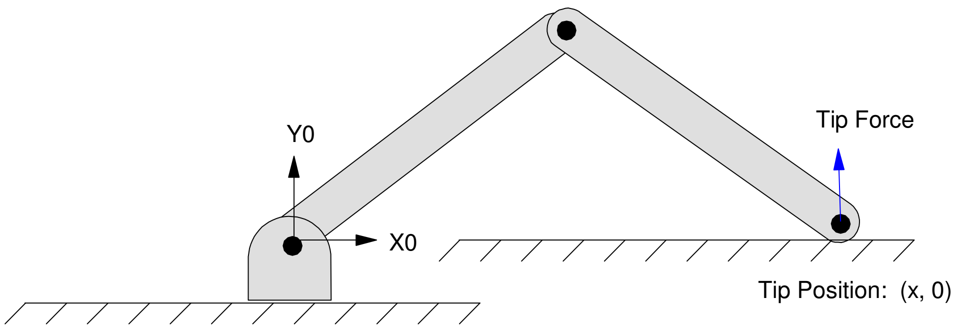
Robots in Contact with the Environment

If the tip is in free space, you can calculate the dynamics using a LaGrangian formulation.

If the tip is anchored to a wall, you can calculate the tip forces using a Jacobian.

How do you simulate a system where

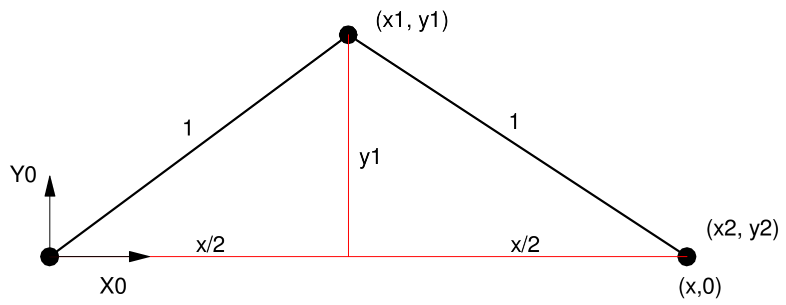
- The motion in one direction is constrained (in contact with a wall) but
- The motion in the other direction is free (the tip can slide up and down)?



Hybrid System: The robot is in contact with a wall (x2 = -1) but motion in the Y direction is free

Robot dynamics when in contact with the environment

If you are constrained to be in contact with the wall at Y=0, then you really only have one degree of freedom. The dynamics can then be reformulated in terms of x. Redrawing the robot in contact with the ground at position x:



you can define the joint positions in terms of joint angle θ_1 noting that

$$\theta_2 = -2\theta_1$$

resulting in

$$x_1 = c_1 \qquad \dot{x}_1 = -s_1\dot{\theta}_1$$

$$y_1 = s_1 \qquad \dot{y}_1 = c_1\dot{\theta}_1$$

$$x_2 = 2c_1 \qquad \dot{x}_2 = -2s_1\dot{\theta}_1$$

$$y_2 = 0 \qquad \dot{y}_2 = 0$$

Setting up the LaGrangian, the kinetic energy is

$$KE = \tfrac{1}{2}\left(\dot{x}_1^2 + \dot{y}_1^2\right) + \tfrac{1}{2}\left(\dot{x}_2^2 + \dot{y}_2^2\right)$$

$$KE = \tfrac{1}{2}\left(\left(-s_1\dot{\theta}_1\right)^2 + \left(c_1\dot{\theta}_1\right)^2\right) + \tfrac{1}{2}\left(\left(-2s_1\dot{\theta}_1\right)^2 + (0)^2\right)$$

$$KE = \tfrac{1}{2}\dot{\theta}_1^2 + 2s_1^2\dot{\theta}_1^2$$

$$KE = \left(\tfrac{1}{2} + 2s_1^2\right)\dot{\theta}_1^2$$

The potential energy is

$$PE = mgy_1 + mgy_2$$

$$PE = gs_1$$

so the LaGrangian is

$$L = KE - PE$$

$$L = \left(\tfrac{1}{2} + 2s_1^2\right)\dot{\theta}_1^2 - gs_1$$

The dynamics are then

$$\frac{\partial L}{\partial \dot{\theta}_1} = 2\left(\tfrac{1}{2} + 2s_1^2\right)\dot{\theta}_1$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_1}\right) = 2\left(\tfrac{1}{2} + 2s_1^2\right)\ddot{\theta}_1 + 4s_1c_1\dot{\theta}_1^2$$

