# Direct and Inverse Kinematics for the NU-Biped using Screw Theory

Togzhan Syrymova Ansar Serik

**Goal** - is to formalize the direct and inverse kinematics of the NU-Biped's single leg using the screw displacements method. The NU biped leg has a total of 6 DOF, i.e. a RRRRRR kinematic architecture.

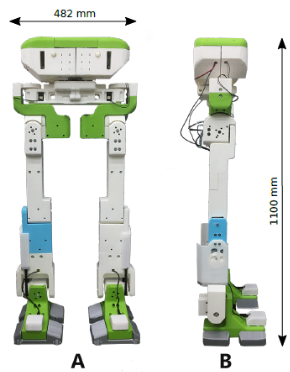



Fig. 1. NU-Biped legs

## I. Introduction

As technology develops with rapid speed the demand for robots increases too. The scientists and engineers tries to create robust, efficient manipulators with different construction for specific tasks. The kinematics for each manipulator differs according to joints and links combination. To achieve kinematics of the robots Direct and Inverse kinematics must be solved. In direct kinematics (DK) all configuration of the joints are given, the goal is to find transformation matrix that gives final point position and orientation. There exist different methods to find DK: using Denavit - Hartenberg convention, Craig convention, successive screw method. In this project the last method will be used [1]. Inverse kinematics (IK) is vice-verse. The final position and orientation of the end effector of the manipulator is given, while goal of IK is to find all possible joint parameters (angle for revolute, and distance for prismatic). There did not exist universal solution to find IK, but only suggestions and advises. Hence IK is more complicated to find than DK. Moreover, transformation matrix of the DK helps to find IK. Therefore, In this project firstly the DK will be calculated then IK. As it was said earlier, one of the common way to find Direct and inverse Kinematics is successive screw displacement method. First of all, the screw axis representation can be seen in fig 2. Main idea behind this theory is to rotate point $P_[1]$ around point $S_p$ to angle

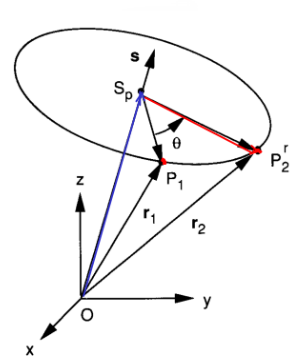



Fig. 2. Screw axis method representation

$\theta$ to obtain final point $P_2$ [2]. This theory stands for two points. The expansion screw method for two and more links in fig. 3. Hence, for each transformation, it is needed to create screw axis. Furthermore, the unit vector $S_i$ and its location $S_{oi}$ needs to be found. The next point, Tsai provides the formula of transformation matrix of one frame which is provided in appendix 1. [2].

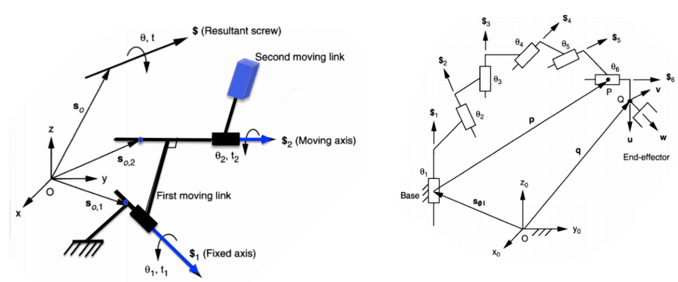



Fig. 3. screw axis representation for multiple links

Finding Inverse kinematics using successive screw displacement method is much complicated and individual for each serial manipulator according to its Direct kinematics transformation matrix and construction of the manipulator.

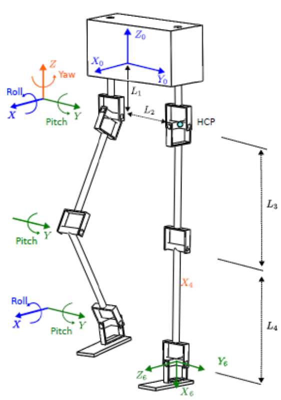
All calculations will be used to find NU-Biped robot, which was created at the Nazarbayev University by robotics department as in fig 1 and fig 4. The aim is to find kinematics of one leg. The initial frame and position is shown in frame 0. The point HCP connecting first 2 links and it rotates consequently in Z-axis, Y-axis and X-axis as shown in fig.4. Next joint located and knee L3 mm negative at $Z_0$ direction.

finally end effector locates at toe, frame 6. and it has two rotation, y-axis and x-axis consequently.

