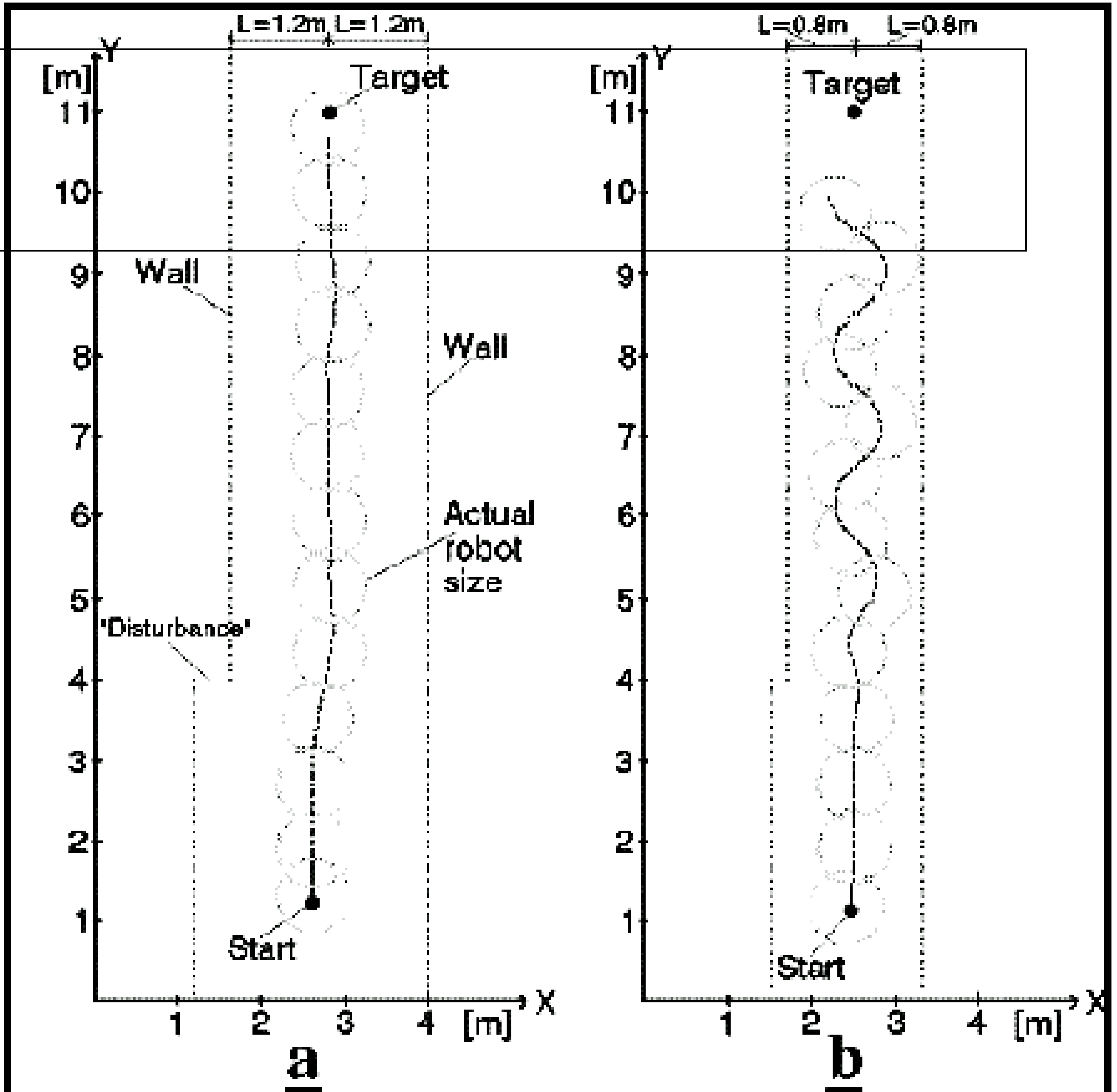
- local minima
  - Compensation of fields,
  - "Trap“ by close obstacles
- oscillating movements for
  - narrow areas
  - high speed

Potential field





Potential field



Stable motion in wide corridor
V=0.8m/s

Unstable motion in narrow corridor. V=0.8m/sec.

Burkhard

Potential field

Additional other fields, e.g.

- rotating fields
- random fields

can

- specify directions
- break symmetries
- avoid (some) local minima