Aalto University
School of Electrical Engineering

Inverse kinematics (closed form solution)

ELEC-C1320 Robotics

Pekka Forsman

Topics

- Closed-form solution
  - Under-actuated mechanisms, examples
  - Six-joint mechanisms
- Example problems

Closed-form solution

In the previous lecture we have shown how to determine the pose of the end-effector given the joint coordinates and optional tool and base transforms. *From manipulator control point of view we need to solve the inverse problem, i.e*. *given the desired pose of the end-effector ξ_E what are the required joint coordinates?*

If we know the desired Cartesian pose of the end-effector frame of the robot manipulator, what are the joint coordinates of the robot in order to move its tool to the desired pose? This is the inverse kinematics problem which is written in functional form as

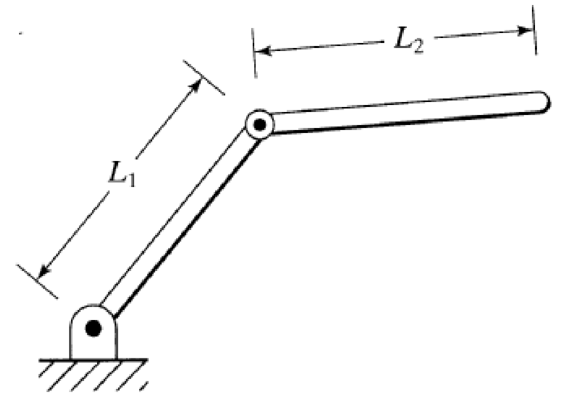
$$\boldsymbol{q} = \mathcal{K}^{-1}(\xi) \tag{7.5}$$

There are two main approaches to define the desired pose for the tool of the robot: on-line teaching and off-line programming

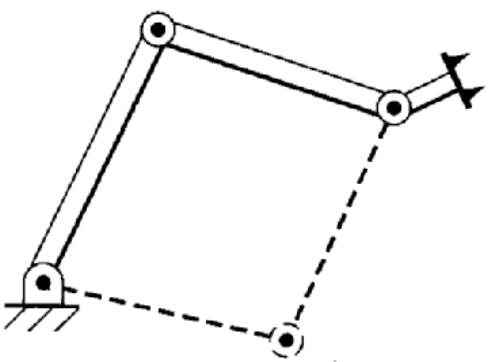
In **on-line teaching** the robot tool is moved with the teach pendant, in the actual robot cell, to the desired pose and the joint sensors are read and joint control set-point values stored as position parameters for the named location (*in many cases, the cartesian position parameters are also recorded*). The robot can then be moved to the same joint space position with very high precision, which is called in the context of robot manipulators **repeatability**.

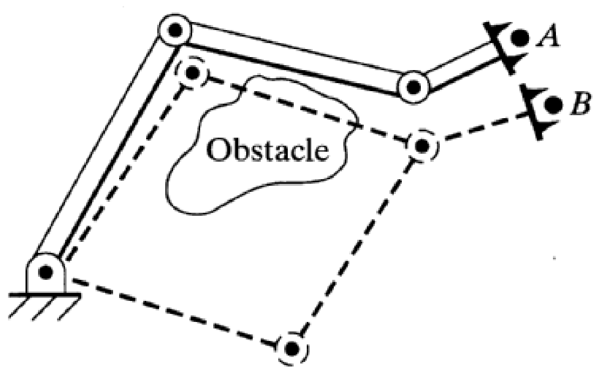
In **off-line programming** the robot path set-points are usually determined on the desired cartesian positions by means of the geometry of the workpiece and specification of the task. In order to move to a desired cartesian position, robot control system computes first the corresponding joint values by solving the inverse kinematics problem. It depends on the accuracy of the kinematic model of the robot, how precisely the tool of the real robot can be moved to the programmed position. This is often refered to as robot manipulator **accuracy**.

Depending on the range of motion of the joints, the control values of the manipulator joints to reach the desired location of the Tool Center Point, TCP, may not be unique. (*compare the manipulator in the figures to the elbow joint of human arm – arm has more restricted motion space*)