The paper is organized as follows.Section II provides with solution of Direct kinematics for NU-Biped. Section II is divided by subsection II-A where screw method was implemented to MatLab program, then subsection II-B shows simple examples of final position and end effector. Next section III stands for calculating and validating the results, subsections III-A and III-B respectively.

## II. Final Displacement using the Screw Theory

NU_Biped legs has a HCP point as shown in the fig4. We will be using this point for further calculations of the direct and inverse kinematics. The HCP point has 3 revolute (yaw, pitch, and roll) joints. Likewise, NU-Biped leg has one revolute joint at the knee (pitch), and two revolute (pitch and roll) joints at the foot. Each joint can be considered as a separate frame. The joints order is Yaw, Pitch, and Roll. As a result, we have 6 frames starting with frame 1 - yaw revolute joint at point HCP - and ending with frame 6 - roll a revolute joint at the foot. However, as we can see from the fig4 the frame 1 is shifted from reference frame 0 by L1 on $Z_0$ axis and L2 on $Y_0$ axis. We took that into consideration when we were calculating the transformation matrices $A_i$.




| Lengths | Value (mm) |
|---|---|
| $L_1$ | 200 |
| $L_2$ | 154.75 |
| $L_3$ | 318.50 |
| $L_4$ | 318.50 |

Fig. 4. NU-Biped legs kinematics

### A. Matlab program

We have implemented the function that calculates and return the transformation matrix for the given inputs based on the screw displacement theory. We have considered Zeros frame as shown in the fig4 and direction of the screws from 1 to 6 coincident to the shown coordinates. Therefore, we have obtained initial position and orientation, $P_o = [0; L2; -L1 - L3 - L4; 1]$, $u_0 = []$, $v_0 = []$, $w_0 = []$ and the table.I for the screw displacement theory: "make_matrix" function takes s_x, s_y, s_z,s0_x, s0_y, s0_z values obtained from applying the screw displacement method and returns the $4x4$ transformation matrix (fig 6).

TABLE I
Screw displacement values for the direct kinematics

| | $S_i$ | $S_{0i}$ |
|---|---|---|
| 1 | (0, 0, 1) | (0, L2, -L1) |
| 2 | (0, 1, 0) | (0, L2, -L1) |
| 3 | (1, 0, 0) | (0, L2, -L1) |
| 4 | (0, 1, 0) | (0, L2, -L1 - L3) |
| 5 | (0, 1, 0) | (0, L2, -L1 - L3 - L4) |
| 6 | (1, 0, 0) | (0, L2, -L1 - L3 - L4) |

$$A_i = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

We have obtained the transformation matrices for all 6 frames and then obtained the resultant transformation matrix. Using that resultant matrix $A_6^0$ and eq1we obtained the location of the end-effector (feet).

$$A_6^0 = A_1 * A_2 * A_3 * A_4 * A_5 * A_6 P = A_6^0 * P_0 \quad (1)$$

```
function A = make_matrix(s_x, s_y, s_z, theta, s_ox, s_oy, s_oz)
a_11 = (s_x^2 - 1)*(1 - cos(theta)) + 1;
a_12 = s_x*s_y*(1 - cos(theta)) - s_z*sin(theta);
a_13 = s_x*s_z*(1 - cos(theta)) + s_y*sin(theta);
a_21 = s_y*s_x*(1 - cos(theta)) + s_z*sin(theta);
a_22 = (s_y^2 - 1)*(1 - cos(theta)) + 1;
a_23 = s_y*s_z*(1 - cos(theta)) - s_x*sin(theta);
a_31 = s_z*s_x*(1 - cos(theta)) - s_y*sin(theta);
a_32 = s_z*s_y*(1 - cos(theta)) + s_x*sin(theta);
a_33 = (s_z^2 - 1)*(1 - cos(theta)) + 1;
a_14 = -s_ox*(a_11 - 1) - s_oy*a_12 - s_oz*a_13;
a_24 = -s_ox*a_21 - s_oy*(a_22 - 1) - s_oz*a_23;
a_34 = -s_ox*a_31 - s_oy*a_32 - s_oz*(a_33 - 1);

A = [a_11 a_12 a_13 a_14;
     a_21 a_22 a_23 a_24;
     a_31 a_32 a_33 a_34;
     0    0    0    1];

end
```

