Assignment 5 - Programming Project

CMPSCI 603: Robotics

Sanuj Bhatia

April 19, 2018

3.4 **Project #7 - ChaseTouch()**

(1) A description of the finite state supervisor.
Solution:

Case	**Touch**	**Chase**	**Action/Return State**
0	0	0	*home* position/0
1	0	1	*home* position; chase setpoints/1
2	0	2	no action/1
3	1	0	*home* position/0
4	1	1	no action/1
5	1	2	no action/1
6	2	0	*home* position/0
7	2	1	*home* position/0
8	2	2	no action/2

Search track, chase, and touch are all called once before the switch statement. The action of *no action* in the table means that the switch statement is empty, not that Roger is still. Touch will give setpoints to the arms in all cases, if the ball is in range. Search track will always be running if the ball is not in vision, and these actions are not captured in the switch cases. The home position is for the arms to retract by Roger's sides. This happens when the ball is not in vision or out of range.

(2) **Solution:** Shown on the next page is the plot Roger's base heading in blue, distance of the base from the ball in orange, and left and right arm distances from the ball in green and red respectively. The first 240 ms, search track runs and looks for the ball, which it finds at the point indicated by the vertical line near 240ms. The ball is far away, and its position estimate is inaccurate and fluctuates, which explains the jitter in the graph. As the CHASE function comes into play and makes the base rotate as well as move it toward the ball, the estimate of the ball's position improves and makes the plot lines more stable. The right arm at around 300 ms begins with a distance more than that of the base and the left arm. This happens because the ball was found in a positive angle to Roger's left, and the distance of both arms equalizes as Roger comes to face the ball (shown by the stabilization of the base heading) and has his arms to his sides. At this point (from 420 ms to 500 ms), the base faces the ball directly and its distance is minimum to the ball among the base and arms, and decreases as it moves toward the ball. As soon as the arms come into range of the ball, they start moving towards it at 500 ms, when TOUCH begins. The distance of the base remains relatively stable during this as the arms go closer to the ball. At 660 ms, the right arm touches the ball, which is shown as a point of minimum in the red line, and the left arm touches the ball shortly after at 695 ms. We can observe that the base distance decreases as the arms touch the ball, applying forces that bring it towards the base itself, and this process ends in a grasp-like configuration.

