Universidad Rey Juan Carlos

Universidad Rey Juan Carlos

E.T.S. Ingeniería de Telecomunicación

Ingeniería de Control

Práctica 3

Autor:
Javier Izquierdo Hernández

December 23, 2023

Práctica 3

Consider the robot manipulator represented in Figure 1, which moves in a vertical plane.

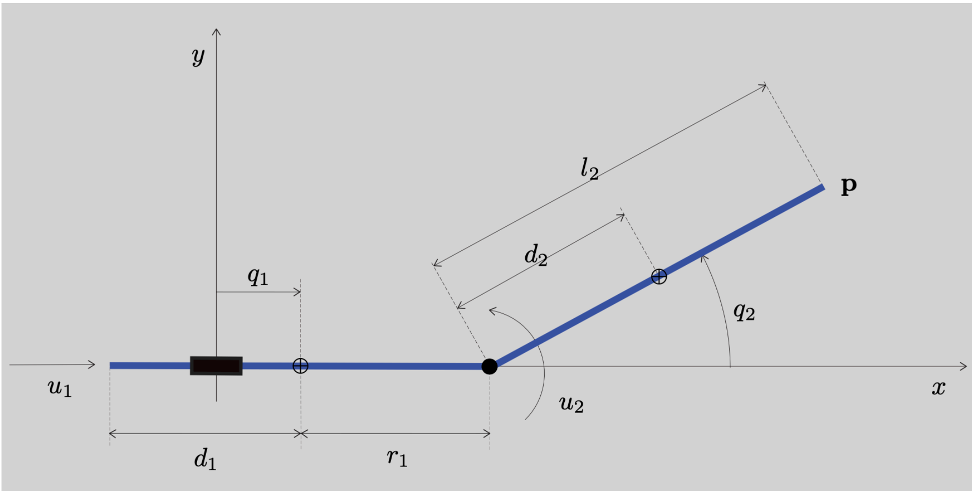


Figure 1: Planar vertical robot manipulator.

The dynamic model of this robotic system is represented by the second order differential equation

$$\mathbf{B}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{N}(\mathbf{q}) = \mathbf{u},$$

where the matrices $\mathbf{B}(\mathbf{q})$, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$, and $\mathbf{N}(\mathbf{q})$ have the following expressions

$$\begin{aligned}
\mathbf{B}(\mathbf{q}) &= \begin{bmatrix} m_1 + m_2 & -m_2 d_2 \sin(q_2) \\ -m_2 d_2 \sin(q_2) & I_2 + m_2 d_2^2 \end{bmatrix}, \\
\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) &= \begin{bmatrix} -m_2 d_2 \cos(q_2) \dot{q}_2^2 \\ 0 \end{bmatrix}, \\
\mathbf{N}(\mathbf{q}) &= \begin{bmatrix} 0 \\ m_2 g d_2 \cos(q_2) \end{bmatrix}.
\end{aligned}$$

The vector $\mathbf{q} = (q_1, q_2)^T$ is the vector of configuration variables, where q_1 is the linear position of the center of mass of link 1 with respect to the y axis of the reference frame $\{x, y\}$ and q_2 is the angular position of link 2 with respect to link 1 as illustrated in Figure 1. The vector $\dot{\mathbf{q}} = (\dot{q}_1, \dot{q}_2)^T$ is the vector of joint velocities, where $\dot{q}_1$ is the linear velocity of link 1 and $\dot{q}_2$ is the angular velocity of link 2. The vector $\ddot{\mathbf{q}} = (\ddot{q}_1, \ddot{q}_2)^T$ is the vector of joint velocities, where $\ddot{q}_1$ is the linear acceleration of link 1 and $\ddot{q}_2$ is the angular acceleration of link 2, The control inputs of the system are $\mathbf{u} = (u_1, u_2)^T$, where u_1 is the force applied by the actuator at joint 1, and u_2 is the torque applied by the actuator at joint 2. l_1 is the length of link 1, l_2 is the length of link 2, m_1 is the mass of link 1, m_2 is the mass of link 2, I_1 is the barycentric moment of inertia of link 1, and I_2 is the barycentric moment of inertia of link 2. Distances d_1, r_1, and d_2 are defined in Figure 1. The parameters of the dynamic model of the robot manipulator are $l_1 = 1$ [m], $l_2 = 1$ [m], $m_1 = 1$ [kg], $m_2 = 1$ [kg], $I_1 = 1$ [kg m^2], $I_2 = 1$ [kg m^2], $d_1 = \frac{l_1}{2}$ [m], $r_1 = \frac{l_1}{2}$ [m], $d_2 = \frac{l_2}{2}$ [m]. Consider the following two robotic tasks.