Fig. 5. Matrix function

We have combined previous steps and created the MATLAB function that takes as input the angles and gives the resultant position vector.

```
function P = forward_solve(theta1, theta2, theta3, theta4, theta5, theta6, L1, L2, L3, L4)
P_0 = [0; L2; -L1 - L3 - L4; 1];

A_01 = make_matrix(0, 0, 1, theta1, 0, L2, -L1);
A_12 = make_matrix(0, 1, 0, theta2, 0, L2, -L1);
A_23 = make_matrix(1, 0, 0, theta3, 0, L2, -L1);

A_34 = make_matrix(0, 1, 0, theta4, 0, L2, -L1 - L3);

A_45 = make_matrix(0, 1, 0, theta5, 0, L2, -L1 - L3 - L4);
A_56 = make_matrix(1, 0, 0, theta6, 0, L2, -L1 - L3 - L4);


A_06 = A_01*A_12*A_23*A_34*A_45*A_56;

P = A_06*P_0;
end
```

Fig. 6. Matrix function

### B. *Validation and visualisation*

We have implemented the visualization of the NU-Biped legs' links to show that our model of direct kinematics works correctly. So on the fig7 you can see the leg in the "home position". In other words, all links are in the stretched position.

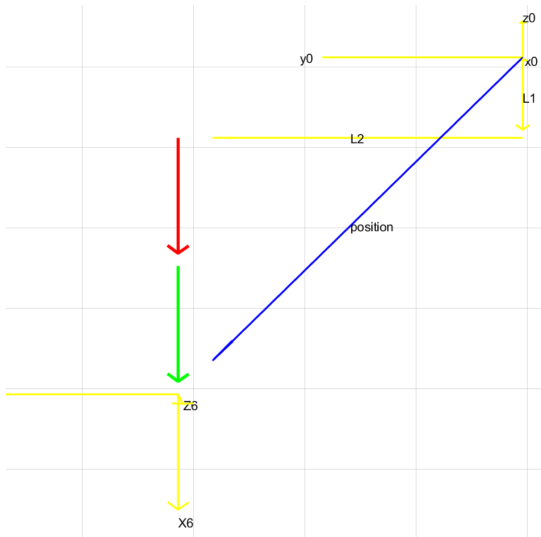



Fig. 7. Model of the leg in the stretched position

On the fig8 you can see the model when the leg is bent on the knee by 90 degrees.

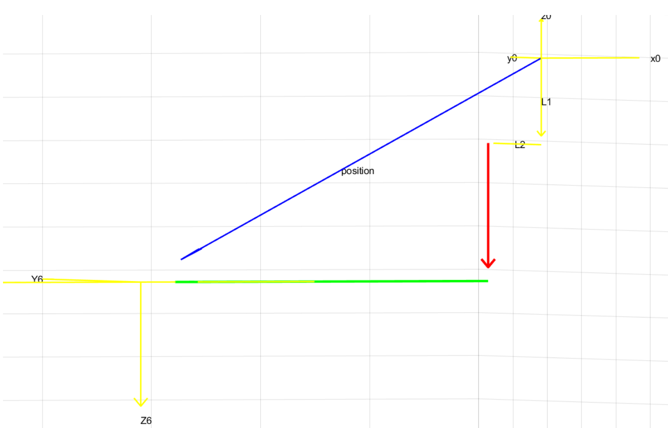



Fig. 8. Model of the leg bent in knee

## III. Inverse Kinematics using the Screw Theory

So for solving the Inverse Kinematics using the approach as in the Direct Kinematics (see SectionII) is not efficient. Hence the manipulator will be considered from the toe to the body. In other word initial position is final position and final position is initial, final position will be considered as HCP point. This was done with the purpose, that point $O$ and point HCP has constant coordinate difference, hence it will not affect to joint results. At the end the results will be transferred from new frame to original frame. ORNOT

