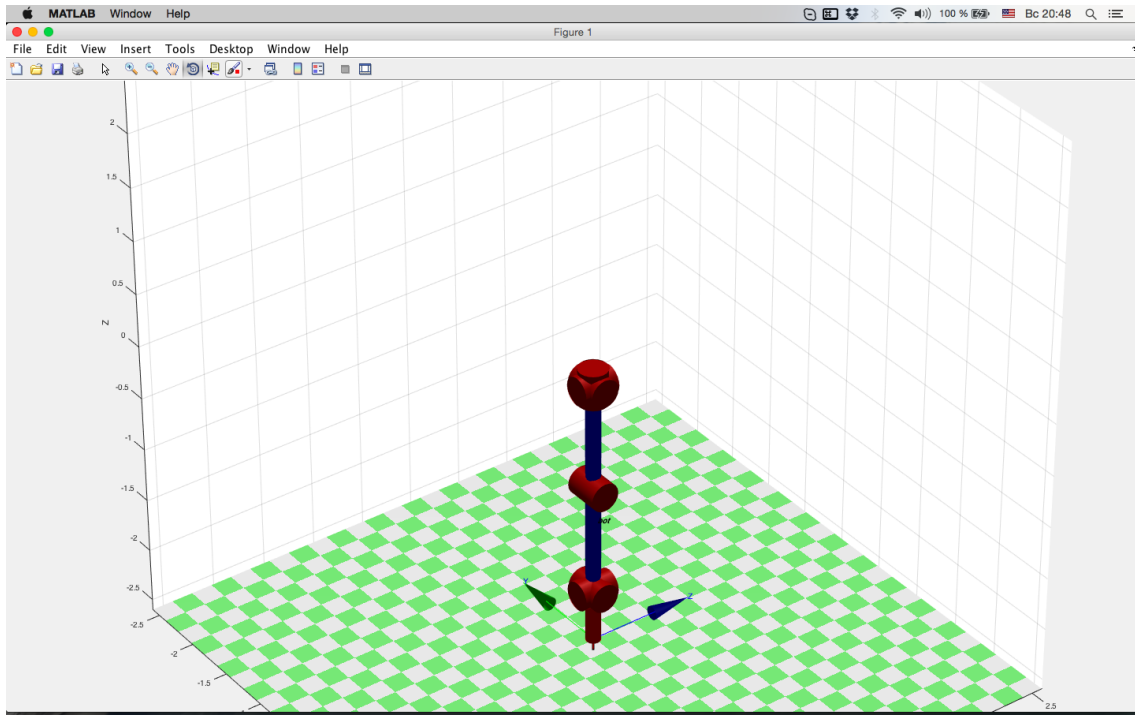


Final Project

It was decided to build simplified model of human leg. The problem was that non simplified leg has 7 DoF, but it is very hard to compute IK in this situation, so one redundant DoF was deleted.



DH:

	a_{i-1}	α_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	0	$\frac{\pi}{2}$	0	$\theta_2 - \frac{\pi}{2}$
3	0	$-\frac{\pi}{2}$	0	$\theta_3 + \pi$
4	L_1	0	0	θ_4
5	L_2	0	0	$\theta_5 + \frac{\pi}{2}$
6	0	$-\frac{\pi}{2}$	0	θ_6
7	L_3	0	0	0

Transformation matrices

$$\begin{aligned}
 {}^0_1T &= \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 {}^1_2T &= \begin{bmatrix} \cos(\theta_2 - \frac{\pi}{2}) & -\sin(\theta_2 - \frac{\pi}{2}) & 0 & 0 \\ 0 & 0 & -1 & 0 \\ \sin(\theta_2 - \frac{\pi}{2}) & \cos(\theta_2 - \frac{\pi}{2}) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 {}^2_3T &= \begin{bmatrix} \cos(\theta_3 + \pi) & -\sin(\theta_3 + \pi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\sin(\theta_3 + \pi) & -\cos(\theta_3 + \pi) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 {}^3_4T &= \begin{bmatrix} \cos(\theta_4) & -\sin(\theta_4) & 0 & L_1 \\ \sin(\theta_4) & \cos(\theta_4) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 {}^4_5T &= \begin{bmatrix} \cos(\theta_5 + \frac{\pi}{2}) & -\sin(\theta_5 + \frac{\pi}{2}) & 0 & L_2 \\ \sin(\theta_5 + \frac{\pi}{2}) & \cos(\theta_5 + \frac{\pi}{2}) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 {}^5_6T &= \begin{bmatrix} \cos(\theta_6) & -\sin(\theta_6) & 0 & L_1 \\ 0 & 0 & 1 & 0 \\ -\sin(\theta_6) & -\cos(\theta_6) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 {}^6_{EE}T &= \begin{bmatrix} 1 & 0 & 0 & L_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned}$$