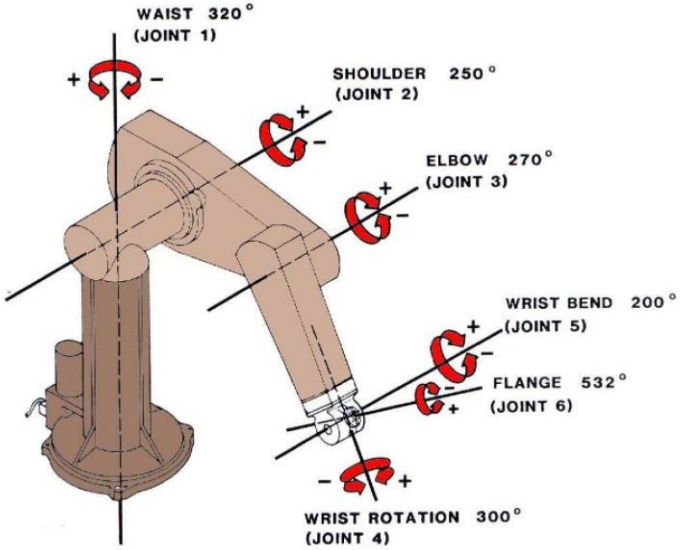


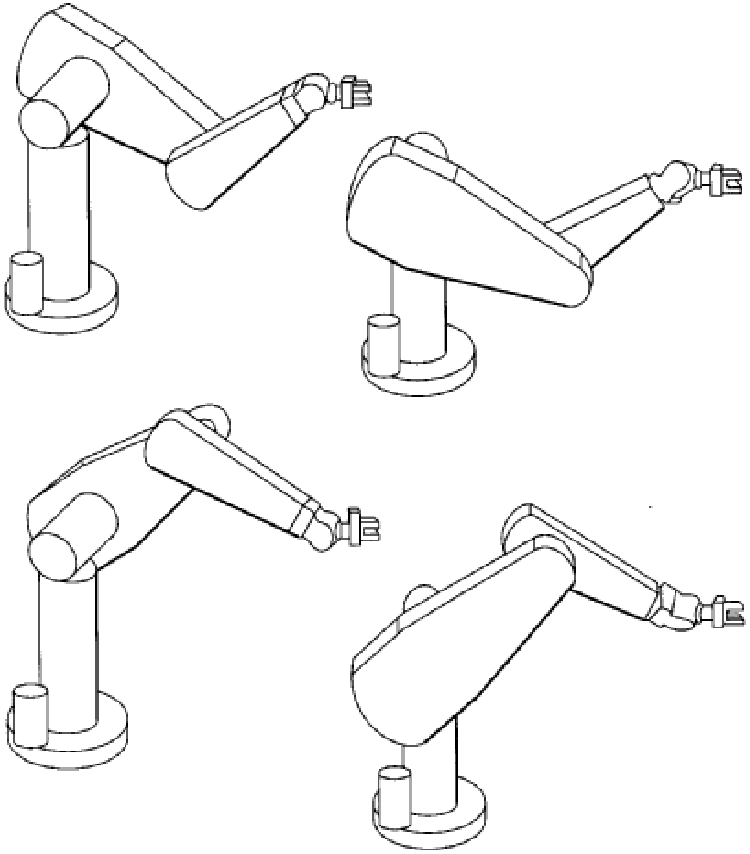
Availability of multiple joint configurations offers flexibility for the planning of robot motions (e.g. offers possibility to avoid collision with objects in the work space as depicted in the figure)





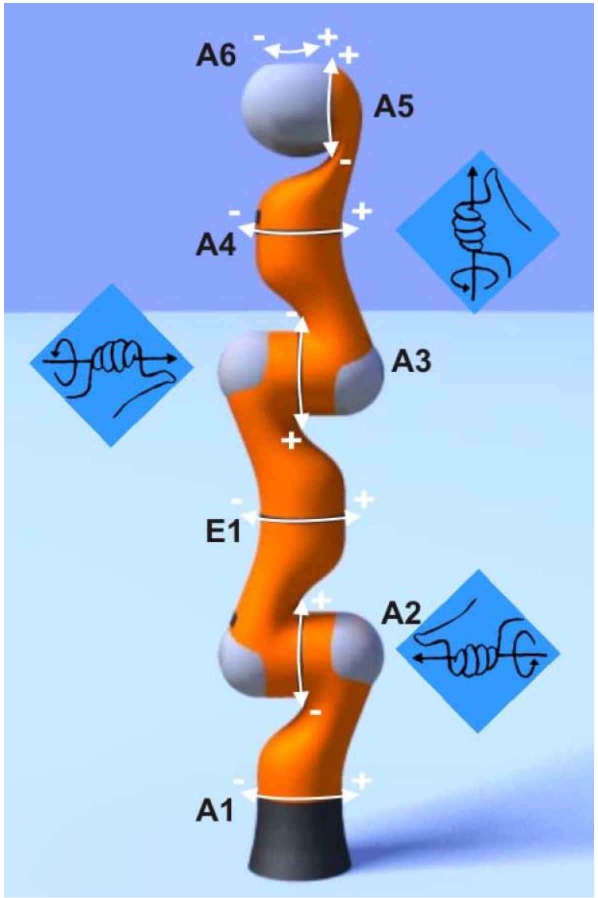
Puma 560 robot has six decrees of freedom and, for some joints, exceptionally wide range of motion. Therefore it has upto 8 different joint configurations to reach a certain TCP pose (four of them depicted in the figure ->).





If the number degrees of freedom of the manipulator is 7 or more, there is, in a general case, an **infinite number of joint coordinate combinations** to reach the desired pose of the end-effector in 3D. As an example of such kind of redundant robots, we have the KUKA LWR4+ robot shown in the figure to the right.

