System HW3

Chen,Zhijia

January 2020

1 Introduction

1.1 q1a

The degrees of freedom required for the manipulator is three. Two rotations and one translation.

1.2 q1b

The design of this manipulator can be similar to the ABB Scara robot series [2]. The difference is that the end-effector of this robot does not need to have rotation, since the candy can be placed in separate baskets, no matter what rotation it is. The end-effector can simply be a vacuum gripper.

1.3 q1c

For the actuators, since the manipulator is expected to move at a high speed, an electric actuator might be a good choice for this task. Because this task requires repeatable motion of the joints, stepper motor is a good selection to reduce the wear. For the transmission, gears would be selected.

1.4 q1d

RGB camera will be selected to detect the position and color of the candy. Force sensors will be chosen to detect the force required to lift the candy.

1.5 q1e

The joints are prone to wear from repeated task execution. The use of proper lubricants and abrasion-resistant cover can minimize the wear.

1.6 q1f

Using a fan to blow the conveyor might be able to blow away the candy whose weight is less than 5 grams. Or if the force sensor on the manipulator is accurate enough, it might be able to detect the weight of the candy and do not put them into the basket.

2 Q2

Seven degrees of freedom manipulators are often considered as redundant, since six degrees of freedom are enough to represent the pose of an end-effector. Therefore, the closed-form solutions of a seven degrees of freedom manipulator will be infinite. However, we can still obtain a set of finite solutions from infinite situations. A possible solution was proposed by Faria et al [1]. The steps of closed-form solutions will be elaborated in this section.

2.1 Analytic Kinematics

The DH parameters for KUKA iiwa are listed in Table 1.

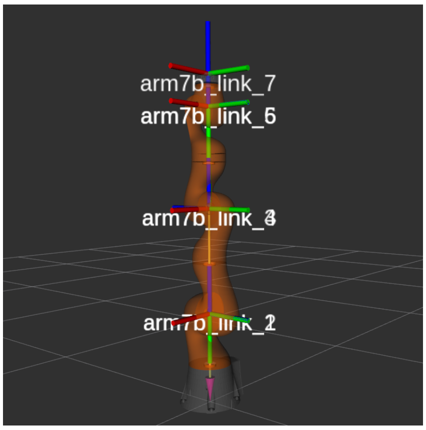


Figure 1: The KUKA robot

i	a_i	α_i	d_i	θ_i
1	0	-$\pi/2$	d$_{bs}$	θ_1
2	0	$\pi/2$	0	θ_2
3	0	$\pi/2$	d$_{se}$	θ_3
4	0	-$\pi/2$	0	θ_4
5	0	-$\pi/2$	d$_{ew}$	θ_5
6	0	$\pi/2$	0	θ_6
7	0	0	d$_{wf}$	θ_7

Table 1: Table1: DH parameters for KUKA iiwa

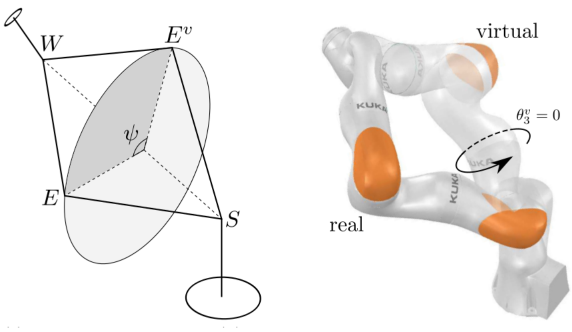
2.2 Self-Motion Parameters

Two parameters are introduced to address the global and local self-motion manifolds. The first parameter is the Global Configuration (GC), which specifies the branch of the inverse kinematics solutions for the global configuration manifold. The GC parameters are defined below:

$$GC_k = \begin{cases} 1 & if \theta_k \geq 0 \\ -1 & if \theta_k < 0 \end{cases}, \forall k \in \{2, 4, 6\} \tag{1}$$

Although we do not know whether the θ_k is positive or negative in advance, we can list all the possible joint configurations, which has eight cases.

The second parameter is arm angle ψ, which indicates the elbow position in the redundancy circle.