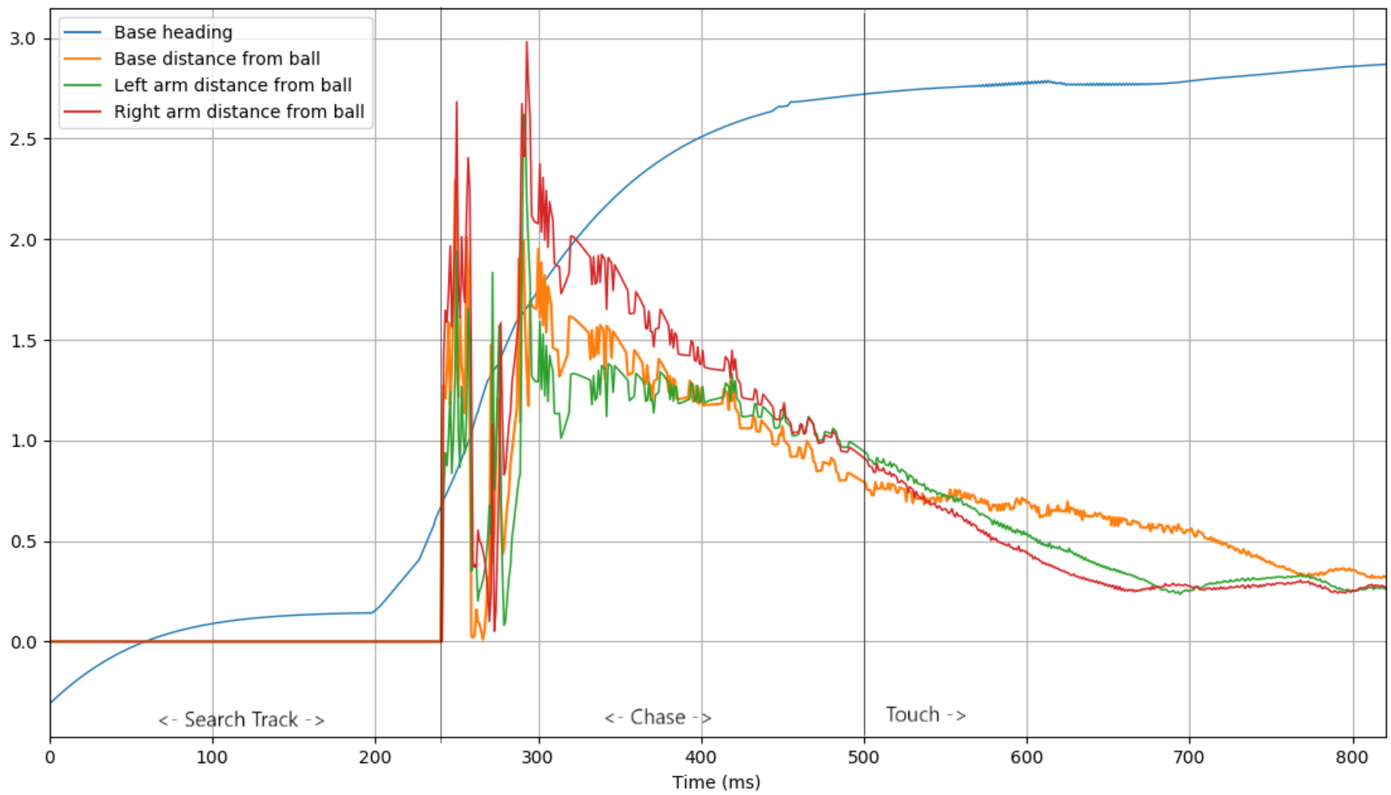


Figure 1

(3) Discussion

Solution: Changing the gains of the base translation and arms will affect how the ball is 'touched'. For precise control, the arms need to approach the ball slowly, and high gains will be needed for a punching motion. In order to 'grasp' the ball, we need somewhat precise control and so high gains will not work here. Establishing pre-postures for the arms is important so that they don't collide into walls or touch the ball without meaning to while looking for it. This also offers a way to think about the path the arms take to get to the ball from a fixed default position, and so we can reason about the approach. The base needs to slow down and stop before it collides with the ball, and so the offset needs to be bigger than at least the radius of the ball. Any comfortable distance from the ball is good, giving it enough time to slow down. It should not be at the edge of the workspace of the arms, however, as a 'grasping' movement will be impossible if the ball can only be touched from the direction of approach. As we extend our arms towards the ball, we can slowly move our base towards it and time it so that both the arms and the base close their respective distances to the ball together - this will be a more robust position for the ball as any extra forces will be compensated instantly. Moving the base to the ball first without the arms present or vice-versa could lead to the ball escaping the grasp. Again, if we need more power while touching the ball, we can simple move towards it faster and time our arms to punch the ball with the added momentum of the moving base. All of these factors will change and affect how Roger interacts with the ball, and whether he is able to successfully grasp it.

Assignment 5 - Written Exercises

CMPSCI 603: Robotics

Sanuj Bhatia

April 19, 2018

Chapter 6

6.3.1 Grip Jacobian

(a) **Solution:** Using the directions of the forces with respect to the given coordinate system:

$$
\begin{aligned}
w =& G\lambda \\
\begin{bmatrix} f_x \\ f_y \\ m_z \end{bmatrix} =& \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & R & 0 & R \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \end{bmatrix}
\end{aligned}
$$

(b) **Solution:**

$$\lambda_1 \geq 0 \tag{1}$$

$$\lambda_3 \geq 0 \tag{2}$$

$$|\lambda_2| \leq \mu\lambda_1 \tag{3}$$

$$|\lambda_4| \leq \mu\lambda_3 \tag{4}$$

(c) **Solution:** We have $w = [0\ 0\ 0.5]^T$. Therefore,

$$
\begin{aligned}
f_x = 0 &\implies \lambda_1 = \lambda_3 \\
f_y = 0 &\implies \lambda_2 = \lambda_4 \\
m_z = 0.5 &\implies R(\lambda_2 + \lambda_4) = 0.5 \\
&\implies \lambda_2 = \lambda_4 = 10 \\
(b)\ 3, 4 &\implies \lambda_1 \geq \frac{\lambda_2}{\mu},\ \lambda_3 \geq \frac{\lambda_3}{\mu} \\
&\implies \lambda_1 \geq 20,\ \lambda_3 \geq 20
\end{aligned}
$$

One solution can thus be $\lambda = [20\ 10\ 20\ 10]^T$, when the force due to friction is maximum.