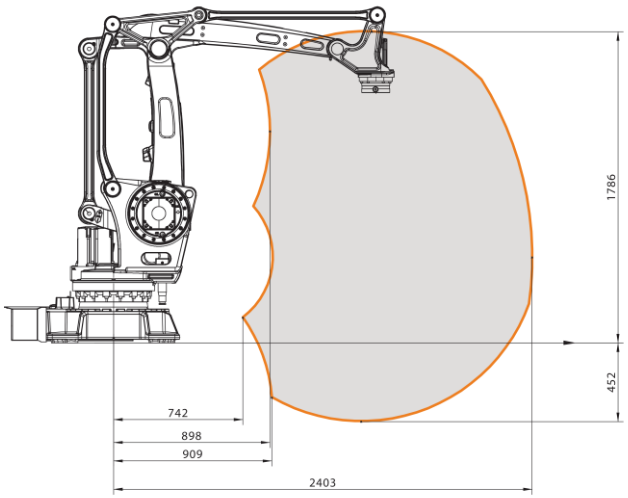
Axis	Range of motion, software-limited	Velocity without payload
A1 (J1)	+/-170°	100°/s
A2 (J2)	+/-120°	110°/s
E1 (J3)	+/-170°	100°/s
A3 (J4)	+/-120°	130°/s
A4 (J5)	+/-170°	130°/s
A5 (J6)	+/-120°	180°/s
A6 (J7)	+/-170°	180°/s



Under-actuated mechanisms, examples

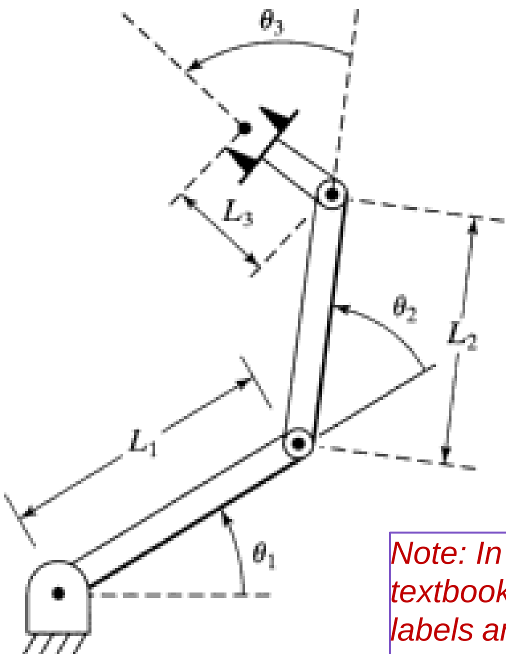
There are many tasks where the task space has less than six degrees of freedom. For example in arc-welding we are not interested in the orientation of the welding wire around its centerline axis. Therefore, industrial robots designed for arc-welding can accomplish their tasks with only 5 Degrees-Of-Freedom (DOF). (*Note however that modern arc-welding robots may have upto 7 DOF for extended motion flexibility*)

Also, packaging/palletizing robots moving boxes from one horizontal flat plane to another, can accoplish their tasks with only 4 DOF. In the figure the working range of ABB IRB 460 4-axes robot has been depicted.



For an under-actuated mechanism it is a good idea to rescrict the task space so that the availaible number of DOF is enough for operating within it.

For example with a 3-axes planar robot we can control the pose of the gripper within the plane (x,y,ϕ) (source Craig, 2005):

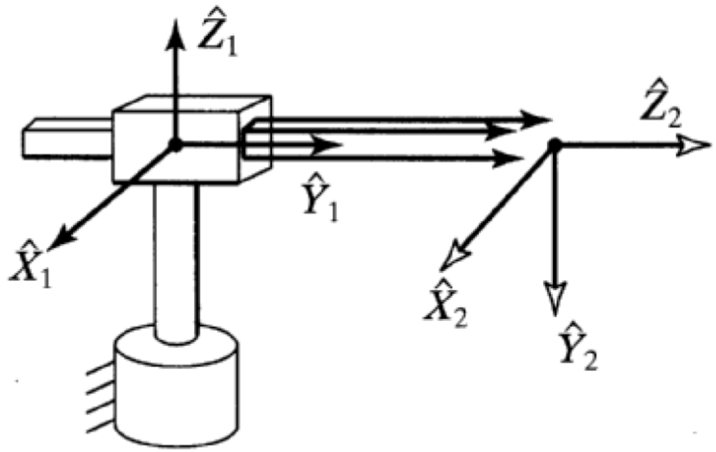


Cartesian space homogeneous transformation matrix corresponding to the desired TCP pose:

$$ {}^{B}_{W}T = \begin{bmatrix} c_\phi & -s_\phi & 0.0 & x \\ s_\phi & c_\phi & 0.0 & y \\ 0.0 & 0.0 & 1.0 & 0.0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

Note: In the notation of Craig's textbook both coordinate frame labels are placed on the left side of the transformation matrix symbol

With a 2-axes RP-robot (shown in the figure, source Craig, 2005) we can only control the x- and y-coordinates of the origin of the frame no. 2 within a horizontal plane. The orientation of frame 2 is ”dictated” by its xy-location and it cannot be controlled independently (*2 DOF is not enough to control pose in 3 dimensional motion space (x,y,θ), which makes sense*)



$$
{}^0_2T = \begin{bmatrix} \frac{y}{\sqrt{x^2+y^2}} & 0 & \frac{x}{\sqrt{x^2+y^2}} & x \\ \frac{-x}{\sqrt{x^2+y^2}} & 0 & \frac{y}{\sqrt{x^2+y^2}} & y \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