(a) (b)

Figure 2: (a) Representation of the arm angle ψ [1]. (b) Configuration of the real and virtual manipulator at the same pose [1].

The arm angle, represents the angle formed by the shoulder-elbow-wrist plane (SEW) and the reference plane (SE^vW), as shown in Figure 2. The way how to define the virtual plane (reference plane) in this report is proposed by Yan et al [3]. It can be proved that the shoulder and wrist positions are common to the real and virtual robot for the same target pose (Figure 2). In order to find a set of finite joint configurations, here the

ψ angle is defined in advance.

Now we can start to calculate the closed-form inverse kinematics. To simplify the notation, the vector from: base to shoulder (${}^0p_2$), shoulder to elbow (${}^2p_4$), elbow to wrist (${}^4p_6$) and wrist to flange (${}^6p_7$) according to the DH parameters are listed below:

$$ {}^0p_2 = \begin{bmatrix} 0 & 0 & d_{bs} \end{bmatrix}^T \tag{2} $$

$$ {}^2p_4 = \begin{bmatrix} 0 & d_{se} & 0 \end{bmatrix}^T \tag{3} $$

$$ {}^4p_6 = \begin{bmatrix} 0 & 0 & d_{ew} \end{bmatrix}^T \tag{4} $$

$$ {}^6p_7 = \begin{bmatrix} 0 & 0 & d_{wf} \end{bmatrix}^T \tag{5} $$

The virtual elbow joint (θ_4^v) is the first to be calculated, since it only depends on the manipulator's kinematic structure and on the shoulder-wrist vector. The shoulder-wrist vector is calculated from:

$$ {}^2p_6 = {}^0p_7 - {}^0p_2 - ({}^0R_7 \cdot {}^6p_7) \tag{6} $$

and θ_4^v is computed using the law of cosines

$$ \theta_4^v = GC_4 arccos(\frac{\|{}^2p_6\|^2 - (d_{se})^2 - (d_{ew})^2}{2d_{se}d_{ew}}) \tag{7} $$

Since θ_3^v is 0, the shoulder-elbow vector (${}^2p_4$) and the elbow-wrist vector (${}^4p_6$) are in the xy-plane. The joint 1_v is thus responsible for moving the virtual arm to the wrist x- and y- position coordinates.

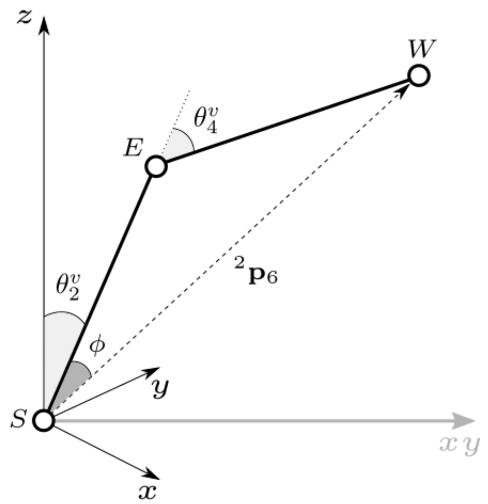


Figure 3: Variables required for the θ_2^v calculation

If the shoulder-wrist vector (${}^2p_6$) is aligned to the z-axis of joint 1 (${}^0R_{1,z}$), then joint 1 is no longer defined and a singularity will occur. Therefore, the calculation of θ_1^v would be,

$$ \theta_1^v = \begin{cases} atan2({}^2p_{6,y}, {}^2p_{6,x}) & \|{}^2p_6 \times {}^0R_{1,z}\| > 0 \\ 0 & \|{}^2p_6 \times {}^0R_{1,z}\| = 0 \end{cases} \tag{8} $$

The angle ϕ can be calculated with the eqaution

$$ \phi = arccos(\frac{(d_{se})^2 + \|{}^2p_6\|^2 - (d_{ew})^2}{2d_{se}\|{}^2p_6\|}) \tag{9} $$

And the value of θ_2^v is

$$ \theta_2^v = atan2(\sqrt{({}^2p_{6,x})^2 + ({}^2p_{6,y})^2}, {}^2p_{6,z}) + GC_4\phi \tag{10} $$