$${}^0_{EE}T = {}^0_1T {}^1_2T {}^2_3T {}^3_4T {}^4_5T {}^5_6T {}^6_{EE}T$$

First of all when we talk about IK problem, we should understand that we have not only position but also the orientation of the EE. For simplicity in this work as orientation will be used pitch roll and yaw.

Then we can calculate the coordinates of the heel:

$$\begin{cases} H_x = x - L_3 \sin(yaw) \cos(pitch) \\ H_y = y - L_3 \cos(yaw) \cos(pitch) \\ H_z = z - L_3 \sin(pitch) \end{cases}$$

We can see that our simplification - deleting one DoF at foot helped us because roll will now definitely cause the place of the knee due to the fact that this DoF can be guaranteed only by the hip.

Now we have enough information to compute angle θ_4

$$\theta_4 = \arccos \left(\frac{(H_x^2 + H_y^2 + H_z^2) - L_1 - L_2}{2L_1L_2} \right)$$

Now we can compute the point of intersection the knee and hip-heel. We we call this point Under Knee.

UK:

First of all we can find η that is the coefficient that show how the point of UK derives the line from hip to the heel.

And after that it is obvious that $UK = (\eta H_x, \eta H_y, \eta H_z)$

After that we can find angles α and β from the following:

$$\begin{cases} UK_x = L'_1 \sin(\alpha) \cos(\beta) \\ UK_y = L'_1 \cos(\alpha) \cos(\beta) \\ UK_z = L'_1 \sin(\beta) \end{cases}$$

And so we are moving forward to compute the coordinates of the knee. We can calculate K_0 that shows coordinates of the knee when roll is equal to zero.

$$\begin{cases} K0_x = h \sin(\alpha) \sin(\beta) + UK_x \\ K0_y = h \sin(\beta) \sin(\alpha) + UK_y \\ K0_z = h \sin(\beta) + UK_z \end{cases}$$

Where h is the distance between the knee and UK point.

And so we have find $\theta_1 = roll$

$\theta_2 = \alpha$

$\theta_3 = \beta$

To find other two angles we will find the rotation around axes hip - heel and then after multiplying this rotation matrix by coordinates of vector K_0 we will find coordinates of point K - the knee.

θ_5 can be easily find from cosine theorem. And it is equal to $\arccos \left(\frac{L_2^2 + L_3^2 - LengthOfShin^2}{2L_2L_3} \right)$

After that it is not a big deal to find θ_6 as $yaw + \nu$, where ν can be fined from

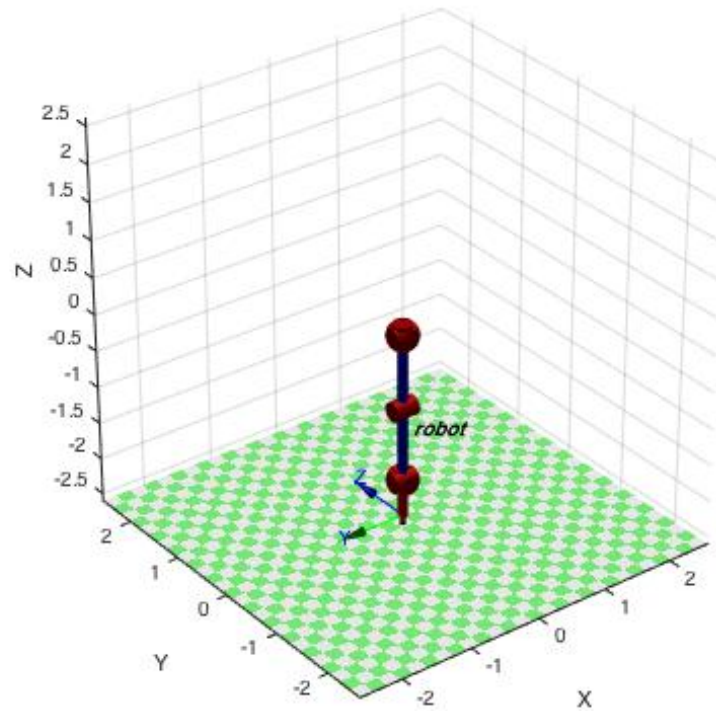
$$\nu = \frac{K_x - H_x}{\sin(\alpha) \sqrt{K_x^2 + K_y^2 + K_z^2}}$$