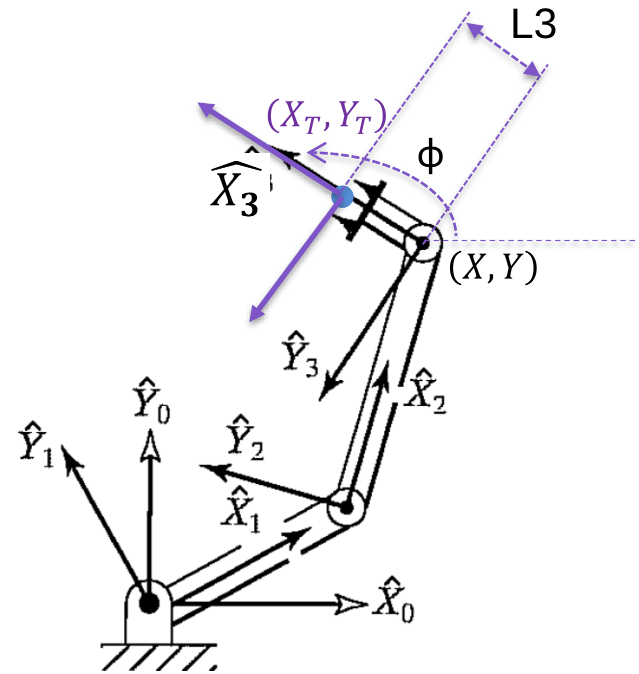
The inverse kinematics problem is usually solved by just considering the ”moving parts” of the mechanism. In the example this corresponds to the task space location (x,y,ϕ), i.e. location of the last link frame.

On the other hand, from the robot work task point of view we are interested in controlling the tool center point (TCP) of the robot. This is described with (X_T, Y_T, ϕ) in the figure.

So, for the inverse kinematics, we should first calculate the (x,y,ϕ)-coordinates from the given TCP-location (X_T, Y_T, ϕ). In case of our example problem, we get:

$$X = X_T - L3\cos\emptyset$$

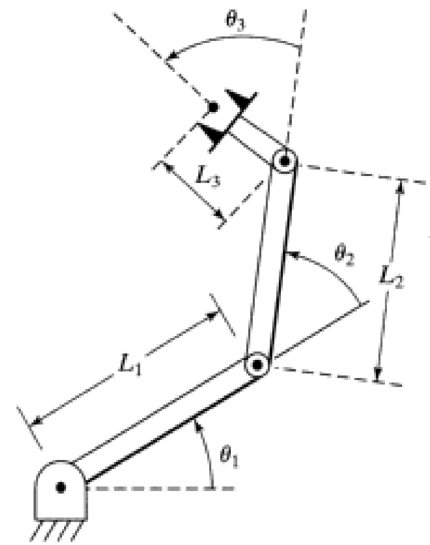
$$Y = Y_T - L3\sin\emptyset$$



Solving inverse kinematics with algebraic approach, example 3-axes manipulator moving on a plane (for more details see Craig, 2005)

First, solve the forward kinematics problem (extract the DH-parameters – here we follow the modified DH-notation)

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	0	L_1	0	θ_2
3	0	L_2	0	θ_3



Then form the forward kinematics model (arm matrix) and equate it with the cartesian space homogeneous transformation matrix which corresponds to the desired TCP (actually last link frame, 3) pose

$$ {}^{B}_{W}T = {}^{0}_{3}T = \begin{bmatrix} c_{123} & -s_{123} & 0.0 & l_1c_1 + l_2c_{12} \\ s_{123} & c_{123} & 0.0 & l_1s_1 + l_2s_{12} \\ 0.0 & 0.0 & 1.0 & 0.0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

From the two matrix equations

$$
{}^B_W T = {}^0_3 T = \begin{bmatrix} c_{123} & -s_{123} & 0.0 & l_1 c_1 + l_2 c_{12} \\ s_{123} & c_{123} & 0.0 & l_1 s_1 + l_2 s_{12} \\ 0.0 & 0.0 & 1.0 & 0.0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad {}^B_W T = \begin{bmatrix} c_\phi & -s_\phi & 0.0 & x \\ s_\phi & c_\phi & 0.0 & y \\ 0.0 & 0.0 & 1.0 & 0.0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

we can start forming equations between corresponding matrix elements:

$$
c_\phi = c_{123} \qquad s_\phi = s_{123} \qquad x = l_1 c_1 + l_2 c_{12} \qquad y = l_1 s_1 + l_2 s_{12}
$$

Next raise the x- and y-equations into power of two and apply the well-known trigonometric equations (*c_{12} means $\cos(\Theta_1 + \Theta_2)$ etc*.)

