Midterm Project—Creating the Trajectory

Rather than having you spending time trying to create the trajectory—better to spend time on the workcell kinematics, *etc.* So here are some tips on creating the trajectory in MATLAB.

1 Parametric Trajectory

1.1 Cubic Parametric Equations

With the following definitions:

$$a = 0.0625 \text{ meters}$$
$$b = 0.25 \text{ meters}$$
$$t = 0 \dots 4\pi \text{ sec}$$

the motion of the center of the workpiece (*e.g.* the 4^{th} column of ${}^{S}_{obj}\mathbf{T}$) is given by the *parametric* equations

$$\gamma(t) = \left[\frac{3}{8\pi}\right] t^2 - \left[\frac{1}{16\pi^2}\right] t^3 = p_2 t^2 + p_3 t^3 \text{ rad} \tag{1}$$

$$x = (b-a)\sin\gamma(t) + 3a\sin\left(\frac{\gamma(t)}{a}(b-a)\right) \text{ meters} \tag{2}$$

$$y = 0 \text{ meters} \tag{3}$$

$$z = (b-a)\cos\gamma(t) - 3a\cos\left(\frac{\gamma(t)}{a}(b-a)\right) + 0.5 \text{ meters} \tag{4}$$

in which the time duration of the trajectory is 4π (really 12.576 seconds). Variable $\gamma(t)$ is the *parameter* (it's really a variable along the path) and is given as a *cubic* function of time. The reasons for using a cubic will be discussed in Chapter 7. Note that I expressed (1) using symbolic coefficients p_2 and p_3 in addition to the numerical form (coefficients p_0 and p_1 are both zero for this rest-to-rest motion).

1.2 Trajectory Generation in MATLAB

Here are some useful MATLAB commands for generating this trajectory for subsequent use in calculating the robot joint angles.

I'll begin by creating a time vector from $0 < t \leq 6.288$ seconds with time step of 0.016 seconds. Then use that to calculate the entire sequence $\gamma(t)$ using equation (1), and finally obtain $x(t)$ and $z(t)$ using (2) and (4), respectively.

First create the symbolic coefficients of the cubic polynomial for $\gamma(t)$:

```
>> p2 = 3/8*pi);      % Coefficients of the cubic
>> p3 = -1/(16*pi^2);   % polynomial for gamma(t).
```

Next create a time vector from 0 to 5 seconds with spacing 0.016 seconds:

```
>> ti = 0;            % ti is the initial time (zero)
>> tf = 12.576;       % tf is the final time (4*pi seconds)
>> dt = 0.016;        % dt is the time step (0.016 seconds)
>> t = [ti:dt:tf]';   % Create time vector as column vector (take transpose)
```

Now use the time vector to evaluate the cubic polynomial. This is most easily done with the MATLAB `y = polyval(p,x)` function. Vector `p` is the polynomial coefficients in decreasing order, while vector `x` is the independent variable (here it is time).

```
>> gamma = polyval([p3 p2 0 0],t);    % Evaluate cubic polynomial for gamma
```

Finally use $\gamma(t)$ to find $x(t)$ and $z(t)$:

```
>> a = 0.0625;  % Parameter a in meters
>> b = 0.25;  % Parameter b in meters
>> x = (b-a)*sin(t)+3*a*sin((t/a)*(b-a));  % x coordinate in meters
>> z = 0.5+(b-a)*cos(t)-3*a*cos((t/a)*(b-a));   % z coordinate in meters
```

Of course the computation of x and z will be done inside a loop, so $x =$`x(i)`, $t =$`t(i)` *etc.*

1.3 Trajectory Plots

Below is a plot of $\gamma(t)$ *vs* time t. The gentle ramping at each end is the reason we use a cubic.

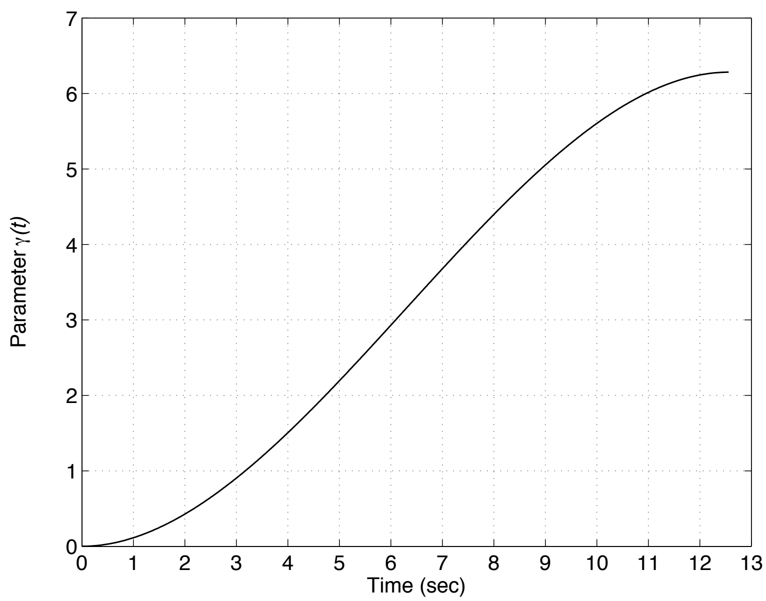

Figure 1: Plot of parameter $\gamma(t)$ for Midterm Project.

This plot was produced using

```
>> plot(t,gamma),grid;
```

You can have multiple MATLAB instructions on the same line, separated by commas (or semicolons). The labels were added afterwards.

Obviously you can plot $x(t)$ and $z(t)$ in the same manner (as well as a "spatial" plot of $x(t)$ *vs* $z(t)$ to see the cloverleaf); also $d_1(t)$, $\theta_2(t)$, and $\theta_3(t)$ as required for the project.