a) Pick and place task: move iteratively the end effector $\mathbf{p}$ between the setpoints $\mathbf{p}_A = (1, 0.75)^T$ [m] and $\mathbf{p}_B = (1.5, -0.75)^T$ [m]. The motions must be rest to rest.

b) Tracking task: follow with the end effector a setpoint $\mathbf{w}$ describing the target line segment from $\mathbf{p}_A = (1, 0.75)^T$ [m] to $\mathbf{p}_B = (1.5, -0.75)^T$ [m], which moves with constant linear velocity 0.1 [m/s].

Suppose that, in both cases, at $t = 0$ [s] the position of the end effector is $\mathbf{p} = \mathbf{p}_A$.

1) Compute the kinetic and potential energy of the two links of the robot manipulator. (Contesta en el informe)

2) Compute the state space representation of the dynamics of the robot manipulator in which $\mathbf{x} = (x_1, x_2, x_3, x_4)^T = (q_1, q_2, \dot{q}_1, \dot{q}_2)^T$, where distances are measured in [m], angles in [rad], linear velocities in [m/s], and angular velocities in [rad/s]. (Contesta en el informe y sube el código Matlab a Aula Virtual en el fichero answer_2.m)

3) Compute the relation between the configuration variables $\mathbf{q}$ and the position of the end effector $\mathbf{p} = (p_1, p_2)^T$. (Contesta en el informe y sube el código Matlab a Aula Virtual en el fichero answer_3.m)

4) Compute the relation between the position of the end effector $\mathbf{p}$ and the configuration variables $\mathbf{q}$. Compute the values of the configuration variables that correspond to $\mathbf{p} = \mathbf{p}_A$ and $\mathbf{p} = \mathbf{p}_B$. (Contesta en el informe y sube el código Matlab a Aula Virtual en el fichero answer_4.m)

5) Compute the relation between the joint velocities $\dot{\mathbf{q}}$ and the velocity of the end effector $\dot{\mathbf{p}}$. (Contesta en el informe y sube el código Matlab a Aula Virtual en el fichero answer_5.m)

6) Compute the relation between the velocity of the end effector $\dot{\mathbf{p}}$ and the joint velocities $\dot{\mathbf{q}}$. Suppose that the robot manipulator is ready to execute the tracking task, that is, the position of the end effector is $\mathbf{p} = \mathbf{p}_A$. Compute the initial joint velocities to execute this task. (Contesta en el informe y sube el código Matlab a Aula Virtual en el fichero answer_6.m)

7) Compute the time derivative of the Jacobian matrix. (Contesta en el informe y sube el código Matlab a Aula Virtual en el fichero answer_7.m)

8) Design a controller based on the feedback linearization method to execute the pick and place task. Write a `Matlab` code that implements the controller to execute the task. Show, plotting the relevant variables and an animation, that the controller satisfies the specifications. (Contesta en el informe y sube el código `Matlab` a Aula Virtual en la carpeta answer_8)

9) Design a controller based on the feedback linearization method to execute the tracking task. Compute the duration of the manoeuvre. Write a `Matlab` code that implements the controller to execute the task. Show, plotting the relevant variables and an animation, that the controller satisfies the specifications. (Contesta en el informe y sube el código `Matlab` a Aula Virtual en la carpeta answer_9)

Justifica todas las respuestas

Solution

1) Compute the kinetic and potential energy of the two links of the robot manipulator. (Contesta en el informe)

Primero calculamos la energía cinética T:

- Brazo 1, solo tiene lineal porque solo se mueve sobre el eje x:

$$T_{B_1} = \frac{1}{2} m_1 \dot{q_1}^2$$

- Brazo 2, componente de rotación porque el brazo 2 rota:

$$T_{B_{2_{Rot}}} = \frac{1}{2} I_2 \dot{q_2}^2$$

- Brazo 2, componente lineal en x porque se mueve sobre el eje x:

$$T_{B_{2_x}} = \frac{1}{2} m_2 (\frac{d}{dt}(q_1 + r_1 + l_2\ cos(q_2)))^2$$

- Brazo 2, componente lineal en y porque se desplaza en el eje y:

$$T_{B_{2_y}} = \frac{1}{2} m_2 (\frac{d}{dt}(l_2\ sin(q_2)))^2$$

- Brazo 2:

$$T_{B_2} = T_{B_{2_{Rot}}} + T_{B_{2_x}} + T_{B_{2_y}}$$

Como se puede apreciar, la componente lineal de la esfera ha sido dividida en 2 ya que se mueve sobre los 2 ejes, no solo en uno.

