# Development of Complex Curricula for Molecular Bionics and Infobionics Programs within a consortial* framework**

**Consortium leader**

## PETER PAZMANY CATHOLIC UNIVERSITY

**Consortium members**

## SEMMELWEIS UNIVERSITY, DIALOG CAMPUS PUBLISHER

The Project has been realised with the support of the European Union and has been co-financed by the European Social Fund ***

**Molekuláris bionika és Infobionika Szakok tananyagának komplex fejlesztése konzorciumi keretben

***A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

# Neuromorph Movement Control

**(Neuromorf mozgás vezérlés)**

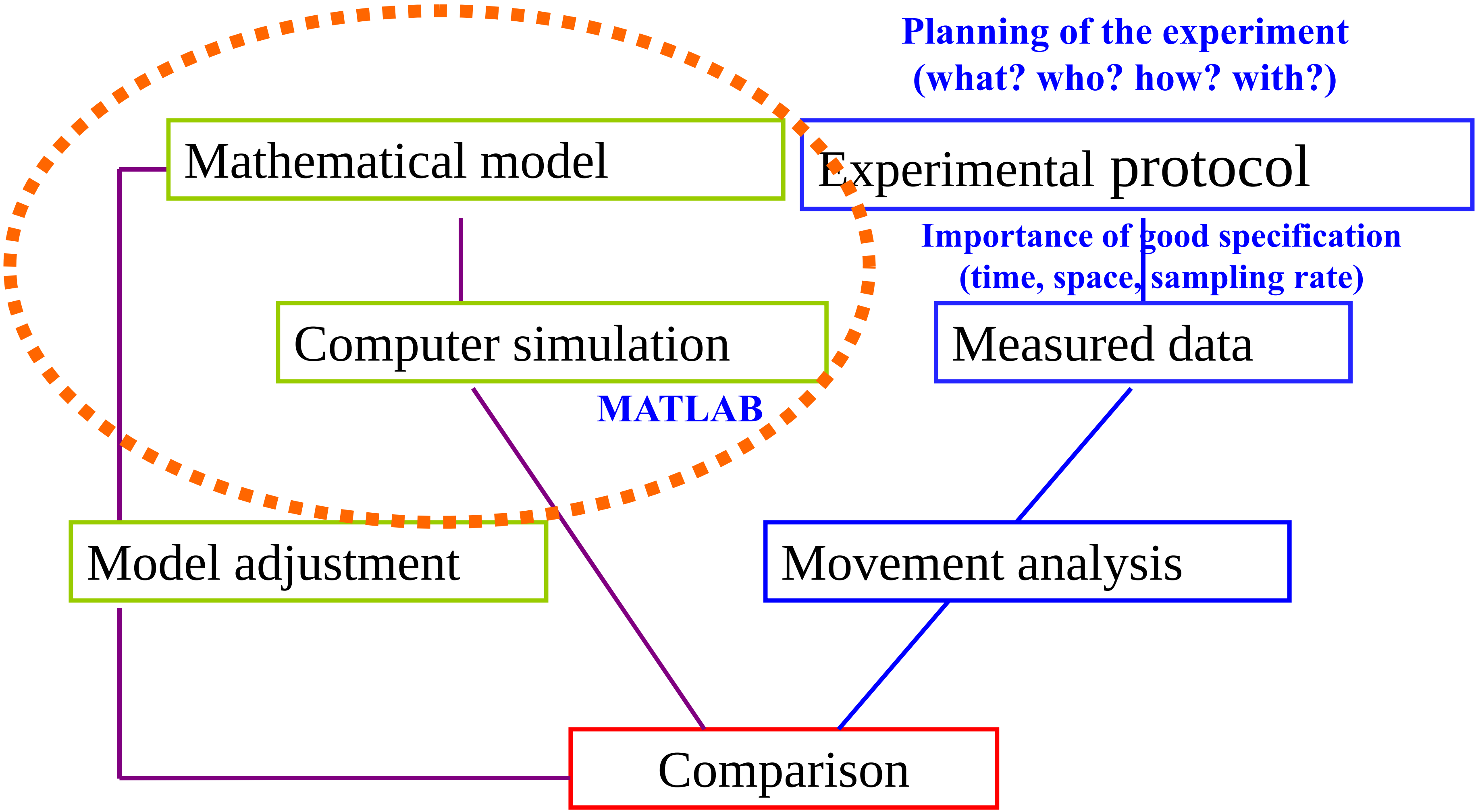
## Solving the issue of Direct Kinematics in Biology and Robotics

**(Direkt kinematikai probléma megoldása a biológiában és robotikában)**

**József LACZKÓ PhD; Róbert TIBOLD**

# Main points of the lecture

- The relation between mathematical modeling and experimental protocols
- General aims of modeling
- Biology vs. Robotics
    - Definitions of direct kinematic (DK) problem
    - Computation of joint coordinates
    - Computation of velocity vector in the endpoint
    - Rotation and transformation matrices (Denavit-Hartenberg algorithm)
- Introduction of a MATLAB based DK solver
    - GUI solution
- Definition and determination of the angular acceleration in 3D space


Planning of the experiment
(what? who? how? with?)
Mathematical model
Experimental protocol
Importance of good specification
(time, space, sampling rate)
Computer simulation
Measured data
MATLAB
Model adjustment
Movement analysis
Comparison

**Ultimate aim:** to describe the mathematical relation between the activity of motoneurons and angular changes in the joints. **This involves kinematics and dynamics.**

– *General objective:*
1. Investigate the relation between neural impulses and muscle forces.
2. Investigate the dependence of joint angular changes on muscle forces.
3. Investigate the dependence of the position of the endpoint of a limb on the inter-segmental angles in the joints of the limb
4. Investigate the dependence of the velocity of the endpoint of a limb on the angular velocities in the joints of the limb

- Those processes of the central nervous system system (CNS) that can’t be measured experimentally might be discerned by theoretical methods (e.g. modeling).

**Question:** Is there any kind of optimization criteria the CNS employs for “optimal” execution of motor tasks?

- Work
- Energy
- Smoothness
- Precision

- The motion of a human body segment respect to an adjacent segment is a combination of rotation and translation.
- In most human movements (e.g. walking or reaching and object with an arm) the translation between the adjacent segments can disregarded since its magnitude is very small comparing the whole movement.
- Thus in human direct kinematics, joint motions are considered as pure rotational movements, while the result of the rotations in a multijoint system's joints may be a translation of the body (e.g. during walking)

# 2 main directions of implementation

## Bottom-Up(BU) = Solving the Direct Kinematic (DK) problem

- *Given: Measured angular changes in the joints of the limb*
- *Question: Position and trajectory of the endpoint of the limb*

## Top-Down(TD) = Solving the Inverse Kinematic problem

- *Given: desired position and trajectory of the endpoint of the limb*
- *Question: joint angles and angular velocities in the joints of the limb*

The direct kinematic problem (Bottom Up) has a unique solution.

The inverse kinematic problem (Top Down) has usually an infinity of different solutions

# Direct kinematics: Biology vs. Robotics

- **Biology and Robotics as two different fields:**
  - might handle direct kinematics (forward kinematics) the same way BUT
  - in general both fields solve DK problem differently

- **Biology:** approaches the DK problem based on measurements

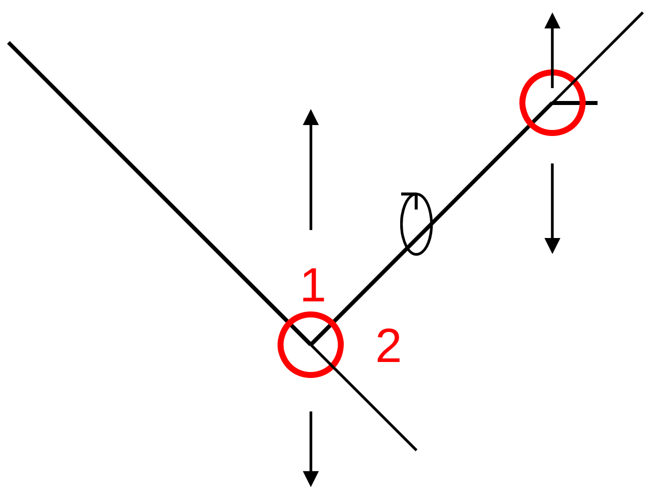
  Measuring of joint coordinates → joint angles can be calculated using vector algebra

- **Robotics:** has the solution of DK problem based on mathematical approaches (applying rotation and transformation matrices)

# Definition of limb from biological point of view (PoV)

Limb: can be regarded as an object containing

- different **segments** linked to each other in **joints**
- **Segments** are capable of **rotating** around **joints** and
- **around their own longitudinal axis (***supination,pronation***)**




1.:Intersegmental joint angles: the angle between adjacent segments

•Smaller than $180^o$

2.:Outer segmental joint angles: $180^o$ – Intersegmental joint angles

**Limb (e.g. a human lower or upper extremity) is regarded as a kinematic chain**

- One end of the chain is fixed. If the other end is not fixed then we speak about **open kinematic chain** and the other end is called endpoint of the limb or working point of the limb.
- If the the working point's position is fixed, then the kinematic chain is a **closed kinematic chain.**

**The size of the workspace depends on**:

- Degrees of freedom (DoF) of joints
- The number of DoF of the limb:

  the sum of the DoF of individual joints
- Segment lengths

# **Direct Kinematics** (*forward kinematics*)

a) compute the position of the endpoint of the limb if the first joint is fixed and the joint angles and segment lengths are given.

b) compute the velocity of the endpoint of the limb if the first joint is fixed and the joint angles, angular velocities and segment lengths are given.