And now all the IK is solved.

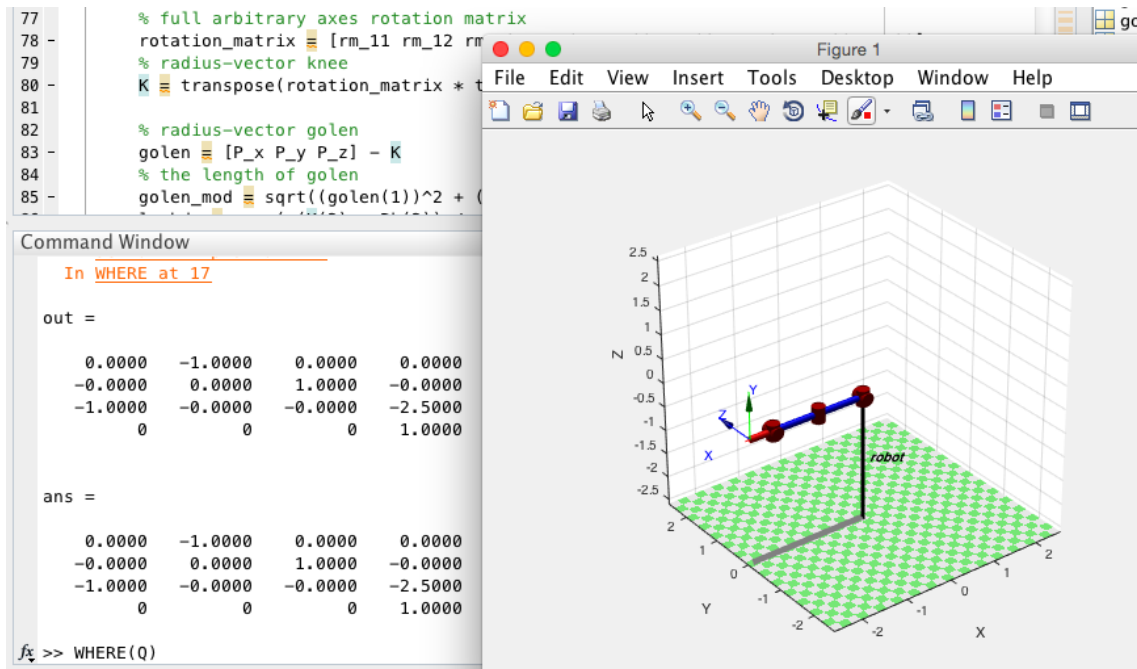
Appendix 1

Implemented commands.