$$
c_{12} = c_1 c_2 - s_1 s_2 \qquad s_{12} = c_1 s_2 + s_1 c_2
$$

after what we get

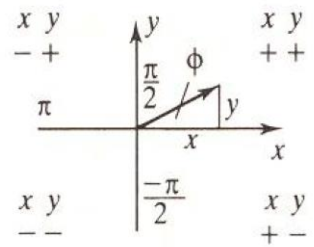
$$
x^2 + y^2 = l_1^2 + l_2^2 + 2 l_1 l_2 c_2
$$

which can be changed into
$$c_2 = \frac{x^2 + y^2 - l_1^2 - l_2^2}{2 l_1 l_2}$$

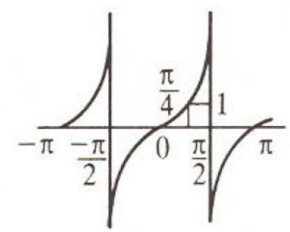
In oder to have a valid solution for the second joint, θ_2, the right hand side of the equation must have a value larger than -1 and smaller than 1. (*otherwise it means that we are trying to command the robot to move to a TCP-location which is outside the robot's motion space*)

Now, we could just take arccos function of both sides of the equation but the solution would not be unique for the whole ±180 motion space. Therefore it is common practise to form an equation also for sin(angle) and then apply the Atan2-function

```
{atan2 function – atan2(y, x)}
{returns φ for −π <= φ < π}
if (x = 0)  or (y = 0) then
            if (x = 0) and (y is positive) then φ = +π/2
            else φ = −π/2
            if (y = 0) and (x is positive) then φ = 0
            else φ = −π
else        {non zero values of x and y}
            sign = x × y/abs(x × y)
            φ = sign × tan⁻¹(abs(y/x))
return φ
```



$$\tan(\phi) = \frac{y}{x} = \frac{\sin(\phi)}{\cos(\phi)}$$



$$\phi = \tan^{-1} \frac{y}{x}$$

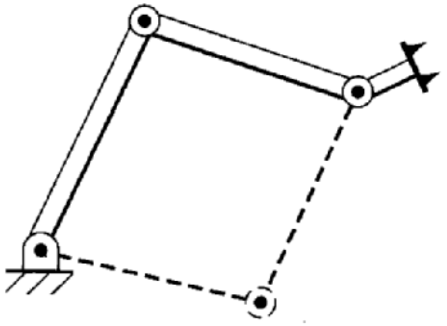
For the sin-function we get

$$s_2 = \pm\sqrt{1 - c_2^2}$$

and now we can solve θ_2

$$\theta_2 = \mathrm{Atan2}(s_2, c_2)$$

There will be two solutions for θ_2 corrresponding to different signs of the $\sin(\theta_2)$ - equation:



In order to solve θ_1 we first modifify the x- and y-equations

$$x = l_1 c_1 + l_2 c_{12} \qquad y = l_1 s_1 + l_2 s_{12}$$

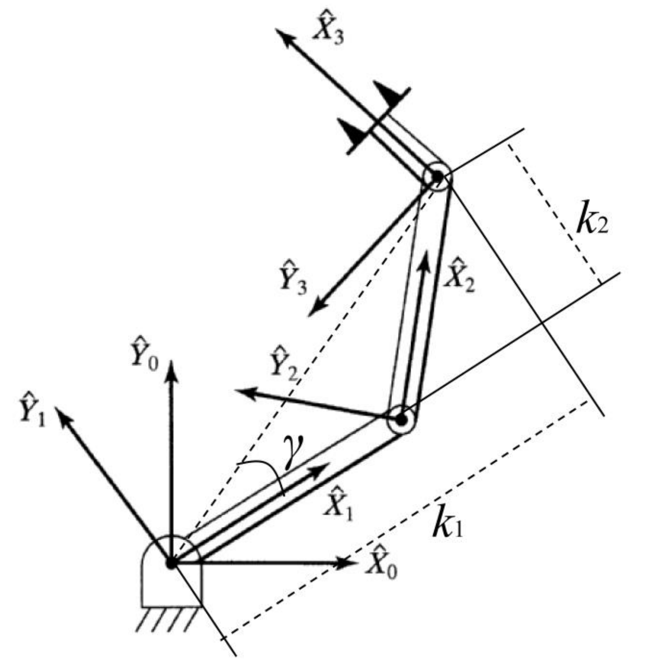
into a different form

$$x = k_1 c_1 - k_2 s_1 \qquad y = k_1 s_1 + k_2 c_1$$

where

$$k_1 = l_1 + l_2 c_2 \qquad k_2 = l_2 s_2$$

Next we can write (compare the figure ->)



$$r = +\sqrt{k_1^2 + k_2^2} \qquad \gamma = \text{Atan2}(k_2, k_1)$$

and we can form another expression for k_1 and k_2

$$k_1 = r \cos \gamma \qquad k_2 = r \sin \gamma$$

and by assigning the result into x- and y- equations of the previous slide we get

$$\frac{x}{r} = \cos\gamma \cos\theta_1 - \sin\gamma \sin\theta_1 \qquad \frac{y}{r} = \cos\gamma \sin\theta_1 + \sin\gamma \cos\theta_1$$

which can be changed into

$$\cos(\gamma + \theta_1) = \frac{x}{r} \qquad \sin(\gamma + \theta_1) = \frac{y}{r}$$

now we can apply the Atan2(sin(), cos())-function

$$\gamma + \theta_1 = \text{Atan2}\left(\frac{y}{r}, \frac{x}{r}\right) = \text{Atan2}(y, x)$$