Joint angles and angular velocities are given in a reference frame defined by independent axes of rotations.
The number of independent axes of rotations depend on the degrees of freedom of the given joint.

**Velocity:** the time rate of change of position

**Speed:** is the magnitude of velocity (rate of change of distance covered by the moving point)

**Distance:** is the magnitude of traveled path

**Displacement:** is the difference between two positions (e.g. the starting and the final position of a point)

**The sequence of translational displacements of a point:** is commutative (the order of consecutive translations do not effect the final displacement)

**The sequence of rotational displacements of a point:** is not commutative in the 3D space.

- If muscle forces and muscle geometry (origin and insertion surfaces and lengths of muscles)
- Is known the joint torque generated by the muscle force can be computed.
- If limb geometry and joint torques are known than the kinematics of the movement can be computed.
- This is called direct dynamics
- If the kinematics of the joints are given than joint torques and muscle forces can be computed by inverse dynamics.
- The inverse dynamical problem has usually an infinity of different solutions.

# Sensation – Execution

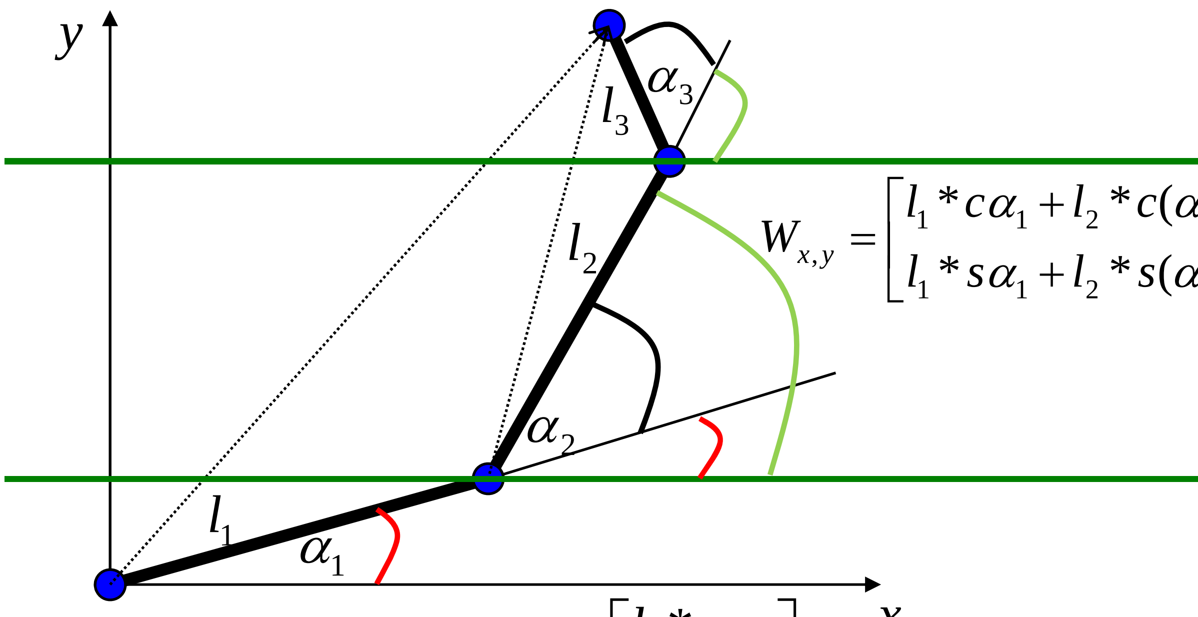
## Receptors sensitive to angular changes – Actuator to move

- Direct and Inverse problems
  - **Kinematics** (*describes the motion of bodies (objects) and systems (groups of objects) without consideration of the forces that cause the motion*)
    - Joint angle ⟷ Limb position
  - **Dynamics** (*the time evolution of physical processes*)
    - Angular acceleration (torques) ⟷ Endpoint forces.

# Computation of Joint coordinates

$$EP_{x,y} = \begin{bmatrix} l_1 * c\alpha_1 + l_2 * c(\alpha_1 + \alpha_2) + l_3 * c(\alpha_1 + \alpha_2 + \alpha_3) \\ l_1 * s\alpha_1 + l_2 * s(\alpha_1 + \alpha_2) + l_3 * s(\alpha_1 + \alpha_2 + \alpha_3) \end{bmatrix}$$

s=sine c = cosine



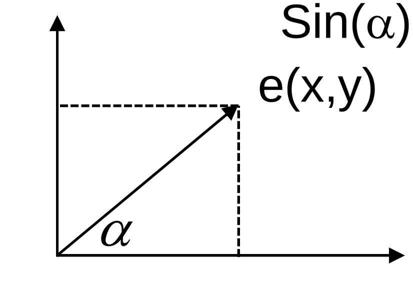

$$W_{x,y} = \begin{bmatrix} l_1 * c\alpha_1 + l_2 * c(\alpha_1 + \alpha_2) \\ l_1 * s\alpha_1 + l_2 * s(\alpha_1 + \alpha_2) \end{bmatrix}$$

$S = [0,0]$

$$E_{x,y} = \begin{bmatrix} l_1 * c\alpha_1 \\ l_1 * s\alpha_1 \end{bmatrix}$$

Cos(α)=x

Sin(α) =y



# Computation of Joint coordinates – An example with given joint angles

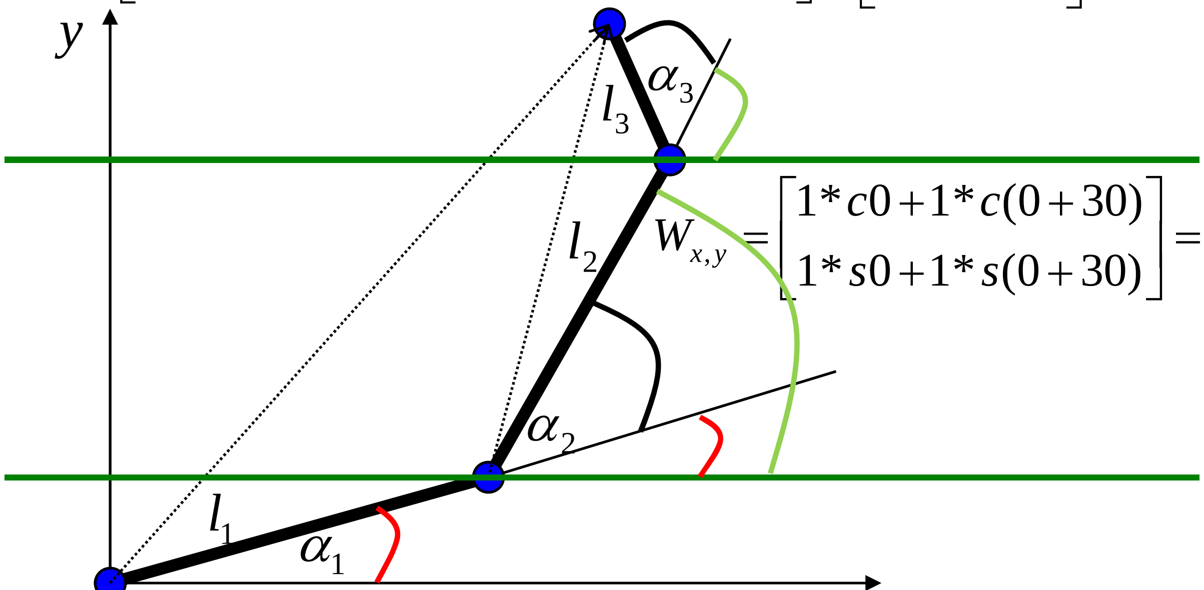
**$l_1=l_2=l_3=1$**

**$\alpha 1=0^o$**

**$\alpha 2=30^o$**

**$\alpha 3=60^o$**

$$EP_{x,y}=\begin{bmatrix}1*c0+1*c(0+30)+1*c(0+30+60)\\1*s0+1*s(0+30)+1*s(0+30+60)\end{bmatrix}=\left[1+\frac{\sqrt{3}}{2},1.5\right]$$