2.3 Inverse Kinematics

The inverse kinematics algorithm requires the final transformation matrix ${}^0T_7$, arm angle ψ, and GC values from the joint positions. θ_4 is the only joint angle that is independent to ψ. In order to determine the other joint values, we need to find the virtual elbow pose at the reference plane (${}^0T_4^v$) first. The real elbow pose is no more than the virtual elbow pose, rotated around the shoulder-wrist axis (${}^2p_6$) of ψ,

$$ {}^0R_4 = {}^0R_\psi^0 R_4^v \tag{11} $$

which is equal to

$$ {}^0R_3 = {}^0R_\psi^0 R_3^v \tag{12} $$

because θ_4 is the same in the virtual and the real manipulator, therefore, ${}^3R_4^v = {}^3R_4$.The elbow redundancy rotation matrix (${}^0R_\psi$) codes the rotation of the angle ψ around the shoulder-wrist vector (${}^2p_6$), Figure2 (a). It is calculated using the Rodrigues's rotation formula in matrix notation,

$$ {}^0R_\psi = I_3 + sin(\psi)[\widehat{{}^2p_6}\times] + (1 - cos(\psi))[\widehat{{}^2p_6}\times]^2 \tag{13} $$

where $\widehat{{}^2p_6}\times$ is the cross-product matrix for the unit vector$\widehat{{}^2p_6}$. Substituting equation 13 into equation 12, ${}^0R_3$ can be obtained in terms of three auxiliary matrices A_s ,B_s, and C_s,

$$ {}^0R_3 = A_s sin(\psi) + B_s cos(\psi) + C_s \tag{14} $$

where,

$$ A_s = [\widehat{{}^2p_6}\times] \cdot {}^0R_3^v \tag{15} $$

$$ B_s = -[\widehat{{}^2p_6}\times]^2 \cdot {}^0R_3^v \tag{16} $$

$$ C_s = [\widehat{{}^2p_6}\widehat{{}^2p_6}^T]^2 \cdot {}^0R_3^v \tag{17} $$

The real values of θ_1,θ_2, andθ_3 are now determined analytically by combining the elements of the${}^0R_3$ matrix – derived from DH parameters – in trigonometric operations,

$$ {}^0R_3 = \begin{bmatrix} * & cos(\theta_1)sin(\theta_2) & * \\ * & sin(\theta_1)sin(\theta_2) & * \\ -sin(\theta_2)cos(\theta_3) & cos(\theta_2) & -sin(\theta_2)sin(\theta_3) \end{bmatrix} \tag{18} $$

$$ \theta 1 = atan2(GC_2[A_{s22}sin(\psi) + B_{s22}cos(\psi) + C_{s22}], GC_2[A_{s12}sin(\psi) + B_{s12}cos(\psi) + C_{s12}]) \tag{19} $$

$$ \theta 2 = GC_2 arccos(A_{s32}sin(\psi) + B_{s32}cos(\psi) + C_{s32}) \tag{20} $$

$$ \theta 3 = atan2(GC_2[-A_{s33}sin(\psi) - B_{s33}cos(\psi) - C_{s33}], GC_2[-A_{s31}sin(\psi) - B_{s31}cos(\psi) - C_{s31}) \tag{21} $$

Similarly, we can calculate ${}^4R_7$

$$ {}^4R_7 = A_w sin(\psi) + B_w cos(\psi) + C_w \tag{22} $$

where,

$$ A_w = {}^3R_4^T A_s^{T0} R_7 \tag{23} $$

$$ B_w = {}^3R_4^T B_s^{T0} R_7 \tag{24} $$

$$ C_w = {}^3R_4^T C_s^{T0} R_7 \tag{25} $$

$$ {}^4R_7 = \begin{bmatrix} * & * & cos(\theta_5)sin(\theta_6) \\ * & * & sin(\theta_5)sin(\theta_6) \\ -sin(\theta_6)cos(\theta_7) & sin(\theta_6)sin(\theta_7) & cos(\theta_6) \end{bmatrix} \tag{26} $$