TABLE II
Screw displacement values for the inverse kinematics

| | $S_i$ | $S_{0i}$ |
|---|---|---|
| 6 | (0, 0, 1) | (0, 0, 0) |
| 5 | (0, 1, 0) | (0, 0, 0) |
| 4 | (0, 1, 0) | (-L4, 0, 0) |
| 3 | (0, 1, 0) | (-L4-L3, 0, 0) |
| 2 | (0, 1, 0) | (-L4-L3, 0, 0) |
| 1 | (1, 0, 0) | (-L4-L3, 0, 0) |

Using similar steps as in section II, the table II and initial data was derived from kinematics as in fig.9. For position, $P_0 = [-L1 - L3 - L4, -L2, 0, 1]$, and for orientation: $u_0 = [], v_0 = [], w_0 = []$ .

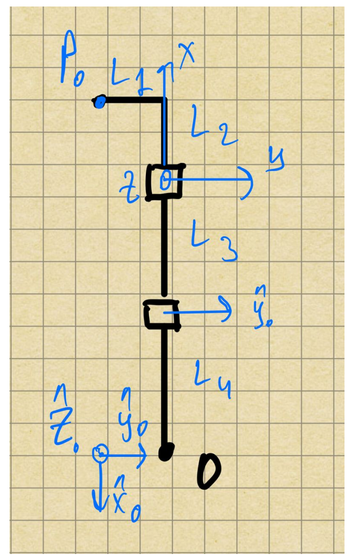



Fig. 9. Screw method for new frame

### A. *Manual Calculation*

The initial problem states that frame 1 is located on the point HCP and the position of the end-effector is dependent on the angles $\theta_1$, $\theta_2$, $\theta_3$ and $\theta_4$, However, it is a challenging problem if we are starting to compute the inverse kinematics from frame 0. Thus, we have considered going in the reverse direction. Frame 6 is the starting point and frame 1 at the HCP is the end-effector. Consequently, the position of the end-effector now is dependent on the angles $\theta_4$, $\theta_5$, and $\theta_6$. Likewise, $\theta_3, \theta_2 and \theta_1$ will affect only on the orientation. To obtain the angles $\theta_4$, $\theta_5$, and $\theta_6$ we have used the equation 2.

$$A_4^6 = A_4 * A_5 * A_6$$

$$P = A_4^6 * P_0 \tag{2}$$

Consequently, we are obtaining following equations: eq3, where $k_1 = p_x * cos(\theta_6) + p_y * sin(\theta_6)$.

$$k_1 = L_4 * (cos(\theta_4 + \theta_5) - cos(\theta_5)) - cos(\theta_4 + \theta_5) * (L_3 + L_4) \tag{3}$$

$$p_y * cos(\theta_6) - p_x * sin(\theta_6) = 0 \tag{4}$$

$$p_z = sin(\theta_4+\theta_5)*(L_3+L_4) - L_4*(sin(\theta_4+\theta_5) - sin(\theta_5)) \quad (5)$$

*1) $\theta_6$:* From eq3 we can find the angle $\theta_6$. $tan(\theta_6) = p_y/p_x$ will give us the angle $\theta_6$.

$$\theta_6 = tan^-1(\frac{p_y}{p_x})$$

$$\theta_6^* = \pi + \theta_6$$

*2) $\theta_4$:* By squaring and adding two equations (eq3 and eq5 we can find the angle $\theta_4$. You can find the detailed solution on the figure 10.