And by assigning the solution for γ we get

$$\theta_1 = \text{Atan2}(y, x) - \text{Atan2}(k_2, k_1)$$

the value for the remaining manipulator joint, θ_3, can be solved from

$$\theta_1 + \theta_2 + \theta_3 = \text{Atan2}(s_\phi, c_\phi) = \phi$$

Solving inverse kinematics with geometric approach,

example 3-axes manipulator moving on a plane – same as in the previous example (for more details see Craig, 2005)

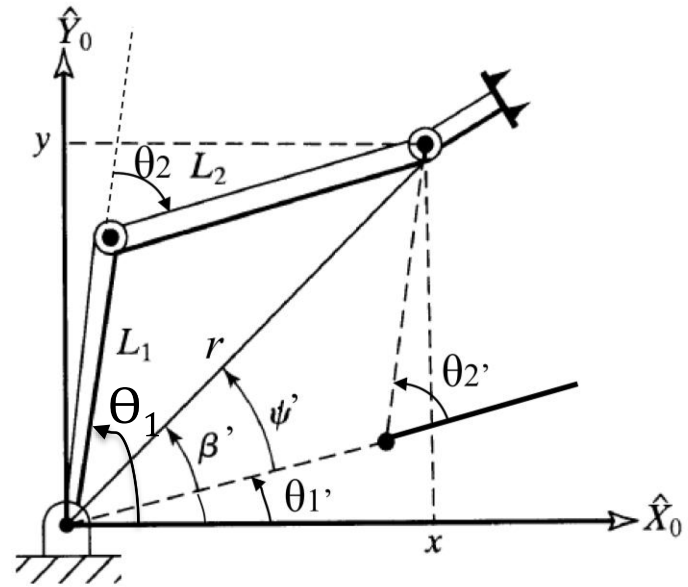
In the figure, the ”elbow up” and ”elbow down” joint angles are marked without and with the ” ’ ”-marking.

Let’s first solve Θ_2 by applaying the ”law of cosines”:

$$x^2 + y^2 = l_1^2 + l_2^2 - 2l_1 l_2 \cos(180 + \theta_2)$$

because $$\cos(180 + \theta_2) = -\cos(\theta_2)$$

we get $$c_2 = \frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1 l_2}$$



In order to have a valid solution, the target point of the arm must be closer than the sum of the lengths of the upper and lower links

$$l_1 + l_2 \geq \sqrt{x^2 + y^2}$$

The value of Θ_2 to be acquired by applying law of cosines here will be negative (between -180º and 0). This is because otherwise the corresponding triangle, used for forming the equations does not exist. The dual angle, for the "elbow down" case is found (by symmetry) to be $-\Theta_2$, i.e. a positive angle.

To solve for Θ_1 we first form formulas for the angles β and Ψ (compare the figure)

$$\beta = \text{Atan2}(y, x)$$

$$\cos\psi = \frac{x^2 + y^2 + l_1^2 - l_2^2}{2l_1\sqrt{x^2 + y^2}}$$

Ψ-angle must be between 0 and 180º for a valid solution, then we get

$$\theta_1 = \beta \pm \psi$$

where the +sign is for the ”elbow up”- case and –sign for the ”elbow down” configuration

Because all three rotation axes are paralled (normal to the plane), third joint angle, Θ_3, can be solved from the equation:

$$\theta_1 + \theta_2 + \theta_3 = \emptyset$$

Solving inverse kinematics with algebraic and geomeric approach, example pan/tilt mechanism pointing a laser range finder

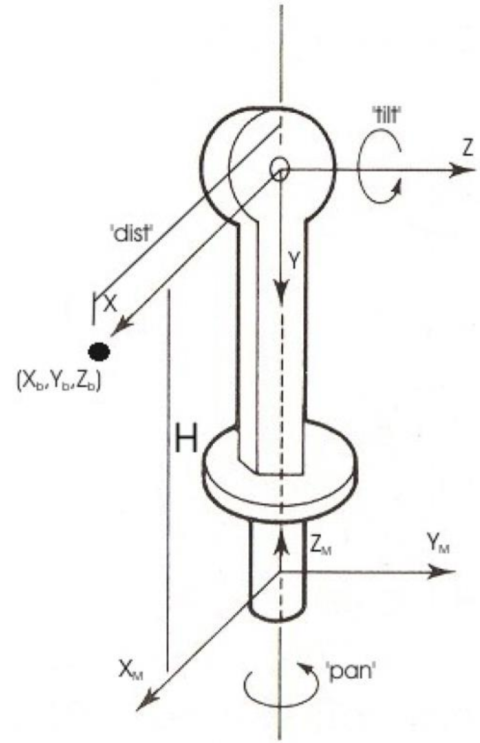
In this example we first form equations for the x-, y- and z-coordinates of the laser beam hit point (x_b,y_b,z_b) by studying the mechanism (DH-parameter formulation not applied here) – *i.e. solve the forward kinematics problem*

$$z = H - dist * \sin(tilt) \quad (1)$$

$$x = \cos(pan) * dist * \cos(tilt)$$

$$= dist * \cos(pan) * \cos(tilt) \quad (2)$$

$$y = dist * \sin(pan) * \cos(tilt) \quad (3)$$



Now, we apply algebraic approach to solve the *pan* and *tilt* angles and the *dist*ance value. From the equations in the previous slide we get