Then we can compute the remaining joint values,

$$ \theta_5 = atan2(GC_6[a_{w23}sin(\psi) + b_{w23}cos(\psi) + c_{w23}], GC_6[a_{w13}sin(\psi) + b_{w13}cos(\psi) + c_{w13}]) \tag{27} $$

$$ \theta_6 = GC_6 arccos(a_{w33}sin(\psi) + b_{w33}cos(\psi) + c_{w33}) \tag{28} $$

$$ \theta_7 = atan2(GC_6[a_{w32}sin(\psi) + b_{w32}cos(\psi) + c_{w32}], GC_6[-a_{w31}sin(\psi) - b_{w31}cos(\psi) - c_{w31}]) \tag{29} $$

The output has eight estimated poses, and one of the poses is the same as the pose of the input ${}^0T_7$, which means the result is correct. Assume we know the pose, then we can calculate the arm angle ψ first, then we can use this ψ to estimated the pose, to see whether the estimated pose is the same as the original one. If I did not do this, it is hard to check whether the result is correct because the solution is infinite. The step to calculate the ψ is mentioned in [3].

3 Q3

Forward dynamics is to find the frame of the center of mass (CoM) of each link that is relative to the frame of the base. The difficulty of the problem is that sometimes it is hard to know the exact position of the CoM of each link. The complicated shape and combination of different materials will increase the difficulty.

The application of finding CoM is to describe the relationship between force and motion, so that we can calculate the amount of force applied in each joint to overcome the influence of gravity, external loads, stiffness and move along a desired trajectory.

4 Q4

Inverse kinematics is to find the joint variables with a given robot pose. There are two ways to find the inverse kinematics.

The first way is iterative method. The problem of this method is that the finally result might not be valid, and the iteration might not be converged.

The second way is to use closed-form inverse kinematics method. Basically, six degrees of freedom is enough to describe the pose of the end-effector. The difficulty of finding the inverse kinematics of seven DoF manipulator is that the unknowns are more than the equations, which means the solution is infinite, especially for the elbow part. It is a highly non-linear problem.

The application of inverse dynamics is to compute the joint forces/torques necessary to move a robot arm from one Cartesian coordinate to another. Inverse Dynamics also helps compute the forces when external forces are involved. Inverse Dynamics can help compute the internal forces for a humanoid picking something up, or carrying something.

5 Q5

5.1 Q5a

5.1.1 Calculation of acceleration

The matrix formulation of the equation of motion is

$$B(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau \tag{30}$$

Therefore, the acceleration can be computed as

$$\ddot{q} = B(q)^{-1}(\tau - C(q, \dot{q})\dot{q} - g(q)) \tag{31}$$

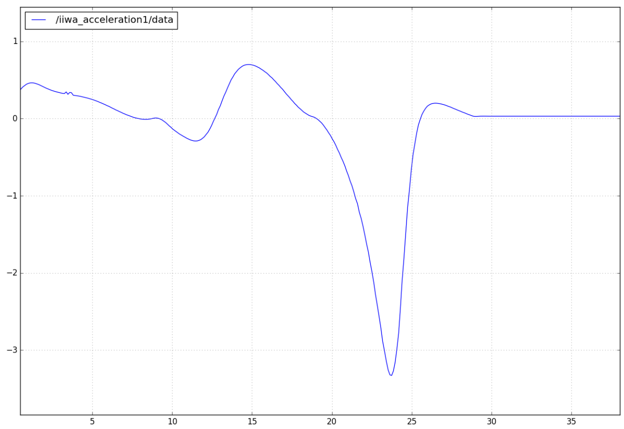


Figure 4: Acceleration for joint 1

$\ddot{q}$ is a seven by one matrix, containing seven accelerations from joint 1 to joint 7. Figure 4 is the acceleration for joint 1. The acceleration for the other joints can be seen in the Appendix.

5.1.2 Forward dynamics

This problem is forward dynamics.