$$\frac{\partial L}{\partial \theta_1} = 4s_1c_1\dot{\theta}_1^2 - gc_1$$

So

$$T_1 = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_1}\right) - \frac{\partial L}{\partial \theta_1}$$

$$T_1 = 2\left(\tfrac{1}{2} + 2s_1^2\right)\ddot{\theta}_1 + 4s_1c_1\dot{\theta}_1^2 - \left(4s_1c_1\dot{\theta}_1^2 - gc_1\right)$$

$$T_1 = 2\left(\tfrac{1}{2} + 2s_1^2\right)\ddot{\theta}_1 + gc_1$$

along with the constraint:

$$\theta_2 = -2\theta_1$$

and hence:

$$\dot{\theta}_2 = -2\dot{\theta}_1$$

$$\ddot{\theta}_2 = -2\ddot{\theta}_1$$

The torque on the second joint from before were:

$$T_2 = \ddot{\theta}_2 + (1 + c_2)\ddot{\theta}_1 + s_2\dot{\theta}_1^2 - gs_{12}$$

Substituting $\theta_2 = -2\theta_1$

$$T_2 = -2\ddot{\theta}_1 + (1 + \cos(2\theta_1))\ddot{\theta}_1 + \sin(2\theta_1)\dot{\theta}_1^2 + gs_1$$

Rotating 90 degrees since zero angle was defined from the x axis (standard) vs y axis (for some reason before)

$$\theta_1 \rightarrow \theta_1 - \tfrac{\pi}{2}$$

$$T_2 = -2\ddot{\theta}_1 + (1 - \cos(2\theta_1))\ddot{\theta}_1 - \sin(2\theta_1)\dot{\theta}_1^2 + gc_1$$

This is the joint torques for no tip force. If there is a tip force, add it to the previous calculations

$$\begin{bmatrix} T_1 \\ T_2 \end{bmatrix} = \begin{bmatrix} 2\left(\tfrac{1}{2} + 2s_1^2\right)\ddot{\theta}_1 + gc_1 \\ (-1 - \cos(2\theta_1))\ddot{\theta}_1 - \sin(2\theta_1)\dot{\theta}_1^2 + gc_1 \end{bmatrix} + (J^T)^{-1}\begin{bmatrix} F_x \\ F_y \end{bmatrix}$$

Note that the dynamics completely change when in contact with a surface:

- When the tip is free, you have two degrees of freedom (θ_1, θ_2) and a 4th-order system to simulate.
- When the tip is constrained in one axis (contact with a wall), you have one degree of freedom and have a 2nd-order system to simulate.
- When the tip is locked to a wall (two constraints), you have a static (zeroth-order) system. There is nothing to simulate here (unless you add an integrator to control the tip forces.)

To control the position in the X direction, you can

- Use a control law to compute the desired acceleration in the x and y direction (i.e. a feedback control law)
- Use a Jacobian to convert these to desired joint accelerations:
- Convert these accelerations to joint torques through the inverse dynamics:

$$\begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} = J^{-1} \begin{bmatrix} \ddot{x}_2 \\ \ddot{y}_2 \end{bmatrix} = J^{-1} \begin{bmatrix} \ddot{x}_2 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} T_1 \\ T_2 \end{bmatrix} = f\left(\theta, \dot{\theta}, \ddot{\theta}\right)$$

To control the tip force, you can use the Jacobian as well

$$\begin{bmatrix} T_1 \\ T_2 \end{bmatrix} = (J^T)^{-1} \begin{bmatrix} F_x \\ F_y \end{bmatrix} = (J^T)^{-1} \begin{bmatrix} 0 \\ F_y \end{bmatrix}$$

although this will be through an integrator:

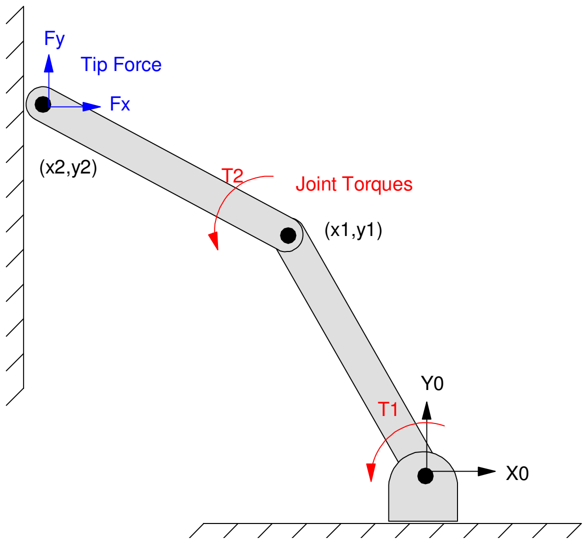
$$\begin{bmatrix} dT_1 \\ dT_2 \end{bmatrix} = (J^T)^{-1} \begin{bmatrix} 0 \\ dF_y \end{bmatrix} = (J^T)^{-1} \begin{bmatrix} 0 \\ k(F_{y:ref} - F_y) \end{bmatrix}$$