$$\begin{gathered} T = T_{B_1} + T_{B_2} = \\ \frac{1}{2} m_1 \dot{q_1}^2 + \frac{1}{2} I_2 \dot{q_2}^2 + \frac{1}{2} m_2 (\frac{d}{dt}(q_1 + r_1 + l_2\ cos(q_2)))^2 + \frac{1}{2} m_2 (\frac{d}{dt}(l_2\ sin(q_2)))^2 = \\ \frac{1}{2} m_1 \dot{q_1}^2 + \frac{1}{2} I_2 \dot{q_2}^2 + \frac{1}{2} m_2 (\dot{q_1}\ - l_2\ \dot{q_2}\ sin(q_2))^2 + \frac{1}{2} m_2 (l_2\ \dot{q_2}\ cos(q_2))^2 = \\ m_1 \dot{q_1}^2 + \frac{1}{2} I_2 \dot{q_2}^2 + m_2\ l_2^2\ \dot{q_2}^2 - m_2\ l_2\ \dot{q_1}\ \dot{q_2} \end{gathered}$$

Por último obtenemos la energía potencial del sistema V. Solo brazo 2 tiene energía potencial:

$$V = m_2\ g\ l_2\ sin(q_2)$$

2) Compute the state space representation of the dynamics of the robot manipulator in which $\mathbf{x} = (x_1, x_2, x_3, x_4)^T = (q_1, q_2, \dot{q}_1, \dot{q}_2)^T$, where distances are measured in [m], angles in [rad], linear velocities in [m/s], and angular velocities in [rad/s]. (Contesta en el informe y sube el código Matlab a Aula Virtual en el fichero answer_2.m)

Para poder calcular la representación en el espacio de estados, necesitamos conocer $\ddot{q}$, para así poder conseguir $\dot{x}$. Luego, para obtener $\ddot{q}$ despejamos de:

$$\begin{gathered} B(q)\ddot{q} + C(q, \dot{q}) + N(q) = u \\ B^{-1}(q)B(q)\ddot{q} + B^{-1}(q)C(q, \dot{q}) + B^{-1}(q)N(q) = B^{-1}(q)u \\ \ddot{q} + B^{-1}(q)C(q, \dot{q}) + B^{-1}(q)N(q) = B^{-1}(q)u \\ \ddot{q} = B^{-1}(q)u - B^{-1}(q)C(q, \dot{q}) - B^{-1}(q)N(q) \end{gathered}$$

Además como sabemos que $\dot{q_1} = x_3$ y que $\dot{q_2} = x_4$ el espacio de estados quedaría así:

$$\begin{pmatrix} \dot{x_1} \\ \dot{x_2} \end{pmatrix} = \begin{pmatrix} \dot{q_1} \\ \dot{q_2} \end{pmatrix}$$
$$\begin{pmatrix} \dot{x_3} \\ \dot{x_4} \end{pmatrix} = B^{-1}(q)u - B^{-1}(q)C(q,\dot{q}) - B^{-1}(q)N(q)$$

File answer_2.m

```
syms q1 q2 dotq1 dotq2 ddotq1 ddotq2 u1 u2 x1 x2 x3 x4;
syms d1 d2 m1 m2 r1 l1 l2 I1 I2 g

B = [m1+m2, -m2*d2*sin(q2); -m2*d2*sin(q2), I2+m2*d2*d2]
invB = inv(B)

C = [-m2*d2*cos(q2)*dotq2*dotq2; 0];
N = [0;m2*g*d2*cos(q2)]

xdot= [dotq1;dotq2;-invB*C-invB*N+invB*[u1;u2]];
xdot = subs(xdot,[q1,q2,dotq1,dotq2], [x1,x2,x3,x4])
```

3) Compute the relation between the configuration variables $\mathbf{q}$ and the position of the end effector $\mathbf{p} = (p_1, p_2)^T$. (Contesta en el informe y sube el código Matlab a Aula Virtual en el fichero answer_3.m)

Para ello estudiamos la figura original, para determinar la posición del extremo para x e y.