5.2 Q5b

5.2.1 Calculation for mass

Forces and torques exerted on the robot has the relationship as the equation below:

$$B(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = \tau + \tau_{ext} \tag{32}$$

Therefore, the torques generated by the mass of the box is

$$\tau_{ext} = B(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) - \tau \tag{33}$$

If we project all forces and torques in Cartesian space to joint space, we have to convert them into joint space using Jacobian matrices,

$$\tau_{F,ext} = \sum_{i=1}^{n} J_P^{(F_i)T} F_{i,ext} \tag{34}$$

In this question, there is only one external force, which is the gravity of the box. Therefore,

$$\tau_{ext} = J_P^{(F_1)T} F_{ext} \tag{35}$$

τ_{ext} is a seven by one matrix, if we just consider joint seven now, we can have

$$\begin{bmatrix} z_{71} \\ z_{72} \\ z_{73} \end{bmatrix} \times \begin{bmatrix} x - p_{71} \\ y - p_{72} \\ z - p_{73} \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ -9.81m \end{bmatrix} = \tau_7 \tag{36}$$

Here, x, y, and z are the coordinates of the center of mass of the box relative to the base frame, m is the mass of the box. Then we have,

$$z_{72} \cdot (z - p_{73}) - z_{73} \cdot (y - p_{72}) \cdot 0 = \tau_7 \tag{37}$$

$$z_{73} \cdot (x - p_{71}) - z_{71} \cdot (z - p_{73}) \cdot 0 = \tau_7 \tag{38}$$

$$[z_{71} \cdot (y - p_{72}) - z_{72} \cdot (x - p_{71})] \cdot (-9.81m) = \tau_7 \tag{39}$$

Euqation 39 can be expressed in a matrix form,

$$(-9.81) \cdot \begin{bmatrix} -z_{72} & z_{71} & -z_{71}p_{72} + z_{72}p_{71} \end{bmatrix} \begin{bmatrix} xm \\ ym \\ m \end{bmatrix} = \tau_7 \tag{40}$$

We can obtain the same equation for the other joints, and put them together in a matrix equation.

$$(-9.81) \cdot \begin{bmatrix} -z_{72} & z_{71} & -z_{71}p_{72} + z_{72}p_{71} \\ -z_{62} & z_{61} & -z_{61}p_{62} + z_{62}p_{61} \\ -z_{52} & z_{51} & -z_{51}p_{52} + z_{52}p_{51} \\ -z_{42} & z_{41} & -z_{41}p_{42} + z_{42}p_{41} \\ -z_{32} & z_{31} & -z_{31}p_{32} + z_{32}p_{31} \\ -z_{22} & z_{21} & -z_{21}p_{22} + z_{22}p_{21} \\ -z_{12} & z_{11} & -z_{11}p_{12} + z_{12}p_{11} \end{bmatrix} \begin{bmatrix} xm \\ ym \\ m \end{bmatrix} = \begin{bmatrix} \tau_7 \\ \tau_6 \\ \tau_5 \\ \tau_4 \\ \tau_3 \\ \tau_2 \\ \tau_1 \end{bmatrix} \tag{41}$$

Equation 41 can be expressed as

$$A \cdot b = \tau_{ext} \tag{42}$$

Where b is the matrix we need to obtain. If we use pseudoinverse, we can solve b.

$$b = \begin{bmatrix} xm \\ dym \\ m \end{bmatrix} = (A^T \cdot A)^{-1} \cdot A^T \cdot \tau_{ext} \tag{43}$$

Then we can have x,y,m, where m is the mass of the box, x,y is the global positions of the center of the box. Since we have three trajectories, we can compute the mass three times. The average of the mass is 1.0802kg, and the error is less than 5 percents.

5.2.2 Calculation for the center of mass relative to frame 7

In this question, we have three configurations, each configuration can obtain the x,y global position of the center of mass of the box.