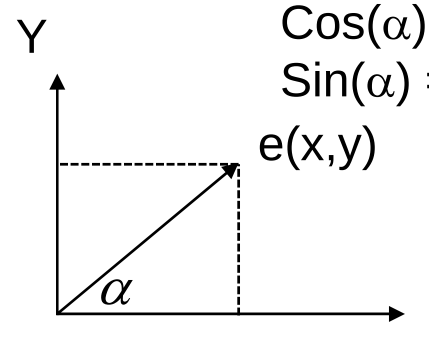

$$W_{x,y}=\begin{bmatrix}1*c0+1*c(0+30)\\1*s0+1*s(0+30)\end{bmatrix}=\left[1+\frac{\sqrt{3}}{2},0.5\right]$$

$$S=[0,0]$$

$$E_{x,y}=\begin{bmatrix}1*c0\\1*s0\end{bmatrix}=[1,0]$$

Cos(α)=x

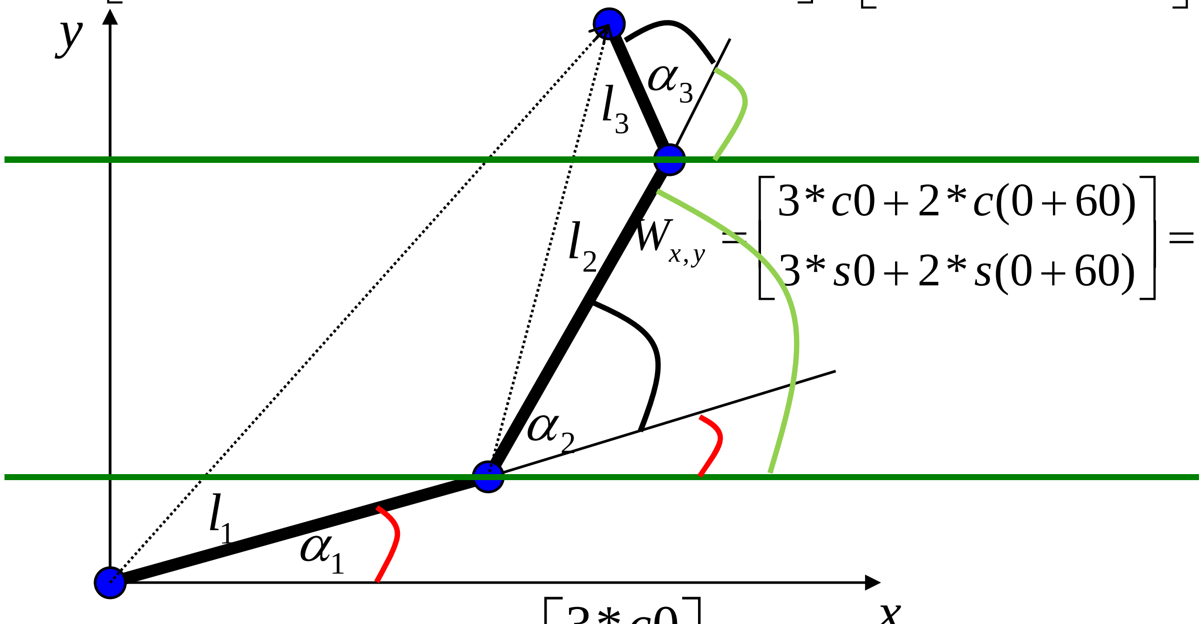
Sin(α) =y



# Computation of Joint coordinates – An example with given joint angles

$$EP_{x,y} = \begin{bmatrix} 3*c0 + 2*c(0+60) + 1*c(0+60+90) \\ 3*s0 + 2*s(0+60) + 1*s(0+60+90) \end{bmatrix} = \left[4 - \frac{\sqrt{3}}{2}, \sqrt{3} + 0.5\right]$$

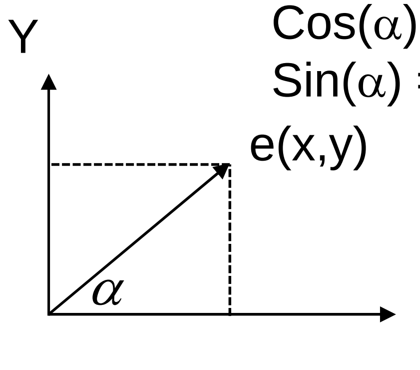
$l_1=3$
$l_2=2$
$l_3=1$
$\alpha 1=0^o$
$\alpha 2=60^o$
$\alpha 3=90^o$




$$W_{x,y} = \begin{bmatrix} 3*c0 + 2*c(0+60) \\ 3*s0 + 2*s(0+60) \end{bmatrix} = \left[4, 3 + 2*\frac{\sqrt{3}}{2}\right]$$

$$S = [0,0]$$

$$E_{x,y} = \begin{bmatrix} 3*c0 \\ 3*s0 \end{bmatrix} = [3,0]$$




Cos(α)=x
Sin(α) =y
e(x,y)

# Computation of Joint coordinates – An example for special posture

$$EP_{x,y} = \begin{bmatrix} 1*c45+1*c(45+0)+1*c(45+0+0) \\ 1*s45+1*s(45+0)+1*s(45+0+0) \end{bmatrix} = \left[ 3\frac{\sqrt{2}}{2}, 3\frac{\sqrt{2}}{2} \right]$$

**$l_1=1$**

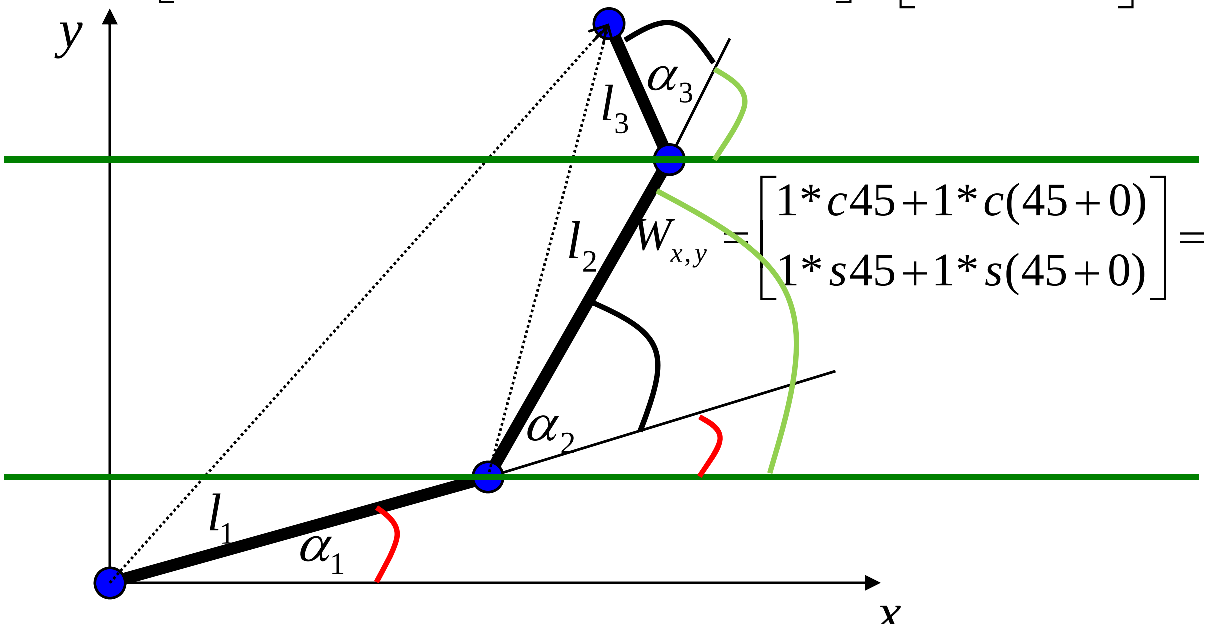
**$l_2=1$**

**$l_3=1$**

**$\alpha 1=45^o$**

**$\alpha 2=180^o$**

**$\alpha 3=180^o$**



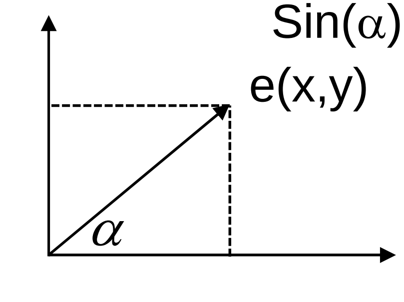

$$W_{x,y} = \begin{bmatrix} 1*c45+1*c(45+0) \\ 1*s45+1*s(45+0) \end{bmatrix} = \left[ 2\frac{\sqrt{2}}{2}, 2\frac{\sqrt{2}}{2} \right]$$

$$S = [0,0]$$

$$E_{x,y} = \begin{bmatrix} 1*c45 \\ 1*s45 \end{bmatrix} = \left[ \frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2} \right]$$