Para x tenemos que se desplaza horizontalmente para el brazo 1 en relación con q_1 y r_1, y para el brazo 2 en relación a la longitud del brazo 2 l_2 y a su proyección en el eje x: $cos(q_2)$.

$$p_1 = p_x = q_1 + r_1 + l_2 cos(q_2)$$

Ahora, para la posición en el eje y, como el primer brazo no se desplaza en este eje, la posición solo estará determinada por el segundo. Luego la posición en el eje y será:

$$p_2 = p_y = l_2 sin(q_2)$$

Con todo esto tenemos que el extremo se encuentra en la posición:

$$p = \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} = \begin{pmatrix} p_x \\ p_y \end{pmatrix} = \begin{pmatrix} q_1 + r_1 + l_2 cos(q_2) \\ l_2 sin(q_2) \end{pmatrix}$$

File answer_3.m

```
p1 = q1 + r1 + l2*cos(q2);
p2 = l2*sin(q2);
```

4) Compute the relation between the position of the end effector $\mathbf{p}$ and the configuration variables $\mathbf{q}$. Compute the values of the configuration variables that correspond to $\mathbf{p} = \mathbf{p}_A$ and $\mathbf{p} = \mathbf{p}_B$. (Contesta en el informe y sube el código Matlab a Aula Virtual en el fichero answer_4.m)

Usamos Matlab para resolver las ecuaciones obtenidas en el apartado anterior, para así obtener los valores iniciales de $q_1 = x_1$ y de $q_2 = x_2$.

Primero resolvemos para $p = \begin{pmatrix} 1 \\ 0.75 \end{pmatrix}$ y obtenemos 2 valores posibles para $q_1 = \begin{pmatrix} 1.16 \\ -0.16 \end{pmatrix}$ y $q_2 = \begin{pmatrix} 2.29 \\ 0.85 \end{pmatrix}$, pero de estos solo podemos elegir uno. En este caso elegimos el segundo valor de cada uno de ellos, ya que para q_1 el valor siempre debe ser menor que la longitud del brazo 1, en este caso $l_1 = 1$, y para q_2 en este modelo sería imposible que el brazo 2 formara un ángulo de $2.29\ rad$ para llegar a ese punto, ya que para eso, el brazo 1 debería ser bastante más largo.

Luego tenemos que para llegar a p_A, q tiene que valer:

$$q = \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} = \begin{pmatrix} -0.16 \\ 0.85 \end{pmatrix}$$

Por último calculamos lo mismo pero para $p = \begin{pmatrix} 1.5 \\ -0.75 \end{pmatrix}$ y obtenemos que $q_1 = \begin{pmatrix} 0.34 \\ 1.66 \end{pmatrix}$ y $q_2 = \begin{pmatrix} -0.85 \\ 3.99 \end{pmatrix}$. Al igual que para la p anterior, en esta elegimos solo en primer valor de cada q. Luego queda que q tiene que valer:

$$q = \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} = \begin{pmatrix} 0.34 \\ -0.85 \end{pmatrix}$$

File answer_4.m

```
syms q1 q2 dotq1 dotq2

d1 = 1/2;
d2 = 1/2;
m1 = 1;
m2 = 1;
r1 = 1/2;
l1 = 1;
l2 = 1;
I1 = 1;
I2 = 1;

% Para pA
p1 = q1 + r1 + l2*cos(q2) == 1;
p2 = l2*sin(q2) == 0.75;

[x1, x2] = solve([p1 p2], [q1 q2])

vpa(x1) % No puede ser mayor que 1
vpa(x2)

% Para pB
p1 = q1 + r1 + l2*cos(q2) == 1.5;
p2 = l2*sin(q2) == -0.75;

[x1, x2] = solve([p1 p2], [q1 q2])

vpa(x1) % No puede ser mayor que 1
vpa(x2)
```

5) Compute the relation between the joint velocities $\dot{\mathbf{q}}$ and the velocity of the end effector $\dot{\mathbf{p}}$. (Contesta en el informe y sube el código Matlab a Aula Virtual en el fichero answer_5.m)

Como se sabe que $\dot{p} = J(q)\dot{q}$, la relación entre $\dot{p}$ y $\dot{q}$ será la matriz jacobiana $J(q)$.