For trajectory 1, we can get ${}^0T_7$. Assume the relative position to the frame 7 is $\hat{x}$, $\hat{y}$,and $\hat{z}$, and the frame of the center of mass does not rotate. Thus we can get the transformation from the base frame to the frame of the center of mass (${}^0T_cm$).

$$
{}^0T_{cm} = {}^0T_7 \cdot {}^7T_{cm} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & p_1 \\ a_{21} & a_{22} & a_{23} & p_2 \\ a_{31} & a_{32} & a_{33} & p_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & \hat{x} \\ 0 & 1 & 0 & \hat{y} \\ 0 & 0 & 1 & \hat{z} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{44}
$$

and then we can have,

$$
a_{11}\hat{x} + a_{12}\hat{y} + a_{13}\hat{z} + p_1 = x_1 \tag{45}
$$

$$
a_{21}\hat{x} + a_{22}\hat{y} + a_{23}\hat{z} + p_2 = y_1 \tag{46}
$$

where x_1, y_1 are the global positions of the center of mass in trajectory 1. Similarly, we can get these equations for trajectory 2 and 3, and we can put them in matrix form together,

$$
\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \end{bmatrix} \cdot \begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{bmatrix} = \begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ x_3 \\ y_3 \end{bmatrix} \tag{47}
$$

We can use the trick as equation 43 again. If we want to obtain $\hat{x}$, $\hat{y}$,and $\hat{z}$, we just need to use pseudoinverse. Finally, the $\hat{x}$ is -0.000420m (≈ 0), $\hat{y}$ is -0.000257m (≈ 0), and $\hat{z}$ is 0.150543m. All of the errors are less than 7 percents.

5.2.3 Calculate the external torque exerting on each joint when the joint position q_d

Similar to equation 33,

$$
\tau_{ext} = B(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) - \tau \tag{48}
$$

The external torques for the pose $[0^0, -90^0, -90^0, 90^0, -90^0, 90^0, -30^0,]^T$ are $[\tau_1 = -0.000002, \tau_2 = -4.438260, \tau_3 = 4.183485, \tau_4 = -0.000454, \tau_5 = 0.019202, \tau_6 = -0.061390, \tau_7 = 0.000001]$. Each time the external torques might be different due to error.

References

[1] Carlos Faria, Flora Ferreira, Wolfram Erlhagen, Sérgio Monteiro, and Estela Bicho. Position-based kinematics for 7-dof serial manipulators with global configuration control, joint limit and singularity avoidance. *Mechanism and Machine Theory*, 121:317–334, 2018.

[2] https://www.robots.com/series/abb-scara-robot series.

[3] Lei Yan, Zonggao Mu, and Wenfu Xu. Analytical inverse kinematics of a class of redundant manipulator based on dual arm-angle parameterization. In *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 3744–3749. IEEE, 2014.

