1 Trajectory Generation

`The material for these notes has been adopted from:`
`John J. Craig: Robotics: Mechanics and Control.`

This example assumes that we have a starting position and goal pose of the end effector and we are asked to move the joint angles to move the end effector from one pose to another. Here we describe a strategy how to do so by designing a trajectory in joint space from one end point to another. Assuming that we know the inverse kinematics of the system, we can compute the desired joint angle for goal position of the end effector. This example shows how to design a trajectory of a single joint $\theta(t)$ as function of time. Suppose that we have following constrains of out trajectory: we have desired position at the beginning and end of the trajectory and we the velocity at the begining and end has to be zero. Hence our desired trajectory has to satisfy the following constraints:

$$\theta(0) = \theta_0;\ \ \theta(t_f) = \theta_d\ \ \dot{\theta}(0) = 0\ \ \dot{\theta}(t_f) = \theta_d \tag{1}$$

Cubic polynomials In order to satisfy the above constraints, our trajectory has to be at least polynomial of the 3^{rd} order, which has four coefficients, and hence can satisfy the above 4 constraints. This can be achieved by third order cubic polynomial which has the following form

$$\theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$

Given the above form the joint velocity and acceleration will have the following forms

$$\begin{aligned} \dot{\theta} &= a_1 + 2a_2 t + 3a_3 t^2 && (2)\\ \ddot{\theta} &= 2a_2 + 6a_3 t && (3) \end{aligned}$$

Using the above equations and instantiating the constraints we can solve for the coefficients of the cubic polynomial and obtain

$$
\begin{aligned}
a_0 &= \theta_0 && (4)\\
a_1 &= 0 && (5)\\
a_2 &= \frac{3}{t_f^2}(\theta_f - \theta_0) && (6)\\
a_3 &= \frac{2}{t_f^3}(\theta_f - \theta_0) && (7)
\end{aligned}
$$

Now given a particular instance of the problem, we can substitute to the above equations the desired parameters θ_0, θ_f, t_f and obtain different trajectories.

Linear functions with parabolic blends If we were to simply just connect the desired position with a linear function, it would cause the velocity to be discontinuous at the beginning and end of the motion. Also note that of the shape of the part in the joint space is linear, that does not mean that the shape of the path in the end effector space is linear. Hence what can be done is to take a linear path in the end effector space and interpolate it linearly. We would like to do it in a way that the velocities at the would not be discontinuous at the places where the pieces meet. One way to achieve this is to add a parabolic blend region, such that the we will create a smooth and continuous path. During the blend portion of the trajectory the acceleration will be constant (i.e. we will assume that it will not be changing in time and that we can instantaneously generate the constant acceleration profile). To construct a one such single segment, we will assume that parabolic blend at the beginning and the end have the same duration and the same constant acceleration (with opposite signs) will be used during those blends. If the blends at the beginning and the end will have the same duration, the final solution will be always symmetric around the half way point t_h and θ_h. To guarantee smoothness the velocity at the end of the blend has to be the same as the velocity of the linear section

$$\ddot{\theta} t_b = \frac{\theta_h - \theta_b}{t_h - t_b}$$

where θ_b is the value of θ at the end of the blend region. Since the blend is parabolic the value of θ_b is given by

$$\theta_b = \theta_0 + \frac{1}{2}\ddot{\theta} t_b^2$$

Combining the above two equations and denoting $t = 2t_h$, we get

$$\ddot{\theta} t_b^2 - \ddot{\theta} t t_b + (\theta_f - \theta_0) = 0$$

where t is the desired time of motion. Given the desired θ_f, θ_0 and t, the above equation gives is constraints on between $\ddot{\theta}$ and t_b which the trajectory has to satisfy. Hence typically $\ddot{\theta}$ is chosen and then we can use the equation to solve for t_b to obtain

$$t_b = \frac{t}{2} - \frac{\sqrt{\ddot{\theta} t^2 - 4\ddot{\theta}(\theta_f - \theta_0)}}{2\ddot{\theta}}$$

Notice that depending on acceleration the time of the blend region will vary. Depending on the acceleration, the path will be composed from two parabolic blends which will meet in the middle with the same slope and the linear portion of the blend will go to zero. If the acceleration is high the blend region will be shorter. In the limit when acceleration is infinite, we will reach the simple linear interpolation case.