Cos(α)=x

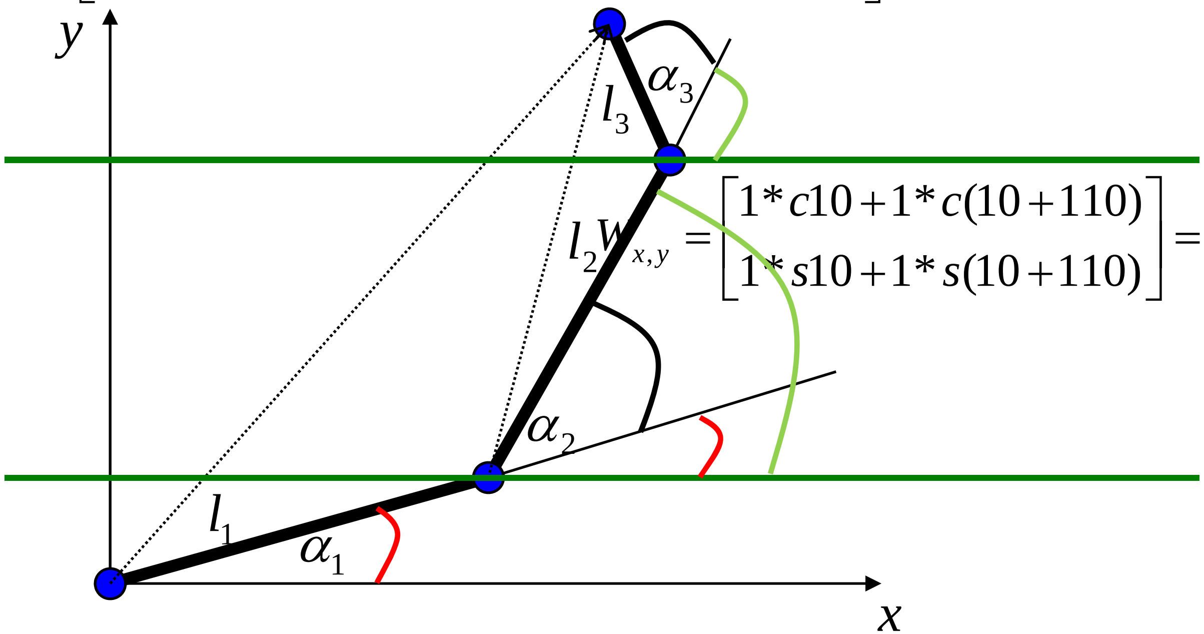
Sin(α) =y



# Computation of Joint coordinates – An example for special posture

$$EP_{x,y} = \begin{bmatrix} 1*c10+1*c(10+110)+1*c(10+110+125) \\ 1*s10+1*s(10+110)+1*s(10+110+125) \end{bmatrix} = [0.0622, 0.1334]$$

$$W_{x,y} = \begin{bmatrix} 1*c10+1*c(10+110) \\ 1*s10+1*s(10+110) \end{bmatrix} = [0.4848, 1.0397]$$

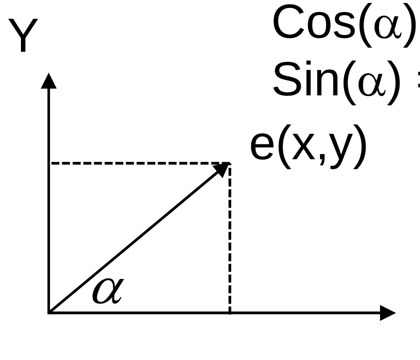



$$S = [0,0]$$

$$E_{x,y} = \begin{bmatrix} 1*c10 \\ 1*s10 \end{bmatrix} = [0.9848, 0.1736]$$

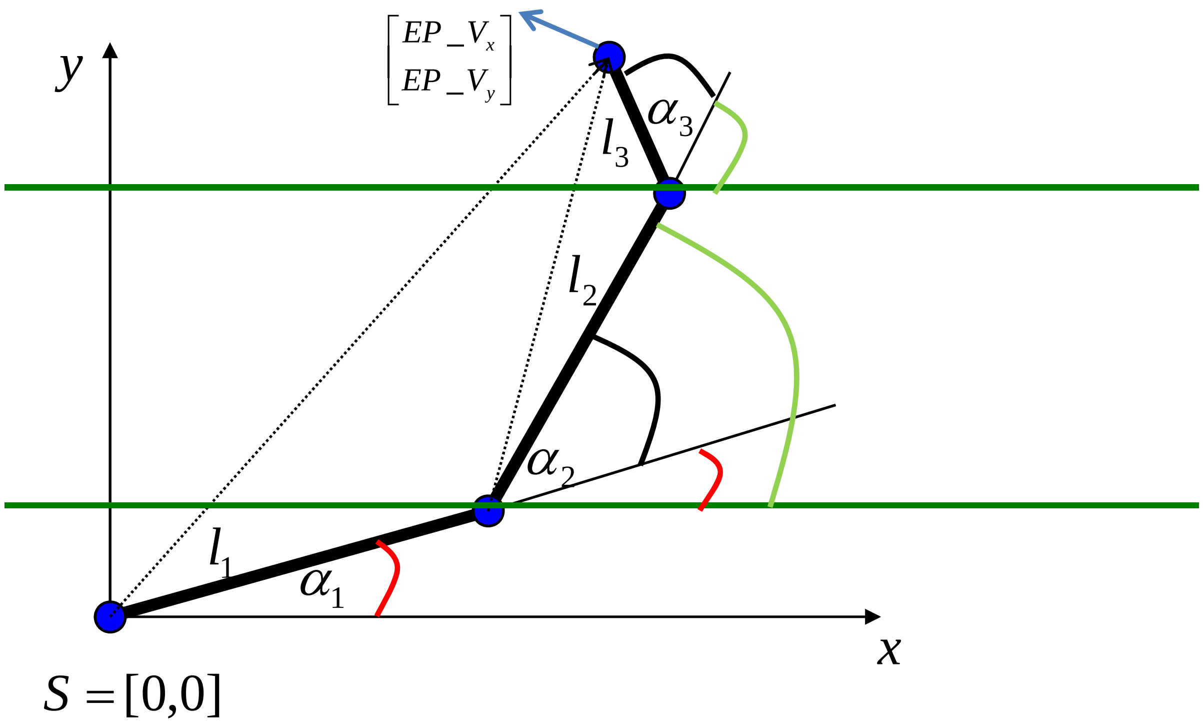
$l_1=1$
$l_2=1$
$l_3=1$
$\alpha 1=10^o$
$\alpha 2=110^o$
$\alpha 3=125^o$

Cos($\alpha$)=x
Sin($\alpha$) =y
e(x,y)

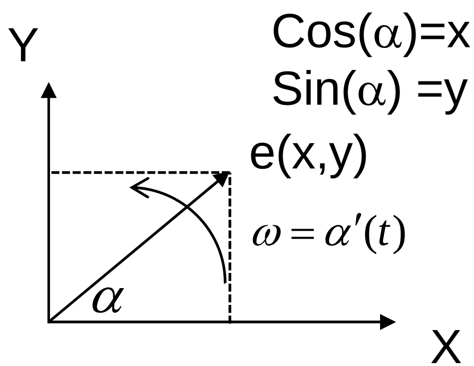


# Computation of the velocity vector in the endpoint

$$\begin{bmatrix}EP\_V_x\\ EP\_V_y\end{bmatrix}=\begin{bmatrix} -l_1s(\alpha_1)-l_2s(\alpha_1+\alpha_2)-l_3s(\alpha_1+\alpha_2+\alpha_3) & -l_2s(\alpha_1+\alpha_2)-l_3s(\alpha_1+\alpha_2+\alpha_3) & -l_3s(\alpha_1+\alpha_2+\alpha_3)\\ l_1c(\alpha_1)+l_2c(\alpha_1+\alpha_2)+l_3c(\alpha_1+\alpha_2+\alpha_3) & l_2c(\alpha_1+\alpha_2)+l_3c(\alpha_1+\alpha_2+\alpha_3) & l_3c(\alpha_1+\alpha_2+\alpha_3)\end{bmatrix}\begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\end{bmatrix}$$