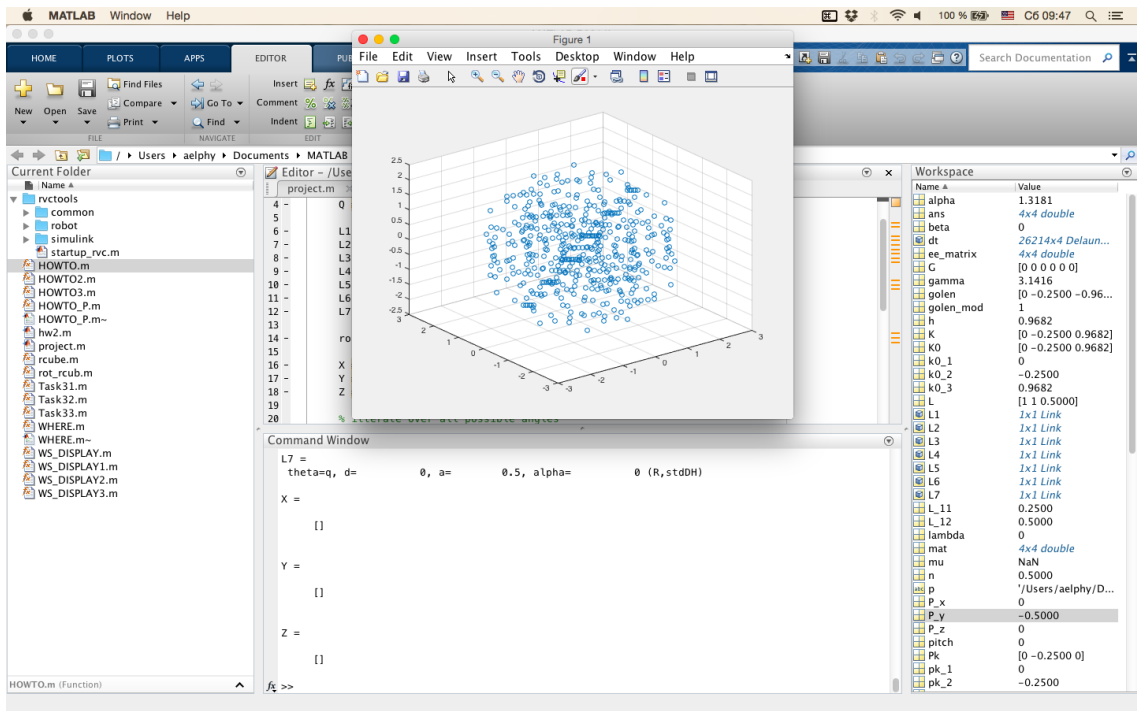
project - draw the robot, returns nothing



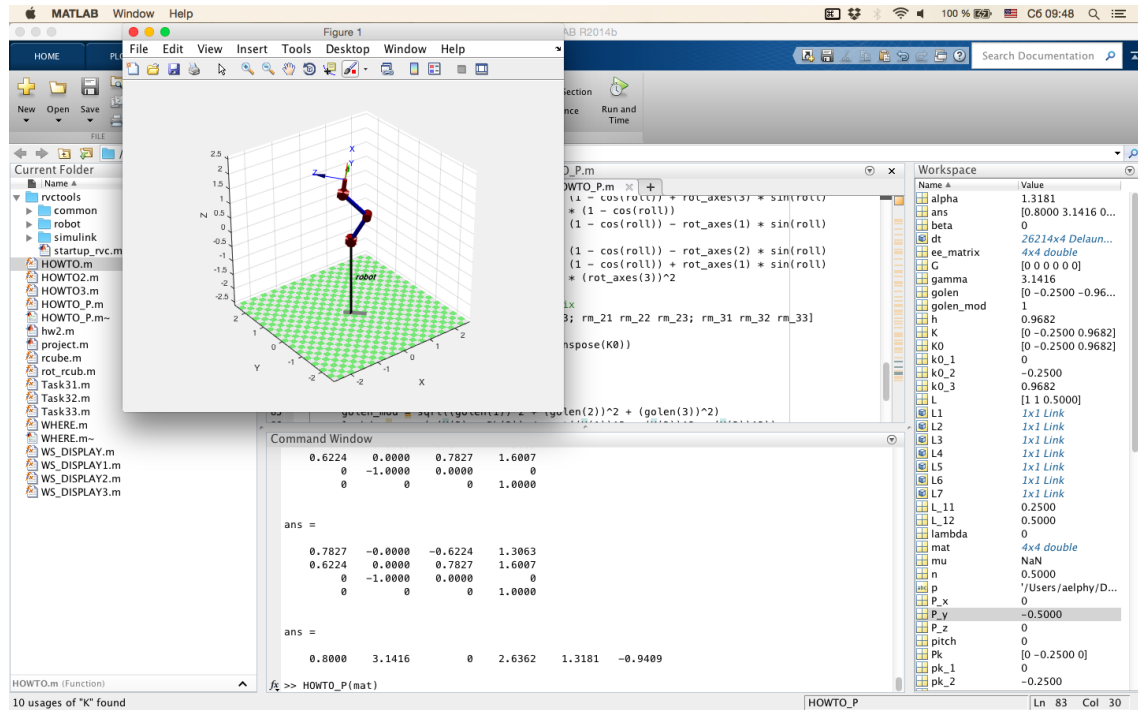
WHERE - returns the coordinates of EE and plots the configuration of the robot.
Returns matrix 4*4 with position and orientation of EE.