File answer_5.m

```
syms l1 l2 q1 q2
p1 = q1 + r1 + l2*cos(q2);
p2 = l2*sin(q2);
J = jacobian([p1, p2], [q1, q2])
```

6) Compute the relation between the velocity of the end effector $\dot{\mathbf{p}}$ and the joint velocities $\dot{\mathbf{q}}$. Suppose that the robot manipulator is ready to execute the tracking task, that is, the position of the end effector is $\mathbf{p} = \mathbf{p}_A$. Compute the initial joint velocities to execute this task. (Contesta en el informe y sube el código Matlab a Aula Virtual en el fichero answer_6.m)

Para calcular la relación inversa, simplemente despejamos $\dot{q}$ de la ecuación anterior y queda:

$$J^{-1}(q)\dot{p} = J^{-1}(q)J(q)\dot{q}$$
$$J^{-1}(q)\dot{p} = \dot{q}$$

Luego para poder calcular la $\dot{q}$ nos queda saber cuál es $\dot{p}$. Para ello necesitamos dividir la velocidad del tracking en sus componentes x e y, y el ángulo en el que se mueve. También es importante resaltar que como nos desplazamos hacia abajo debo de restarle π al ángulo para así conseguir el ángulo necesario para dividirla en x e y.

Con todo esto nos queda que:

$$\dot{p} = \begin{pmatrix} 0.0316 \\ -0.0949 \end{pmatrix}$$

Con $\dot{p}$ obtenida y con $J(q)$ del apartado anterior ya podemos resolver $\dot{q}$:

$$\dot{q} = \begin{pmatrix} -0.076369 \\ -0.14374 \end{pmatrix}$$

File `answer_6.m`

```
syms l1 l2 q1 q2 dotq1 dotq2 r1

p1 = q1 + r1 + l2*cos(q2);
p2 = l2*sin(q2);
J = jacobian([p1, p2], [q1, q2])

% Calcular las componentes de la velocidad
ang = atan((1.5 - 1)/(-0.75 - 0.75)) - pi
max_speed = 0.1

dotpx = max_speed*sin(ang)
dotpy = max_speed*cos(ang)

J = subs(J,[q2, l2], [0.85, 1])

vpa(inv(J)*[dotpx;dotpy])
```

7) Compute the time derivative of the Jacobian matrix. (Contesta en el informe y sube el código Matlab a Aula Virtual en el fichero answer_7.m)

Para calcular la derivada respecto al tiempo de la matriz jacobiana J, realizo un cambio de variables para tenerla con respecto a las del espacio de estado, que además las pongo con respecto al tiempo.

Luego con la función diff de Matlab calculo la derivada.

File `answer_7.m`

```
syms l2 x2(t)

J = [1, -l2*sin(x2(t)); 0,  l2*cos(x2(t))];

Jdot = diff(J,t)
```

8) Design a controller based on the feedback linearization method to execute the pick and place task. Write a `Matlab` code that implements the controller to execute the task. Show, plotting the relevant variables and an animation, that the controller satisfies the specifications. (Contesta en el informe y sube el código `Matlab` a Aula Virtual en la carpeta answer_8)

Para esta parte he utilizado el código que habíamos usado para los últimos ejercicios y lo he modificado para que las variables fueran las de este sistema y que este se comportara de forma correcta.

El estado inicial del sistema lo he definido gracias a los resultados de los apartados anteriores (las velocidades son 0, ya que empieza en reposo):

$$x = \begin{pmatrix} -0.16 \\ 0.85 \\ 0 \\ 0 \end{pmatrix}$$

Con respecto a las matrices K_P y K_D, las he inicializado a los siguientes valores:

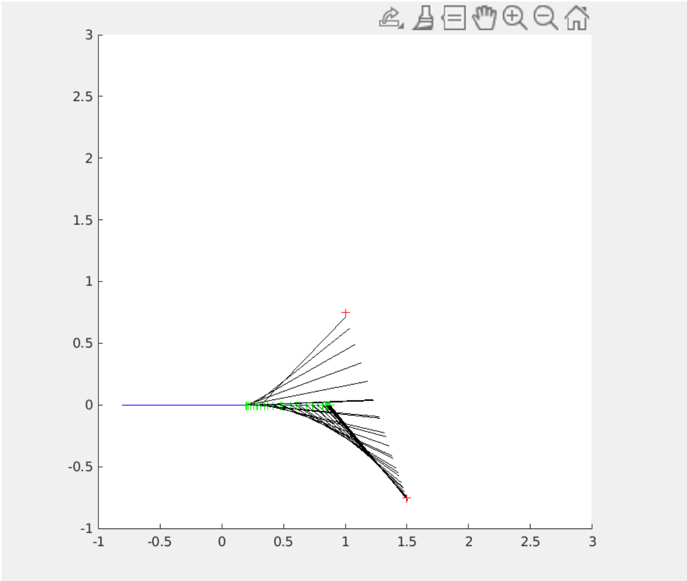
$$K_P = \begin{pmatrix} 10 & 0 \\ 0 & 10 \end{pmatrix} \; K_D = \begin{pmatrix} 5 & 0 \\ 0 & 5 \end{pmatrix}$$