2 Control of Second-Order Systems

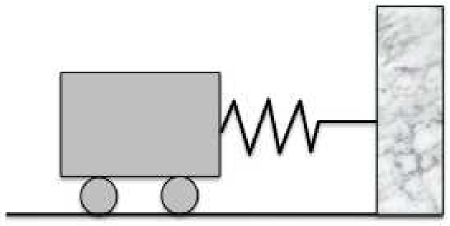

Figure 1: Block with mass m attached to the wall with spring with stiffness k.

Before we start considering the trajectory tracking problem, lets consider a simpler problem. Consider a block with the mass m sliding along a surface and attached with the spring to the wall. The equation of motion of the block is

$$m\ddot{x} + b\dot{x} + kx = 0$$

where x is the position the block (distance to the wall), the $b\dot{x}$ is the frictional force proportional to the velocity and kx is the related to the position and stiffness of the spring. We would like to study the behavior of the system by understanding the trajectories $x(t)$. From the study of differential equations the form of the solution depends on he roots of its characteristic equation

$$ms^2 + bs + k = 0$$

with the roots

$$s_1 = -\frac{b}{2m} + \frac{\sqrt{b^2 = -4mk}}{2m} \text{ and } s_1 = -\frac{b}{2m} - \frac{\sqrt{b^2 = -4mk}}{2m}$$

It can be easily shown by substitution that the solution $x(t)$ has the following form

$$x(t) = c_1 e^{s_1 t} + c_2 e^{s_2 t}$$

where c_1 and c_2 are constants which can be determined from the initial conditions. We will now show 3 different cases of qualitatively different solutions which depend of the values s_1 and s_2 and consequently of the parameters of the system m, b, k.

1. The first case we consider is $s_1 = -2$ and $s_2 = -3$, where two roots are real and have negative parts. In case the initial conditions, $x(0) = -1$ and $\dot{x}(0) = 0$, substituting to the differential equation

$$\begin{aligned} c_1 + c_2 &= 0 & (8)\\ -2c_1 - 3c_2 &= 0 & (9) \end{aligned}$$

   which is satisfied by $c_1 = -3$ and $c_2 = 2$. The motion of the system is then

$$x(t) = -3e^{-2t} + 2e^{-3t}$$

   The trajectory of the system is plotted in Figure 2.

2. The second case we consider is when the two roots have complex roots and the solution has the form

$$x(t) = c_1 e^{s_1 t} + c_2 e^{s_2 t}$$

   where $s_1 = \lambda + i\mu$ and $s_2 = \lambda - i\mu$. Using the well known **Euler formula**

$$e^{ix} = \cos x + i \sin x$$

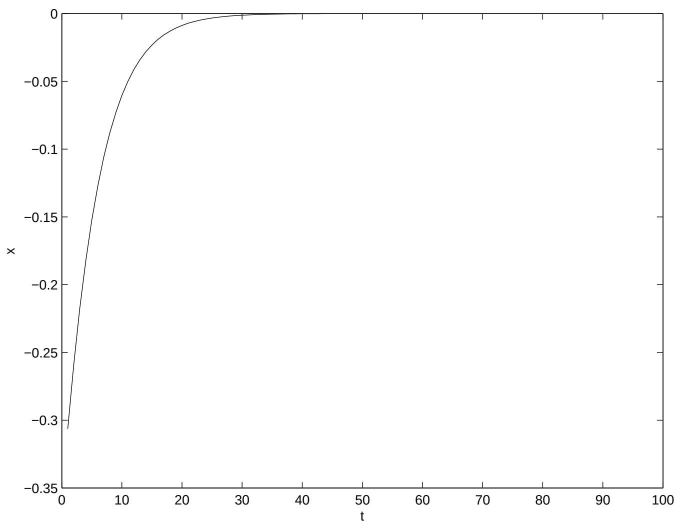

Figure 2: Case 1 $x(t)$.

we can rewrite the trajectory in the following form

$$x(t) = c_1 e^{\lambda t}\cos(\mu t) + c_2 e^{\lambda t}\sin(\mu t)$$

where the coefficients c_1 and c_2 can be computed from initial conditions. If we rewrite them in the following way

$$c_1 = r\cos\delta \qquad (10)$$

$$c_2 = r\sin\delta \qquad (11)$$

then using the formula for $cos(\alpha+\beta)$ we can write the trajectories in the following way

$$x(t) = re^{\lambda t}\cos(\mu t - \delta)$$

where

$$r = \sqrt{c_1^2 + c_2^2} \text{ and } \delta = \arctan(c_2, c_1)$$

In the above form is is easier to see that the resulting trajectories will be oscillations, with the amplitude exponentially decreasing to zero. This type of oscillatory system is also often described in terms of following parameters, which are the functions of the terms already defined above. First it is the