6 Appendix

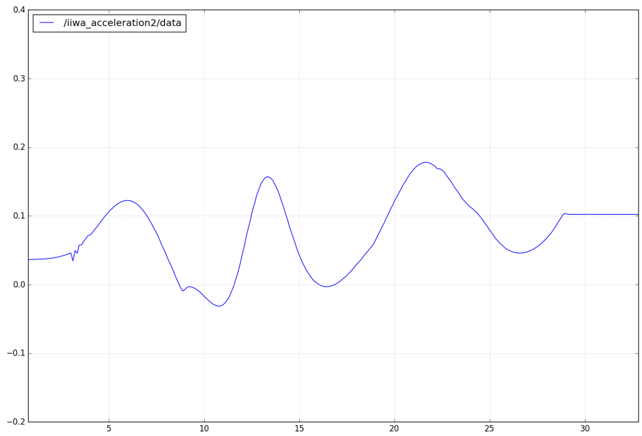


Figure 5: Acceleration for joint 2

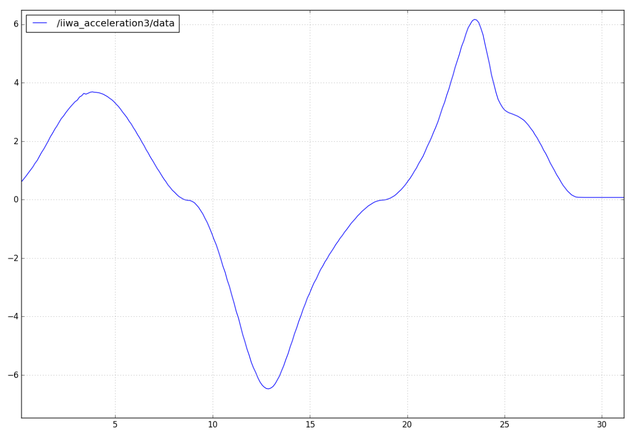


Figure 6: Acceleration for joint 3

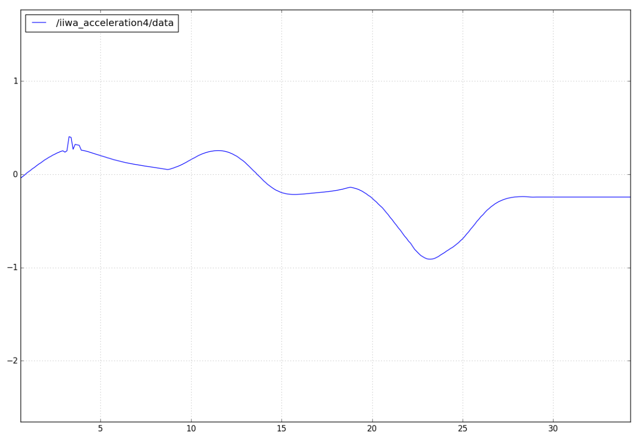


Figure 7: Acceleration for joint 4

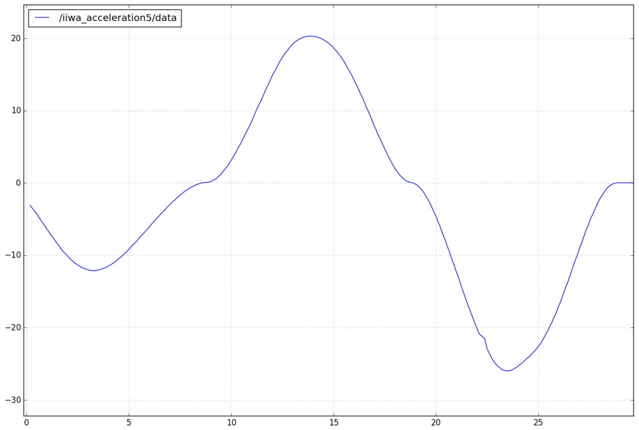


Figure 8: Acceleration for joint 5

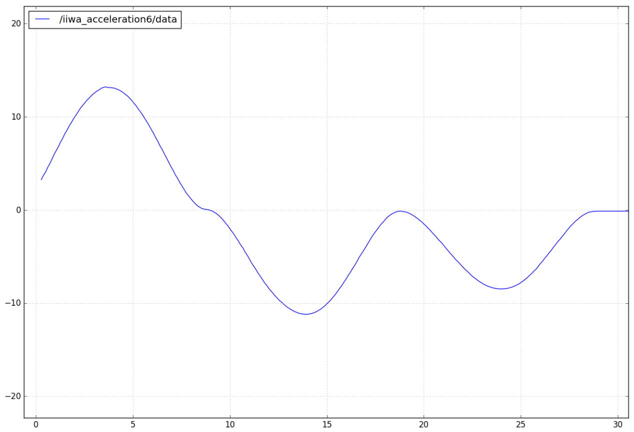


Figure 9: Acceleration for joint 6

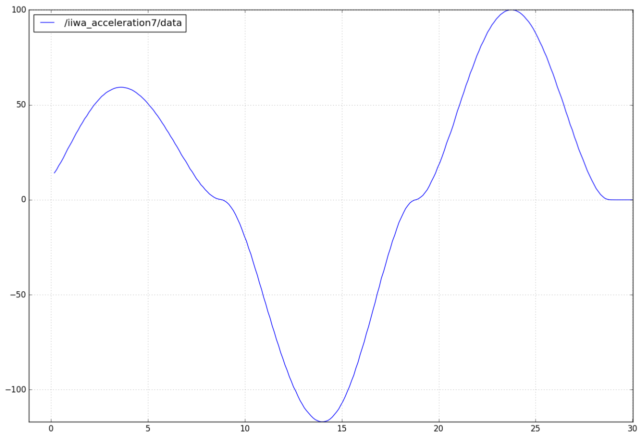


Figure 10: Acceleration for joint 7