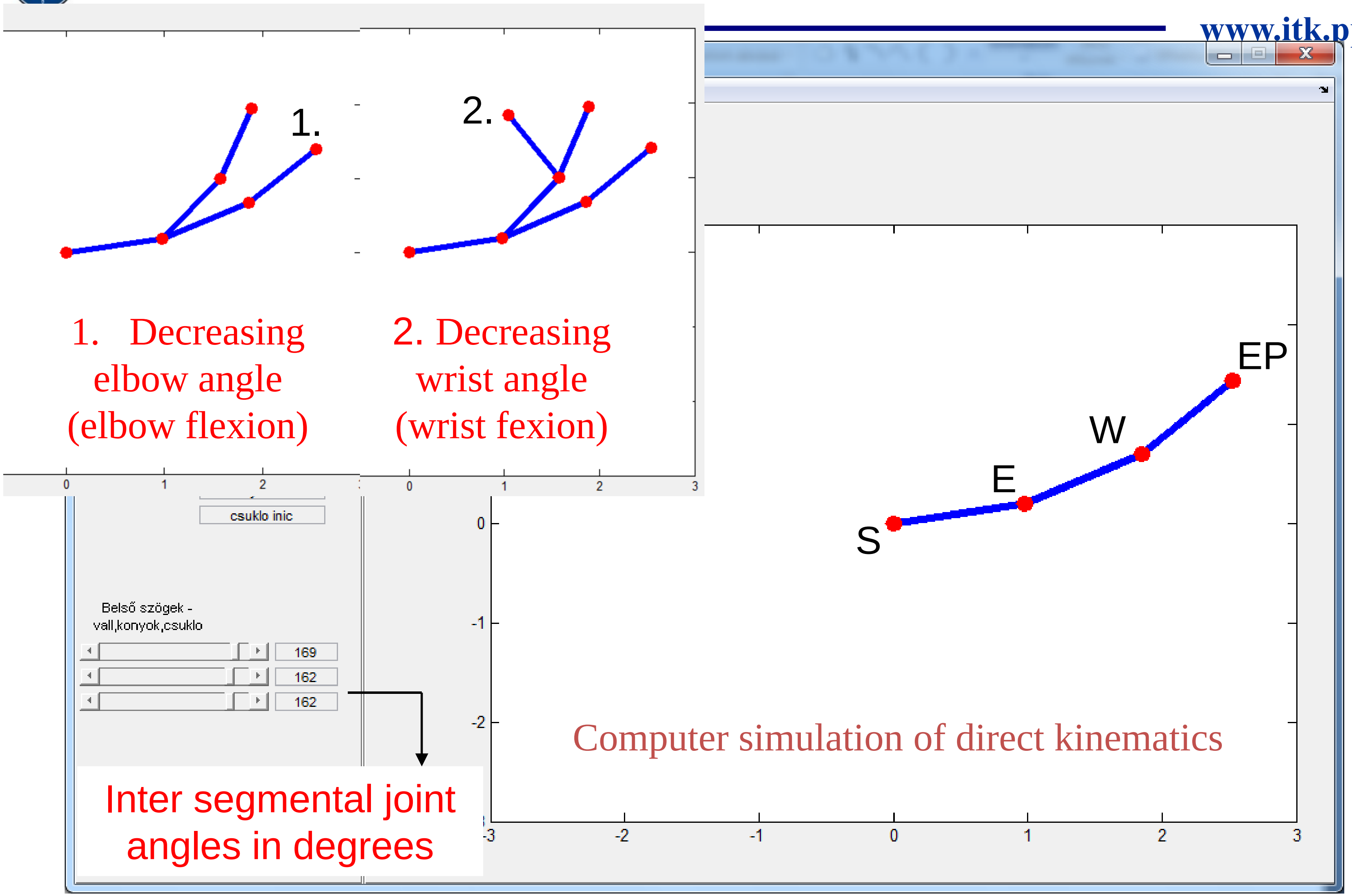



**s=sine c = cosine**

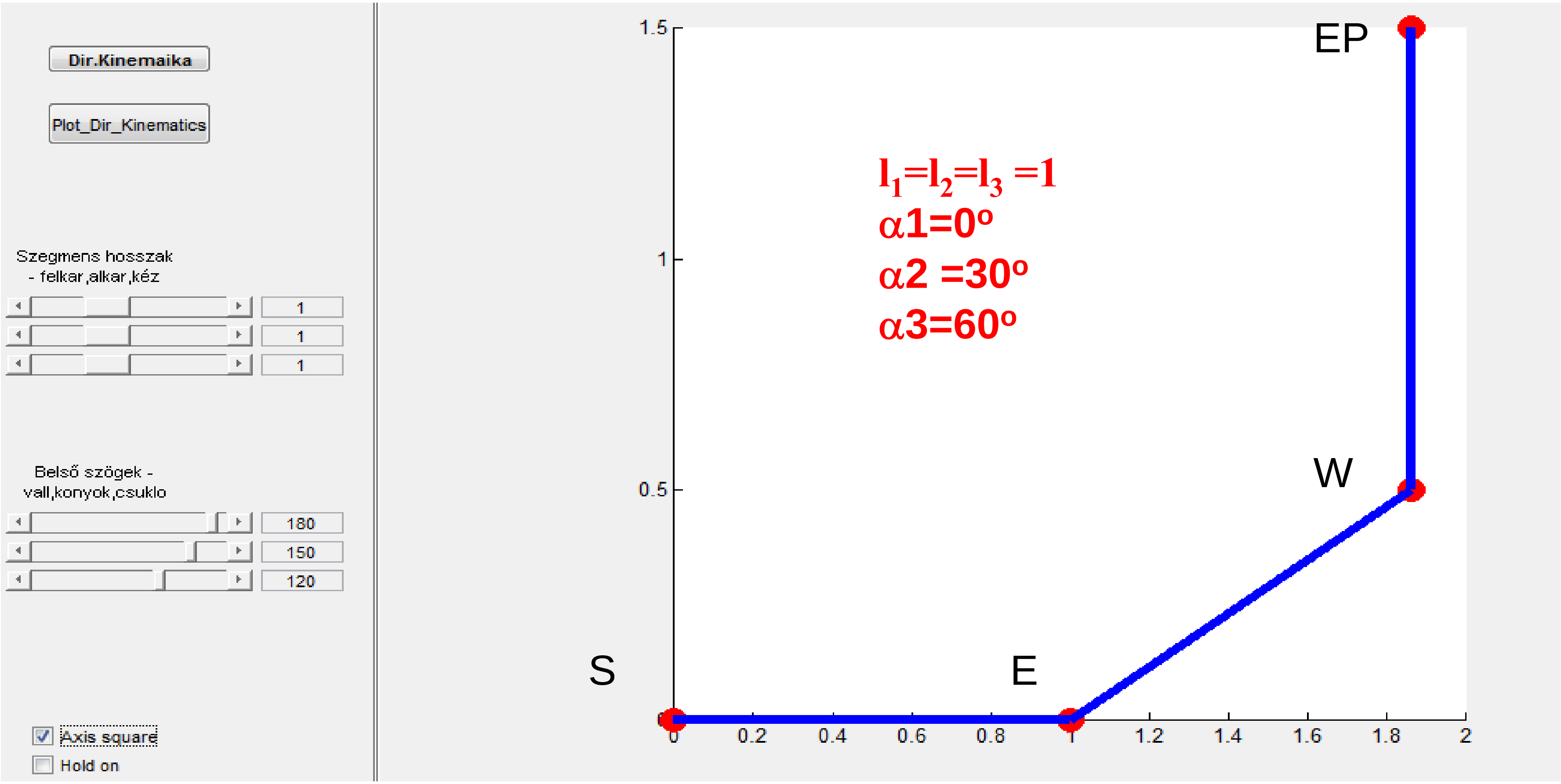


- **A MATLAB based program was developed to**
  - Solve the DK problem
  - Visualise the solution (**moving limb**)
    - Self-developed GUI(**graphical user interface**)
  - Present:
    1. Angular changes
    2. Joint coordinates of a 3 joint system in 2D space