6.3.3 Grasping - Redundant Contact Control

$$
\begin{aligned}
W_1^T &= [-cos\theta_1\ \ -sin\theta_1\ 0],\ W_2^T = [-cos\theta_2\ \ -sin\theta_2\ 0] \\
\rho &= \sum_{i=1}^{2} W_i = [-(cos\theta_1 + cos\theta_2)\ \ -(sin\theta_1 + sin\theta_2)\ 0]^T
\end{aligned}
$$

(a) **Solution:**

$$\epsilon = \rho^T \rho = [-(cos\theta_1 + cos\theta_2) \;\; -(sin\theta_1 + sin\theta_2) \;\; 0] \begin{bmatrix} -(cos\theta_1 + cos\theta_2) \\ -(sin\theta_1 + sin\theta_2) \\ 0 \end{bmatrix}$$
$$= (cos\theta_1 + cos\theta_2)^2 + (sin\theta_1 + sin\theta_2)^2$$
$$= 2 + 2cos\theta_1 cos\theta_2 + 2sin\theta_1 sin\theta_2$$
$$\implies \epsilon = 2(1 + cos(\theta_1 - \theta_2))$$

(b) **Solution:**

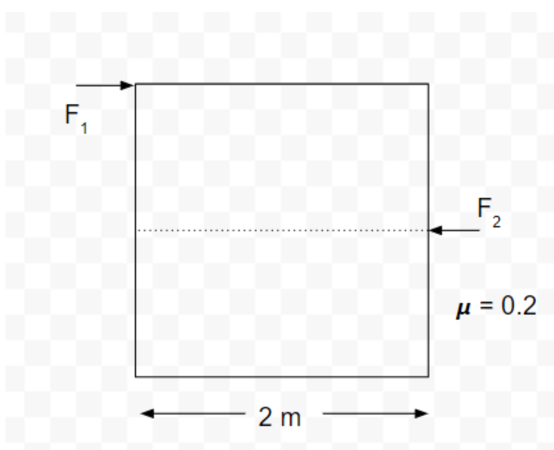
$$J^F = \begin{bmatrix} \frac{\partial \epsilon}{\partial \theta_1} & \frac{\partial \epsilon}{\partial \theta_2} \end{bmatrix} = \begin{bmatrix} 2sin(\theta_2 - \theta_1) & 2sin(\theta_1 - \theta_2) \end{bmatrix}$$

(c) **Solution:**

$$J^F = \begin{bmatrix} 2sin(\theta_2 - \theta_1) & 2sin(\theta_1 - \theta_2) \end{bmatrix}, \; (J^F)^T = \begin{bmatrix} 2sin(\theta_2 - \theta_1) \\ 2sin(\theta_1 - \theta_2) \end{bmatrix}$$
$$(J^F)^\# = (J^F)^T \left[J^F (J^F)^T \right]^{-1}$$
$$= \begin{bmatrix} 2sin(\theta_2 - \theta_1) \\ 2sin(\theta_1 - \theta_2) \end{bmatrix} \left[\begin{bmatrix} 2sin(\theta_2 - \theta_1) & 2sin(\theta_1 - \theta_2) \end{bmatrix} \begin{bmatrix} 2sin(\theta_2 - \theta_1) \\ 2sin(\theta_1 - \theta_2) \end{bmatrix} \right]^{-1}$$
$$= \begin{bmatrix} 2sin(\theta_2 - \theta_1) \\ 2sin(\theta_1 - \theta_2) \end{bmatrix} \left[8sin^2(\theta_2 - \theta_1) \right]^{-1}$$
$$\implies (J^F)^\# = \frac{1}{4sin(\theta_2 - \theta_1)} \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

6.3.4 Create your own homework

Question: As shown in the figure, two point contacts F_1 (frictionless) and F_2 apply forces on the edges of a square. What is the minimum value of F_2 to make the square not rotate, given that $F_1 = 10\ N$.



Answer: The moment due to F_1 is $1 \times 10 = 10\ Nm$. This needs to be canceled by the friction force we can apply due to F_2, since F_2 itself passes through the centre of the square. The minimum friction force $f \times 1 = 10 \implies f = 10\ N$. This frictional force needs to be applied in the upward direction, against the torque applied by F_1. Therefore, the minimum force $F_2 = \frac{f}{\mu} \implies F_2 = 50\ N$ in the direction shown.