$$T_0 = \int dT$$

Tip Forces and Jacobians

So far, we've considered a free robot (i.e. not in contact with anything). Suppose a robot does come in contact with the environment? How do you account for the tip forces?

For this, we'll consider a 2-link manipulator in contact with a wall:



2-Link Arm in Contact with a Wall

Relationship between Tip Forces and Joint Torques

The Jacobian also relates tip forces to joint torques. At any point, the energy must balance: the work done at the tip must equal the work done at the joints:

$$P_{xy} = P_{\theta_{12}}$$

$$\begin{bmatrix} F_x & F_y \end{bmatrix} \begin{bmatrix} \dot{x}_2 \\ \dot{y}_2 \end{bmatrix} = \begin{bmatrix} T_1 & T_2 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$

Replacing the tip forces with the Jacobian:

$$\begin{bmatrix} F_x & F_y \end{bmatrix} \begin{bmatrix} \dot{x}_2 \\ \dot{y}_2 \end{bmatrix} = \begin{bmatrix} T_1 & T_2 \end{bmatrix} J \begin{bmatrix} \dot{x}_2 \\ \dot{y}_2 \end{bmatrix}$$

results in

$$\begin{bmatrix} F_x & F_y \end{bmatrix} = \begin{bmatrix} T_1 & T_2 \end{bmatrix} J$$

or

$$\begin{bmatrix} F_x \\ F_y \end{bmatrix} = J^T \begin{bmatrix} T_1 \\ T_2 \end{bmatrix}$$

The transpose of the Jacobian relates joint torques to tip forces

Example: Determine the tip forces if the joint torques are:

$$\begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.6 \end{bmatrix} \qquad \text{radians}$$

$$\begin{bmatrix} T_1 \\ T_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \qquad \text{Nm}$$

In Matlab:

```
>> J = [-c1-c12,  -c12;   -s1-s12,  -s12]

   -1.3312   -0.4536
   -1.3706   -0.8912

>> T = [1;2]

      1
      2

>> F = J' * T

   -4.0724           Tip force Fx
   -2.2360           Tip force Fy
```

Determine the joint torques if the tip force is

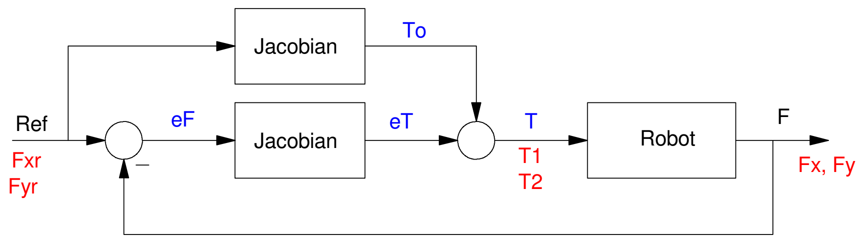
$$\begin{bmatrix} F_x \\ F_y \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \end{bmatrix} \qquad \text{N}$$

```
>> F = [0;2]

      0
      2

>> T = inv(J')*F

    4.8549     T1     (Nm)
   -4.7151     T2     (Nm)
```

Force Control:

Assume a robot is in contact with the wall in such a way that the tip will not slip. Design a control law to regulate the tip force.

One approach is to use the Jacobian: if you know the desired tip forces, you can calculate the joint torques. If there is an error in the resulting joint torques, adjust the torque based upon this error and the Jacobian matrix:

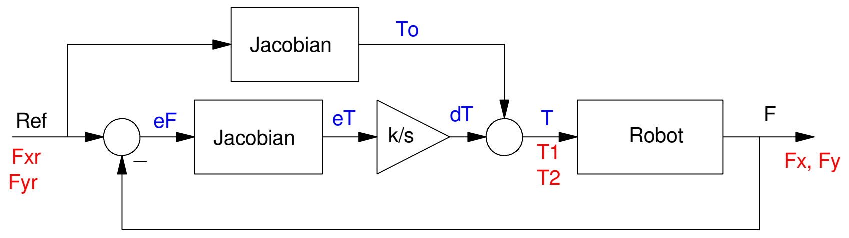


One approach to controlling tip forces. This produces an algebraic loop, however.