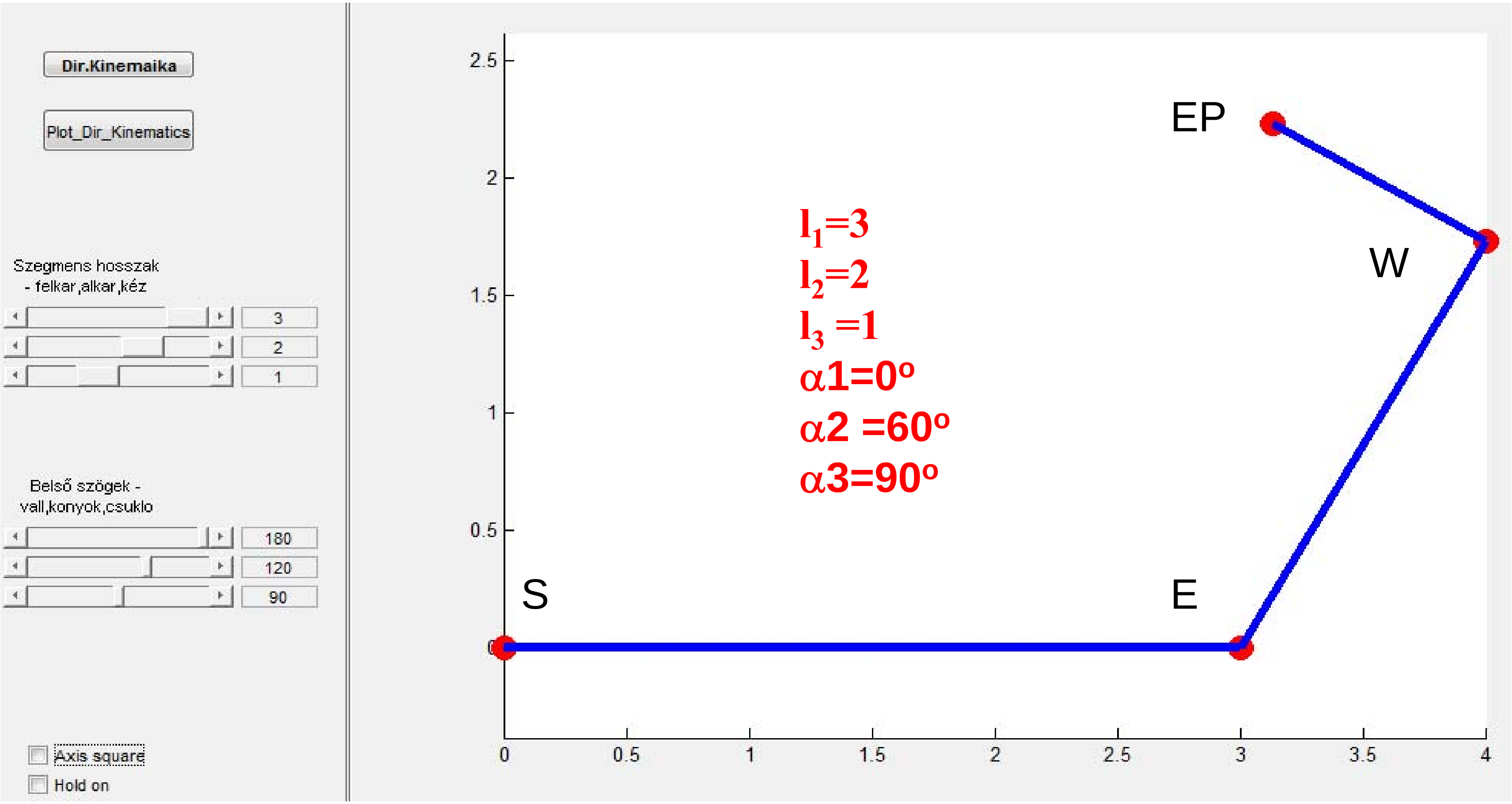
- **In the next slide:**
  - Upper limb (with fixed shoulder coordinates S=0,0)
  - Using the sliders one can define the joint angles ($0^o$-$180^o$):
    - Shoulder
    - Elbow
    - Wrist


1.
2.
1. Decreasing elbow angle (elbow flexion)
2. Decreasing wrist angle (wrist fexion)
csuklo inic
Belső szögek - vall,konyok,csuklo
169
162
162
Inter segmental joint angles in degrees
S
E
W
EP
Computer simulation of direct kinematics

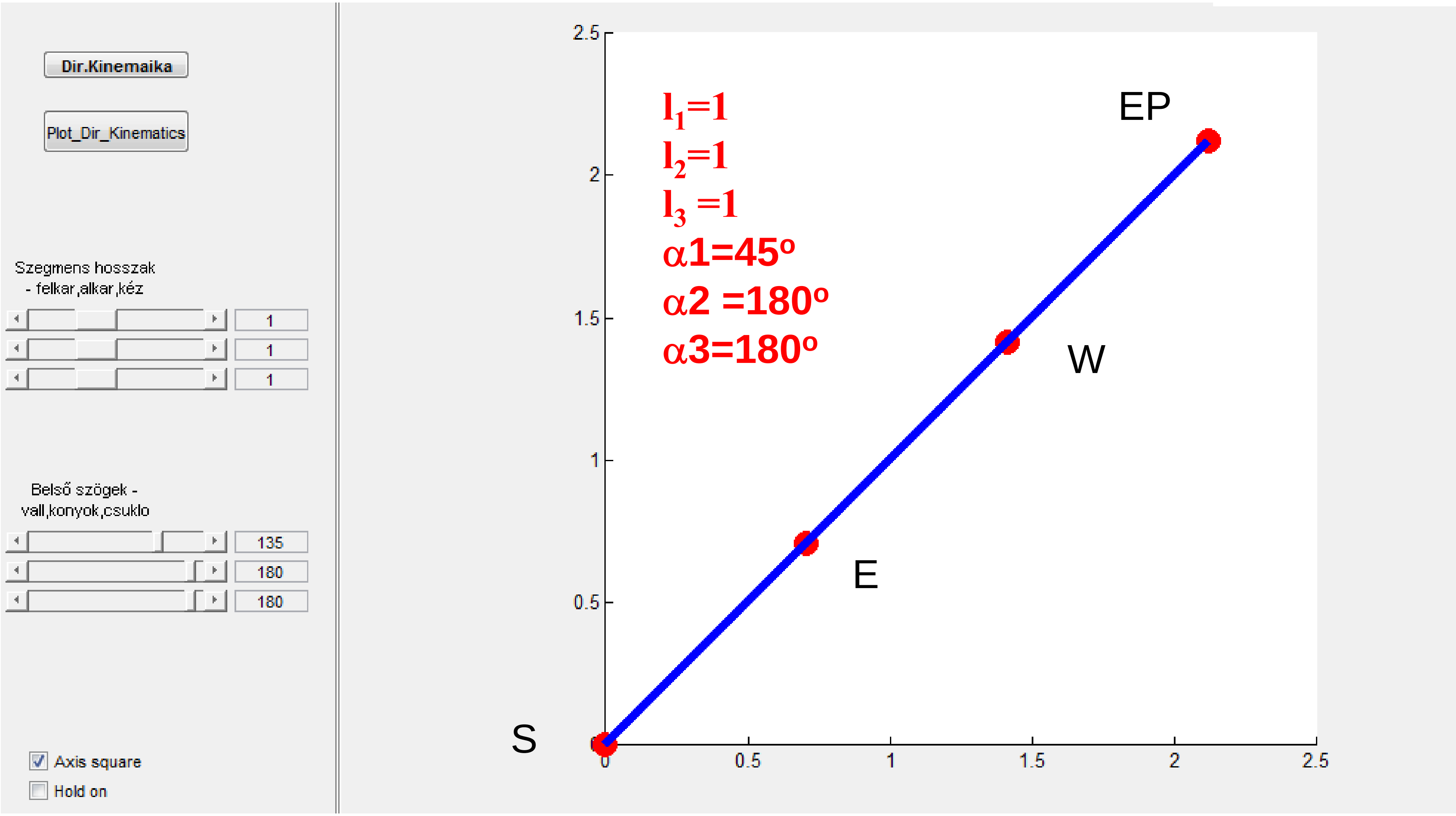
# Computation of Joint coordinates – An example with given joint angles (Simulation)