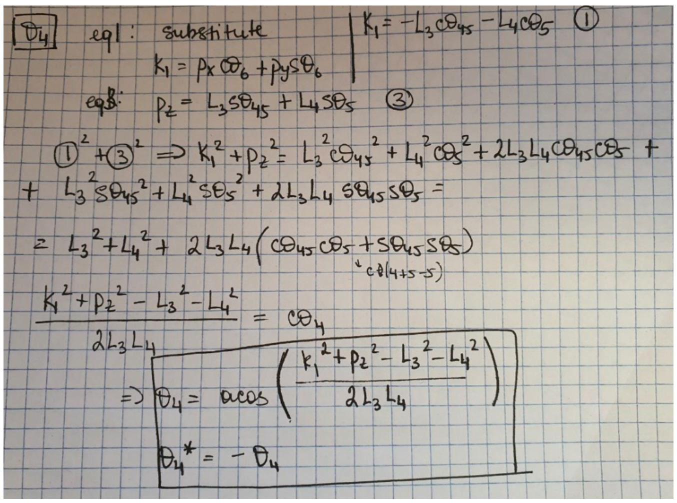



Fig. 10. Solution for the $\theta_4$

*3) $\theta_5$:* To find the angle $\theta_5$ we have used the equations 3 and 5. From equation 3 we can find $cos(\theta_5)$ that has variable $sin(\theta_5)$ in it. Then we can substitute the $cos(\theta_5)$ into the equation 5 and find $sin(\theta_5)$. And after that we obtain the value of $cos(\theta_5)$. You can find the detailed solution in the figure 11 and figure 12.

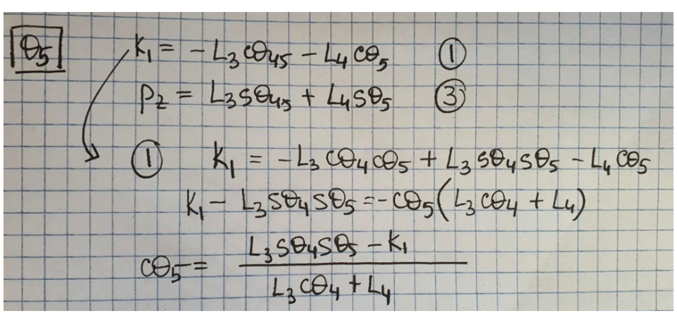



Fig. 11. Solution for the $\theta_5$: part 1

*4) $\theta 3$:* Since the $\theta_3, \theta_2 and \theta_1$ define the orientation only, we can find their values using the rotation matrices and the equation 6.

$$R_4^T * R_5^T * R_6^T * w = R_3 * R_2 * R_1 * w_0 \quad (6)$$

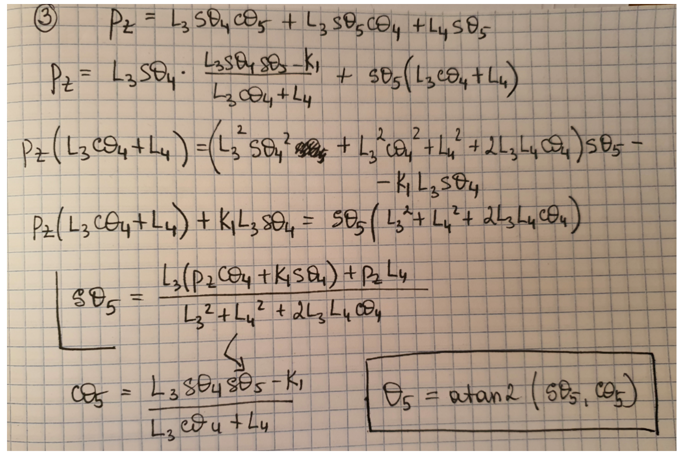



Fig. 12. Solution for the $\theta_5$: part 2

Left side of the equation ($R_4^T * R_5^T * R_6^T * w$) is known thus we can substitute it with $k_2, k3andk_4$. Likewise, we are left with the following equations:

$$k_2 = -cos(\theta_2) * cos(\theta_3) \quad (7)$$

$$k_3 = -cos(\theta_2) * sin(\theta_3) \quad (8)$$

$$k_4 = sin(\theta_2) \quad (9)$$

From equations 7 and 8 we have $cos(\theta_2) = -\frac{k_2}{cos(\theta_3)}$ and $cos(\theta_2) = -\frac{k_3}{sin(\theta_3)}$. From this we can obtain angle $\theta_3$.

$$\theta_3 = tan^{-1}(\frac{k_3}{k_2})$$

$$\theta_3^* = \theta_3$$

You can find the full solution in the figure 13.