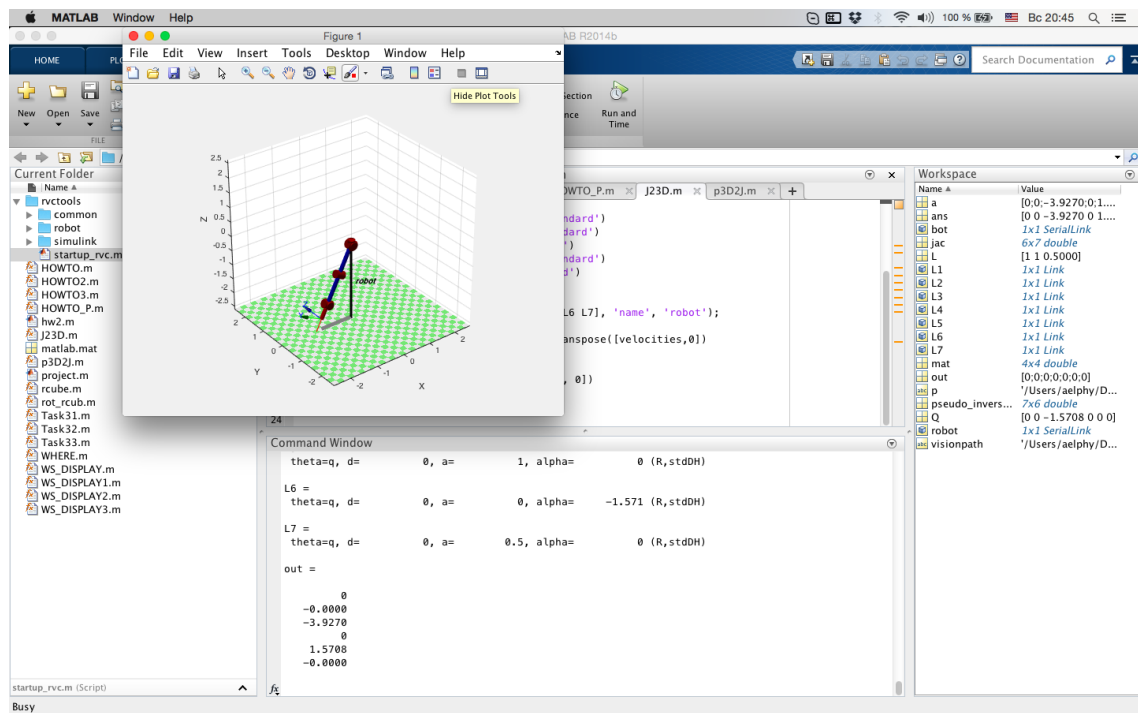
WS_DISPLAY - calculates reachable work space. Works long time. returns nothing.



HOWTO_P - as input 4*4 matrix of position and orientation is needed. Solves the IK. Returns the 1*6 array of joint angles.



J23D - as input joint velocities vector 1*6.



Appendix 2

Commands:

project:

input - NULL, output - NULL

WHERE(Q):

input - [0 0 -1.5708 0 0 0]

output -

$$\begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & -2.5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

WS_DISPLAY: input - NULL, output - NULL

HOWTO_P(ee_matrix)

input:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.6967 & -0.7174 & 0 \\ 0 & 0.7174 & 0.6967 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

output:

$$\begin{bmatrix} 0.7827 & 0 & -0.6224 & 1.3063 \\ 0.6224 & 0 & 0.7827 & 1.6007 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

J23D(velocities)

input: [0 0 -1.5708 0 0 0]

output: [0 0 -3.9270 0 1.5708 0]

p3D2J(v_velocities, w_velocities)

input: [1,1,1], [1,1,1]

output: [0.0312 0.0312 0.7500 0.0312 0.3125 0.5938 1.7500]