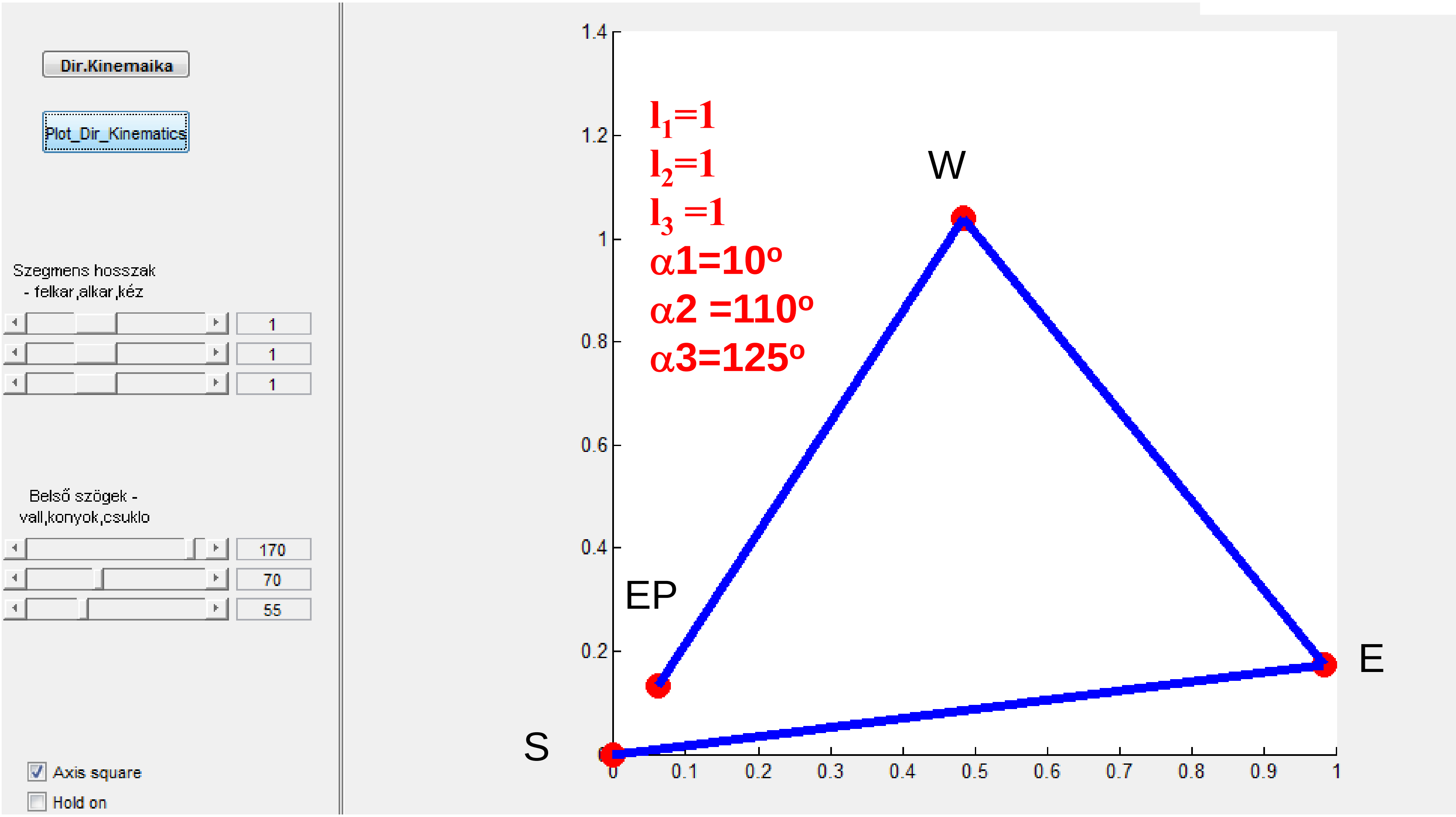
# Computation of Joint coordinates – An example with given joint angles (Simulation)



# Computation of Joint coordinates – An example for special posture



# Computation of Joint coordinates – An example for special posture



# Determination the angular acceleration (β(t))

- **The direction of the angular velocity ω(t)** is perpendicular to the plane of rotation (**PoR**) (right hand rule).
- **The perpendicular direction to PoR:**
  Can be computed at any instant as the ***cross product of adjacent limb segments.***
- **The direction of the angular acceleration β(t)** is also perpendicular to the plane of rotation.
  It is either points to the same direction as the angular velocity or to the opposite direction.

# Determination the angular acceleration (β)

- **The right side of Plane of Rotation (PoR):**

  the cross product of the unit vector pointing from the joint to the direction of the distal limb segment and the unit vector pointing from the joint to the direction of the proximal one.

- **If α(t) is decreasing (flexion):**
  - at **β(t)>0** than the angular acceleration vector (**AAV**) points to the left of the PoR. (the speed of the flexion is decreasing).
  - at **β(t)<0** the AAV points to the right PoR (the speed of the flexion is increasing)(flexion is associated with negative angular velocity and its speed is the absolute value of the angular velocity).

- **If $\alpha$(t) is increasing (extension):**
  - at **$\beta$(t)<0** angular acceleration vector points toward the right of Plane of Rotation
  - at **$\beta$(t)>0** angular acceleration vector points toward the left of Plane of Rotation.

**Question:** which muscles must be activated to generate a desired angular acceleration in the joint?

This depends on the instantaneous direction of the angular acceleration that aligns with the direction of the desired total torque.

The direction of the total torque depends both on on muscle forces and and on external forces (e.g. gravity).

$$\beta(t) = \frac{d^2\alpha(t)}{dt^2}$$ Magnitude of the angular acceleration

- **Direction:** perpendicular to the plane of rotation
- **Flexion** (inter segmental angle decreasing)

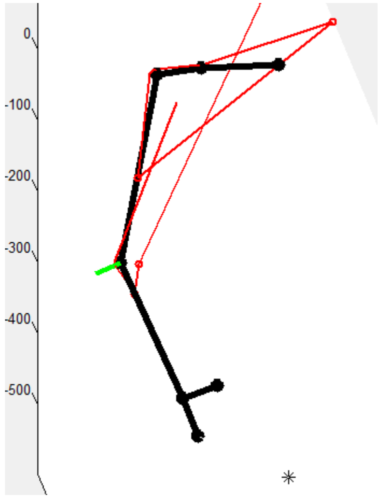
$\beta(t) > 0$ left

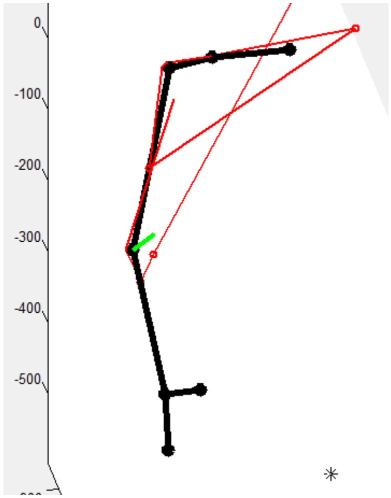
$\beta(t) < 0$ right

- **Extension** (inter segmental angle increasing)

$\beta(t) > 0$ right

$\beta(t) < 0$ left




Angular acceleration vector

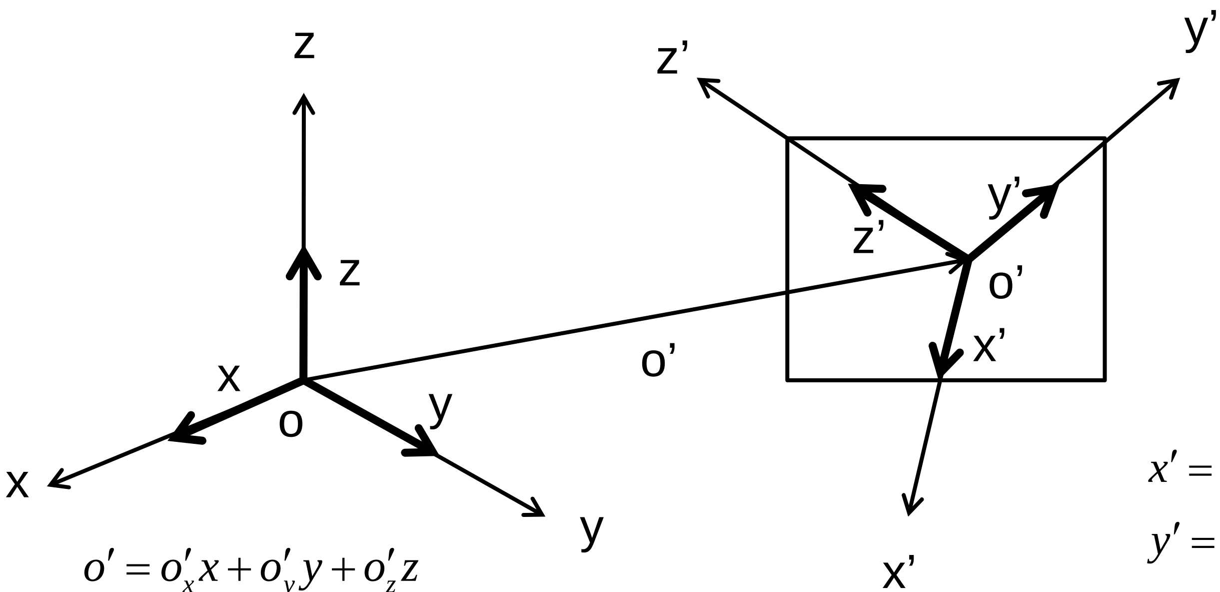
Muscle action lines

# Definition of limb from robotics point of view (PoV)

- **Limb,(manipulator):**
- consists of a series of rigid bodies (**links**) connected via kinematic pairs or joints
- Joints can essentially be **2 types** and are controlled via **actuators:**
  - revolute
  - prismatic
- The whole structure forms an **open kinematic chain** consisting of
  - fixed base
  - end effector
- Sometimes closed kinematic chain is applied as manipulator