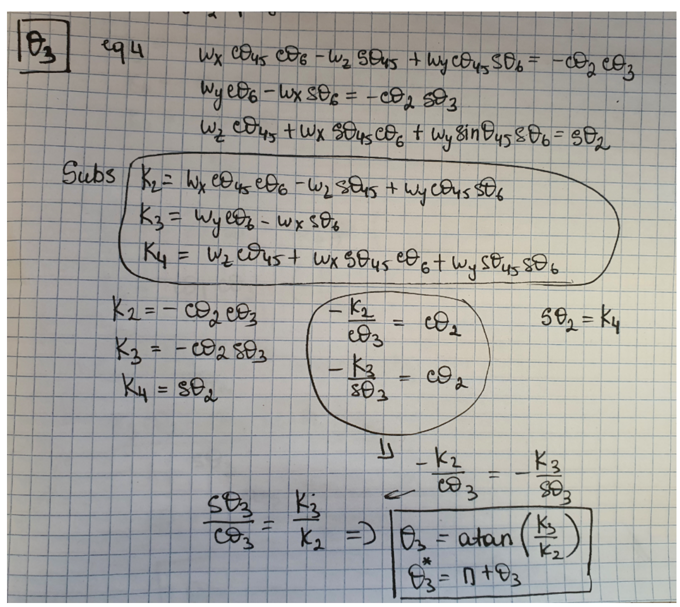



Fig. 13. Solution for the $\theta_3$

*5) θ2:* We know $cos(\theta_2) = -\frac{k_2}{cos(\theta_3)}$ and $sin(\theta_2) = k_4$.

$$\theta_2^* = atan2(k_4, \frac{-k_2}{cos(\theta_3)})$$

*6) θ1:* From equation 10 we have $sin(\theta_1) = \frac{-k_7}{cos(\theta_2)}$ and $k_5 = -cos(\theta_1) * sin(\theta_3) - cos(\theta_3) * sin(\theta_1) * sin(\theta_2)$.

$$R_4^T * R_5^T * R_6^T * v = R_3 * R_2 * R_1 * v_0 \qquad (10)$$

You can find the full solution in the figure 14.

$$\theta_1^* = atan2(\frac{-k_7}{cos(\theta_2)}, \frac{cos(\theta_3) * k_7 * k_4 - k_5 * cos(\theta_2)}{sin(\theta_2)})$$