(2) ja (3) →

$$\frac{y}{x} = \frac{dist \cdot \sin(pan) \cdot \cos(tilt)}{dist \cdot \cos(pan) \cdot \cos(tilt)}$$

$$= \tan(\text{pan})$$

$$\Rightarrow \quad \text{pan} = \text{atan}(\frac{y}{x})$$

preferably use atan2(y,x)
instead of simple atan()

(1) ja (3) →

$$\frac{H-z}{y} = \frac{dist \cdot \sin(tilt)}{dist \cdot \sin(pan) \cdot \cos(tilt)}$$

$$= \frac{\tan(tilt)}{\sin(pan)}$$

$$\Rightarrow \tan(\text{tilt}) = \frac{(H-z) \cdot \sin(pan)}{y}$$

$$= \frac{(H-z) \cdot \sin(a\tan(\frac{y}{x}))}{y}$$

$$\Rightarrow \text{tilt} = \text{atan}\left(\frac{(H-z) \cdot \sin(a\tan(\frac{y}{x}))}{y}\right)$$

And finally, for the distance value we get

$$\mathrm{dist} = \frac{H - z}{\sin(tilt)}$$

$$= \frac{H - z}{\sin\left(\mathrm{atan}\left(\frac{(\mathrm{H} - \mathrm{z}) \cdot \sin\left(\mathrm{atan}\left(\frac{\mathrm{y}}{\mathrm{x}}\right)\right)}{\mathrm{y}}\right)\right)}$$

preferably use atan2(y,x)
instead of simple atan()

Alternatively, we can solve the inverse kinematics problem geometrically.
By studying the figure we get:

$$\mathrm{dist}^2 = (\mathrm{H} - \mathrm{z})^2 + \mathrm{x}^2 + \mathrm{y}^2$$

$$\Rightarrow \mathrm{dist} = \sqrt{(H - z)^2 + x^2 + y^2}$$

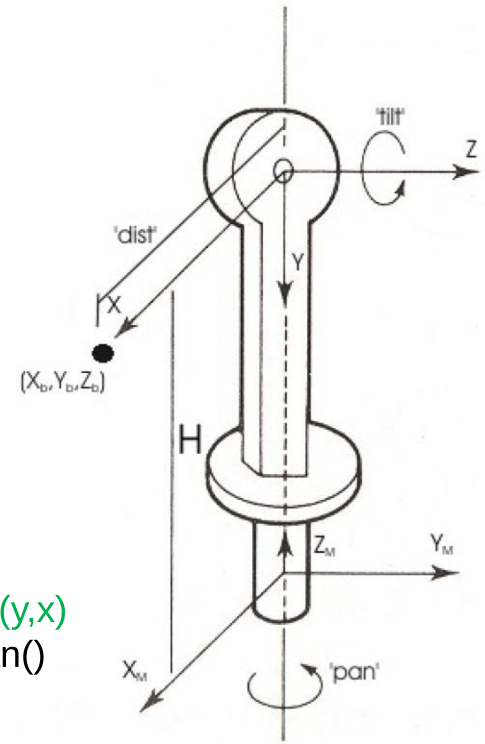
$$\tan(\mathrm{tilt}) = \frac{H - z}{\sqrt{x^2 + y^2}}$$

$$\Rightarrow \mathrm{tilt} = \mathrm{atan}\left(\frac{H - z}{\sqrt{x^2 + y^2}}\right)$$

$$\tan(\mathrm{pan}) = \left(\frac{y}{x}\right)$$

$$\Rightarrow \mathrm{pan} = \mathrm{atan}\left(\frac{y}{x}\right)$$

preferably use atan2(y,x)
instead of simple atan()



Six-joint mechanisms

"All robot manipulators with revolute and prismatic joints having a total of six degrees of freedom in a single series chain are solvable" (Craig, 2005). Generally speaking, the solution is a numerical one. Numerical solutions of nonlinear systems can be quite slow to compute and therefore analytical (closed-form) solutions are preferred.

Robot manipulators with six degrees of freedom can be solved analytically only in special cases. Conditions for the existence of a closed-form solution include, for example, that the robot has several intersecting joint axes or many of the twist angles, α_j, are 0 or $\pm 90^o$.

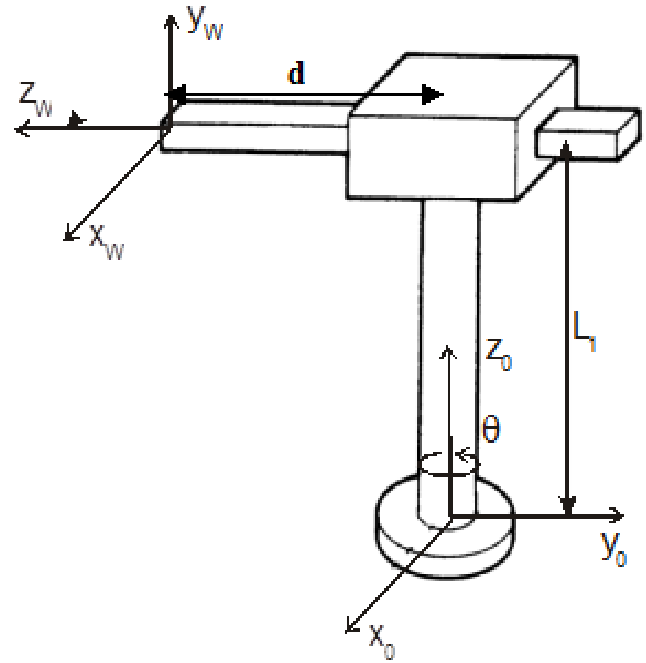
A sufficient condition that a manipulator with six rotational joints has a closed-form solution is that three neighbouring joint axes intersect at a point. This structural feature is part of PUMA 560-robot and is applied also with most of modern six-axes robot manipulators.