The problem with this approach is there is an algebraic loop:

- The joint torques produce a tip force.
- Which through the feedback loop changes the joint torques

To break this loop, add an integrator.



An integrator is added to break the algebraic loop.

The integrator is essentially 'searching' to find the constant which forces its input to zero (the error in the joint torques or the error in the tip forces.) By adjusting the gain, k, the speed at which the torques zero in can be set with

$$F_{xy} = \left(\frac{k}{s+k}\right) R_{F_{xy}}$$

The 2% settling time should be approximately 4/k seconds.

Simulation Results:

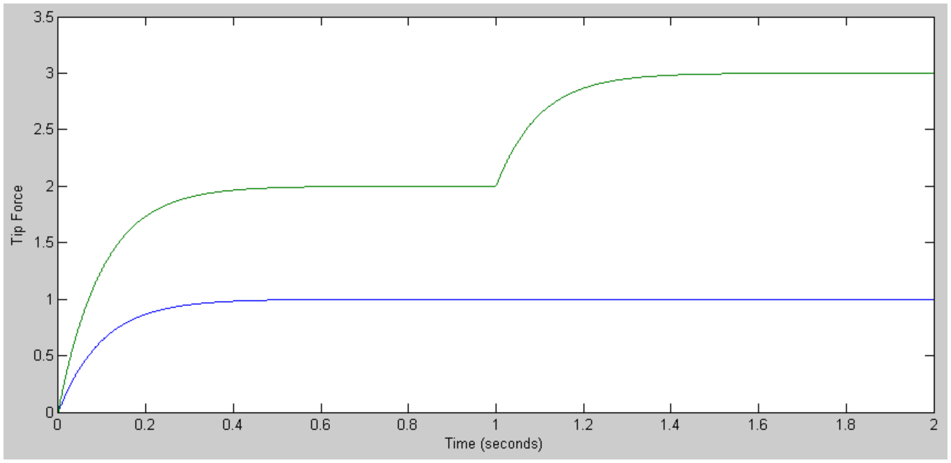
Setting the integrator gain to 10 and leaving out the feedforward term so you can see the integrator effect, the resulting tip force for the robot held at position

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0.3 \end{bmatrix}$$

with the set point (Ref) being

$$\begin{bmatrix} F_x \\ F_y \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} N$$

is as follows. Here, the feedforward term is left out so you can see the effect of the integrator (it adjusts the torque to compensate for any error in the computed torques from the feedforward term). The integration gain, k, was set to 10 as well, resulting in a settling time of 4/10 second.



Resulting Tip Forces for a Set Point of [1, 2]N for t<1. [1, 3]N for t>1

Note that

- The tip force behaves as a 1st-order system (the only dynamics in the loop is the integrator)
- There is no coupling between the tip force in the X or Y direction due to the use of the Jacobian

Also note that the settling time can be made very fast by using higher gains. Force control can be very quick.

Matlab Code:

```
% Force Control

% Tip Position
Tip = [ 1.0 ;  0.0 ];

% Joint Angles
Q = InverseRR(Tip)

% Desired Tip Force
Ref = [1 ; 2];

% Compute the Jacobian
J = [-sin(Q(1)) - sin(Q(1)+Q(2)),  -sin(Q(1)+Q(2)) ; cos(Q(1)) + cos(Q(1)+Q(2)),
cos(Q(1)+Q(2)) ];

% Initial Values
T = [0; 0];
t = 0;
dt = 0.001;

% plotting data
Fref = [];
Ftip = [];

while(t < 2)

% Compute the tip forces given the joint torques
F2 = J' * T;

% the error in the tip forces
dF = Ref - F2;

% convert to the error in the joint torques
dT = inv(J')*dF;

% integrate with a gain of 10
T = T + 10*dT*dt;
t = t + dt;

% Save the tip forces for plotting

Fref = [Fref, Ref];
Ftip = [Ftip, F2];

end

t = [1:length(Ftip)] * dt;
plot(t,Fref,t,Ftip);
xlabel('Time (seconds)');
ylabel('Tip Force (N)');
```