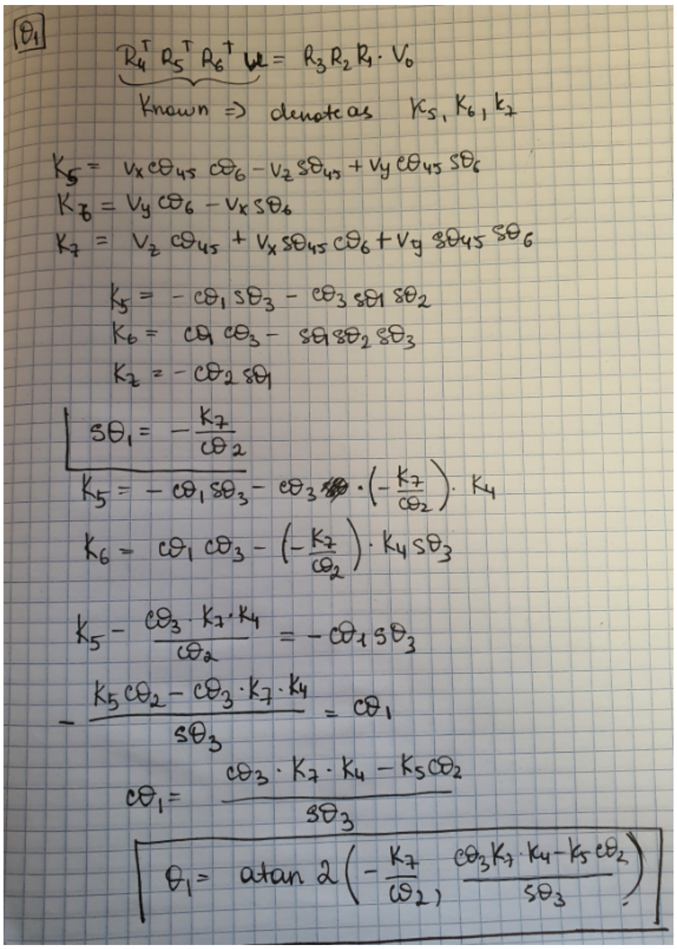



Fig. 14. Solution for the $\theta_1$

## *B. Validation using MATLAB*

To validate the model we need to inverse the input desired position matrix and multiply it by the transformation matrix. In original problem the end effector is at frame 6 and to move it frame 0 we need to inverse it. And then to translate it from the frame 0 to point HCP we need to post-multiply it by the translation matrix. $A_t \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & L2 \\ 0 & 0 & 1 & -L1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ First thing to check was knee bent position where $\theta_4 = \pi/6$ and the result is in fig.

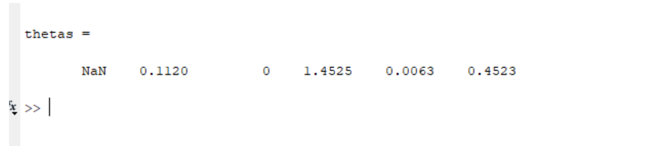



Fig. 15. Inverse Kinematics for $\theta_4 = \pi/2$

The $\theta_3 = \pi/6 change$ result can be seen in fig.16;

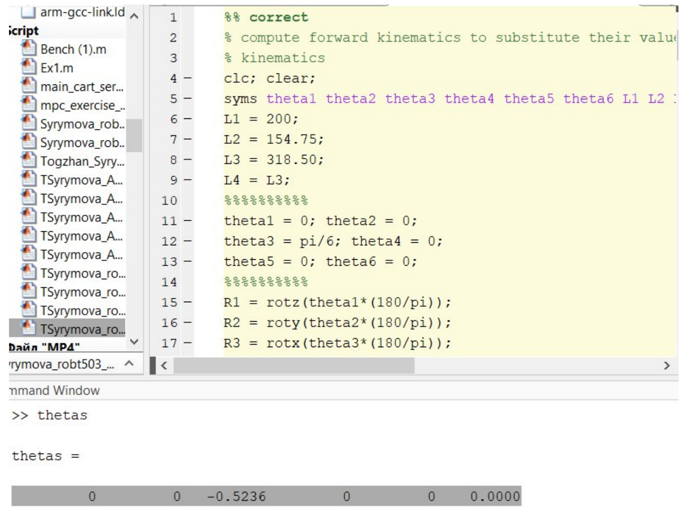



Fig. 16. Inverse Kinematics for $\theta_3 = \pi/6$

Some break points will be discussed in Conclusion section IV.

# IV. CONCLUSION

The Direct Kinematics of the NU-Biped robot leg was found and validate successfully. The visualization is shown in figure 7 and knee bent position 8. The program works correctly provides user with all relevant information.

As inverse kinematics was constructed, the validation process began. Main problem occured during testing period, Matlab calculation can not calculate division by zero, and returns $NaN$ result. it was considered to have 0 degree. Overall, the IK works with some mistakes, where the sign for some degree can change. The mistake was not found.

# REFERENCES

[1] J. J. Craig, *Introduction to robotics: mechanics and control, 3/E*. Pearson Education India, 2009.

[2] L.-W. Tsai, *Robot analysis: the mechanics of serial and parallel manipulators*. John Wiley & Sons, 1999.