In (Craig, 2005, pp 114-125) some systematic approaches for computing closed form inverse kinematics for a six-DOF manipulator are presented.

Example problems

1. In the figure to the right, a two decree-of-freedom manipulator is shown in its home/zero position, the first degree-of-freedom (dof) is a rotational joint (controlling the orientation of the upper link on the horizontal plane), ϴ, and the second dof is a translational joint (controlling the length of the upper link), d. (The upper link is above the negative y-axes of the 0-frame when the control angle, ϴ, has a zero value).

Find the inverse kinematic transform for the manipulator. Describe also, for which of the x,y,z-positions of the origin of the (W) frame a reachable inverse kinematic solution exists, i.e. what is the Cartesian task space of this simple manipulator (answer , for example, in the form of equations or inequalities)?



Solution

In the zero-configuration (i.e. all joint control values equal zero) the manipulator upper arm is above the negative y_0-axis. Then for positive angles of θ, the arm starts to rotate counterclockwise (as seen from above). The rotation angle can be calculated with the equation:

$$\theta = atan2(x, -y)$$

Joint variable d, on the other hand, equals the length of the projection of the upper arm on the x_0y_0-plane (*actually the arm is always oriented parallel to the* x_0y_0*-plane*):

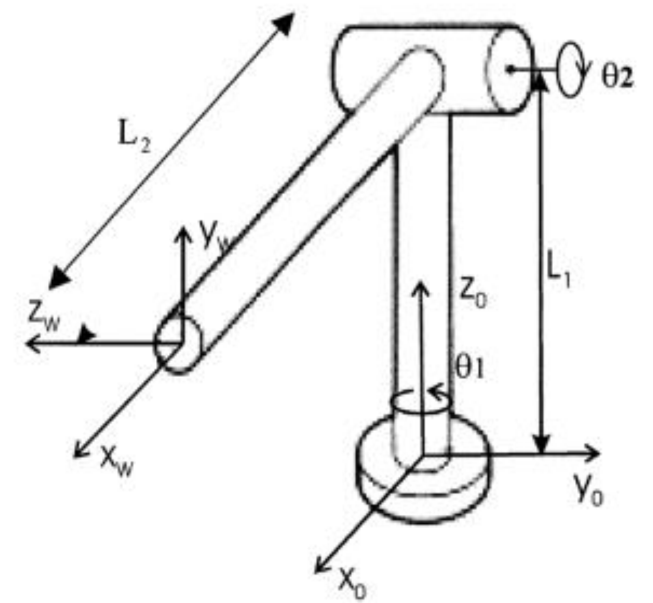
$$d = \sqrt{x^2 + y^2}$$

where x and y are the coordinates of the origin of the w-frame on the x_0y_0-plane.

The reachable task space for the manipulator is the disk centered at $x=0, y=0, z=L_1$ and with the radius d_{max}.

2. In the figure to the right, a two decrees-of-freedom manipulator in its home/zero position is shown (upper arm oriented horizontally above the x_0-axis). Both degrees-of-freedom (dof) are rotational (the first rotating the upper link on the horizontal plane, Θ_1, and the second tilting the upper link with respect to the horizontal plane, Θ_2).

Find the inverse kinematic transform for the manipulator. Describe also, for which of the x,y,z-positions of the origin of the (W) frame a reachable inverse kinematic solution exists i.e. what is the Cartesian task space of this simple manipulator (answer , for example, in the form of an equation or an inequality)



Solution

In the zero-configuration (i.e. all joint control values equal zero) the manipulator upper arm is above the positive x_0-axis. Then for positive angles of θ_1, the arm starts to rotate counterclockwise (as seen from above). The rotation angle can be calculated from the xy-coordinates of the origin of the w-frame as follows:

$$\theta_1 = \text{atan2}(y,x)$$

Only exception is the case when the upper arm points straight upwards and the x- and y-coordinates equal zero. In this case θ_1 is undefined (singular configuration).

The rotation angle of the shoulder, θ_2, can be calculated as follows:

$$\theta_2 = \text{atan2}(z-L_1, \sqrt{x^2 + y^2})$$

The reachable task space for the manipulator is a sphere centered at $x=0, y=0, z=L_1$ and with the radius L_2.

Recommended reading:

Craig, J.J, Introduction to Robotics: Mechanics and Control, Third Edition, Prentice Hall, 2005, pages 101-114, 127

Peter Corke, Robotics, Vision and Control, Fundamental Algorithms in MATLAB, Second Edition, Springer, 2017, page 205.