# Position and orientation of a body in an arbitrary coordinate system

Reference coordinate system        Body coordinate system




$$o' = \begin{bmatrix} o'_x \\ o'_y \\ o'_z \end{bmatrix}$$ position

$$o' = o'_x x + o'_y y + o'_z z$$

$$x' = x'_x x + x'_y y + x'_z z$$
$$y' = y'_x x + y'_y y + y'_z z$$
$$z' = z'_x x + z'_y y + z'_z z$$
orientation

# Generalized rotation matrices (R) and their properties

$$R = \begin{bmatrix} x' & y' & z' \end{bmatrix} = \begin{bmatrix} x'_x & y'_x & z'_x \\ x'_y & y'_y & z'_y \\ x'_z & y'_z & z'_z \end{bmatrix}$$

**R** as the rotational matrix describes the 3D position of the body to the reference coordinate system.

## Rotation matrix features

Dimension: 3x3

Orthogonal: $R^T R = I$

If $\mathbf{R_1}$ and $\mathbf{R_2}$ are rotation matrices: $\mathbf{R=R_1R_2}$ where **R** is a rotational matrix

**If R is rotational matrix:** det(R)=1

# Elementary rotation matrices

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c(\alpha) & -s(\alpha) \\ 0 & s(\alpha) & c(\alpha) \end{bmatrix}$$

**Rotation about x axis by α**

$$R_y(\alpha) = \begin{bmatrix} c(\alpha) & 0 & s(\alpha) \\ 0 & 1 & 0 \\ -s(\alpha) & 0 & c(\alpha) \end{bmatrix}$$

**Rotation about y axis by α**

$$R_z(\alpha) = \begin{bmatrix} c(\alpha) & -s(\alpha) & 0 \\ s(\alpha) & c(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

**Rotation about z axis by α**

**Consecutive rotations:** $R = R_x(\alpha) R_y(\alpha) R_z(\alpha)$

# DK problem of an open kinematic chain

- **DK problem in robotics:** Describing the position and orientation of the end effector in world coordinates (the coordinate system of the base of the kinematic chain)
- **DK problem is given by the so called:** Denavit-Hartenberg (DH) algorithm

**DH convention:** gives all the parameters needed to describe a given kinematic chain with given joints

**DH algorithm:** solves DK problem

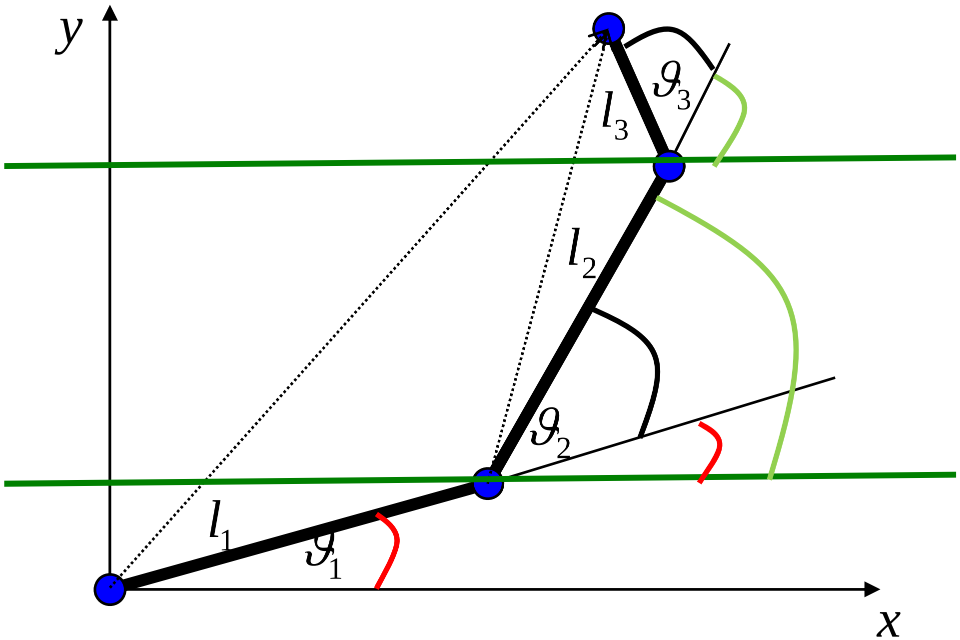
$$T_n^0(q) = A_1^0(q_1)A_2^1(q_2)A_n^{n-1}(q_n)$$

Where **0** is the base, **n** is the end effector, so the number of joints is **n, q** joint variables

- See DH convention and the algorithm for details:

http://www.cs.duke.edu/brd/Teaching/Bio/asmb/current/Papers/chap3-forward-kinematics.pdf

## DK problem of an open kinematic chain in 2D using DH algorithm




$S = [0,0]$

### DH parameters of the rotational joints

| Link | $l_i$ | $\alpha_i$ | $d_i$ | $\vartheta_i$ |
|---|---|---|---|---|
| 1 | $l_1$ | 0 | 0 | $\vartheta_1$ |
| 2 | $l_2$ | 0 | 0 | $\vartheta_2$ |
| 3 | $l_3$ | 0 | 0 | $\vartheta_3$ |

DH parameters depend on joint geometry

$$T_3^0(q) = A_1^0 A_2^1 A_3^2 = \begin{bmatrix} c_{123} & -s_{123} & 0 & l_1c_1 + l_2c_{12} + l_3c_{123} \\ s_{123} & c_{123} & 0 & l_1s_1 + l_2s_{12} + l_3s_{123} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$C_{12}$ =cos(1st joint angle+2nd joint angle)
$S_{12}$ =cos(1st joint angle+2nd joint angle)

# Summary

- **DK problem in biology:** the main aim is to describe a mathematical relation between the position of the endpoint of a limb and angular changes in the joints of the limb.
  - Matlab based GUI to give exact solutions of DK problem in 2D
- **DK problem in robotics:** the issue is to give the position and orientation of the endpoint of the given manipulator basically in the same coordinate system as the base of the robotic arm.
  - Rotation matrices
  - Denavit-Hartenberg algorithm

# Suggested literature

## Biology, Medicine and Sport Sciences


- Elliott B, Fleisig G, Nicholls R, Escamilia R (2003), Technique effects on upper limb loading in the tennis serve, J.Science and Med. in Sport 6(1),76-87
- Rab G, Petuskey K, Bagley A (2002), A method for determination of upper extremity kinematics, Gait and Posture 15(2) 113-119
- Analysis of human arm joints and extension of the study to robot manipulator: http://www.iaeng.org/publication/IMECS2009/IMECS2009_pp1348-1351.pdf
- McClure, P. W., L. A. Michener, et al. (2001). "Direct 3-dimensional measurement of scapular kinematics during dynamic movements in vivo." Journal of Shoulder and Elbow Surgery 10(3): 269-277.

# Suggested literature

## Robotics

•http://www.cs.duke.edu/brd/Teaching/Bio/asmb/current/Papers/chap3-forward-kinematics.pdf

•Etemadizanganeh, K. and J. Angeles (1995). "Real-Time Direct Kinematics of General 6-Degree-of-Freedom Parallel Manipulators with Minimum-Sensor Data." Journal of Robotic Systems 12(12): 833-844.

•Gosselin, C. M. and J. P. Merlet (1994). "The Direct Kinematics of Planar Parallel Manipulators - Special Architectures and Number of Solutions." Mechanism and Machine Theory 29(8): 1083-1097

•Kohli, D., S. H. Lee, et al. (1988). "Manipulator Configurations Based on Rotary-Linear (R-L) Actuators and Their Direct and Inverse Kinematics." Journal of Mechanisms Transmissions and Automation in Design-Transactions of the Asme 110(4): 397-404.