Luego a la hora de ir variando entre p_A y p_B he usado el siguiente código para ir modificando el valor de w dependiendo de la posición del extremo y de las velocidades del sistema:

Modificar w

```
% Es necesario el margen porque sino nunca entra
if (y(1) >= 0.99 && y(1) <= 1.01 && y(2) >= 0.74 && y(2) <= 0.76 && x(3) >= -0.01 &&
    x(3) <= 0.01 && x(4) >= -0.01 && x(4) <= 0.01)
    w = [1.5;-0.75];
elseif (y(1) >= 1.49 && y(1) <= 1.51 && y(2) >= -0.76 && y(2) <= -0.74 && x(3) >= -0.01
        && x(3) <= 0.01 && x(4) >= -0.01 && x(4) <= 0.01)
    w = [1;0.75];
end
```

En cuanto a la representación del sistema he dado al brazo 1 el color azul y al brazo 2 negro. También he puesto el extremo del brazo 1 con una cruz de color verde y al objetivo una cruz de color rojo.



9) Design a controller based on the feedback linearization method to execute the tracking task. Compute the duration of the manoeuvre. Write a `Matlab` code that implements the controller to execute the task. Show, plotting the relevant variables and an animation, that the controller satisfies the specifications. (Contesta en el informe y sube el código `Matlab` a Aula Virtual en la carpeta answer_9)

Esta parte es muy similar al apartado anterior, pero cambia el estado inicial y la w.

Esta vez en el estado inicial si debe tener velocidad inicial, por lo tanto el sistema queda:

$$x = \begin{pmatrix} -0.16 \\ 0.85 \\ -0.076369 \\ -0.14374 \end{pmatrix}$$

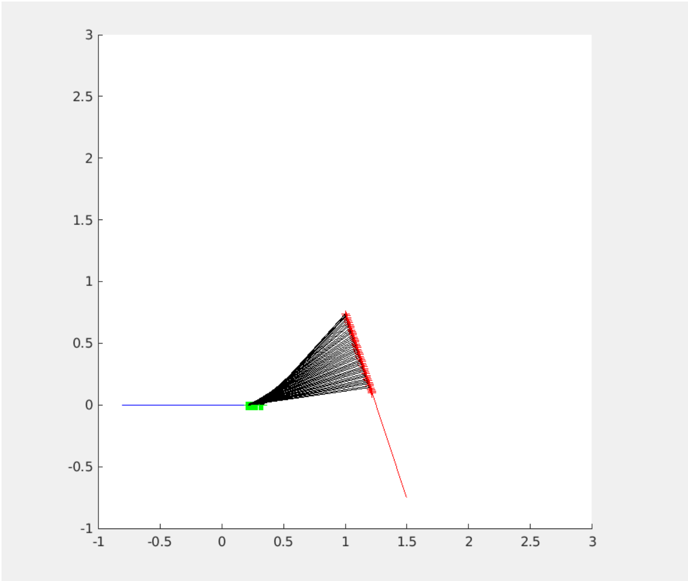
Luego para la w al aplicar la fórmula de un segmento obtengo que la w es

$$w = \begin{pmatrix} 1 - 0.1tsin(ang) \\ 0.75 - 0.1tcos(ang) \end{pmatrix}$$

donde el ángulo es el mismo que en apartado 6, y consecuentemente ahora $\dot{w}$ pasa de ser todo 0 a

$$\dot{w} = \begin{pmatrix} -0.1sin(ang) \\ -0.1cos(ang) \end{pmatrix}$$

Con este controlador conseguimos que el tiempo de la maniobra sea de 15.81 segundos.



3
2.5
2
1.5
1
0.5
0
-0.5
-1
-1
-0.5
0
0.5
1
1.5
2
2.5
3