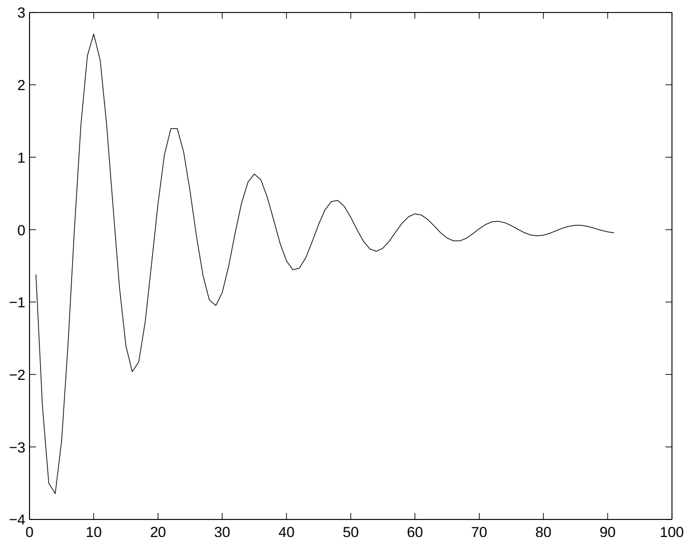

Figure 3: Case 2 $x(t)$.

natural frequency of the system ω_n, the damping ratio ζ

$$\lambda = -\zeta\omega_n \tag{12}$$

$$\mu = \omega_n\sqrt{1-\zeta^2} \tag{13}$$

These symbols are related to the canonical form of the characteristic equation of the second order system

$$s^2 + \zeta\omega_n s + \omega_n^2 = 0$$

3. Another interesting case is the case when, the solutions to the characteristic equations are two real repeated roots, i.e.

$$s_1 = s_2 = -\frac{b}{2m}$$

In this case the trajectory will have the following form

$$x(t) = (c_1 + c_2)e^{\frac{-b}{2m}t}$$

When the roots of the characteristics equations (also called poles of the second order system) are real and equal, the system is **critically damped** and exhibits the fastest possible non-oscilatory response.

2.1 Control of second order systems

We saw in the previous section that the behavior of the second order system (involving second derivatives of the position) depends on the coefficient of the system. If we want to achieve a desired behavior we need to modified these coefficients by means of control. Suppose for example that we are going to apply some external force to the system, which will yield the following equation of motion

$$m\ddot{x} + b\dot{x} + kx = f \tag{14}$$

Assuming that we have at our disposal sensors which can measure the position x and the velocity $\dot{x}$, we would like to make the force proportional to the sense feedback. Hence suppose the control of the following form

$$f = -k_p x - k_v \dot{x} \tag{15}$$

where k_p and k_v are some constants, also referred to as gains determining how big the force will be as proportion of velocity and position. This particular control law will strive to keep the position of the block at zero and stationary, i.e. when both $x = 0$ and $\dot{x} = 0$, the applied force will be 0. If we now bring the equation of motion to the canonical form above (right hand side is zero), we will have an equation of motion of closed feedback loop system

$$m\ddot{x} + (b + k_v)\dot{x} + (k + k_p)x = 0 \tag{16}$$

or

$$m\ddot{x} + b'\dot{x} + k'x = 0 \tag{17}$$

Notice now that we can now chance the control gains k_p and k_v so as to obtain the coefficients of the second order system which would generate the desired behavior.

2.2 Trajectory following

So how is this related to the trajectory following? Well instead of designing a control to maintain the block a a particular position, we can design a control which will make the block to follow particular trajectory. Suppose now that the trajectory is given to us as $x_d(t)$ which specifies the desired position of the block. We also assume that out trajectory is smooth (i.e. first two derivatives exist) and that our trajectory generator provides us with $x_d, \dot{x}_d$ and $\ddot{x}_d$ at all times. We now define the servo error as $e = x_d - x$. A servo control law which we will then use for trajectory following will have the following form

$$f = \ddot{x}_d + k_v \dot{e} + k_p e \tag{18}$$

The block diagram of such feedback control law is in Figure 4.

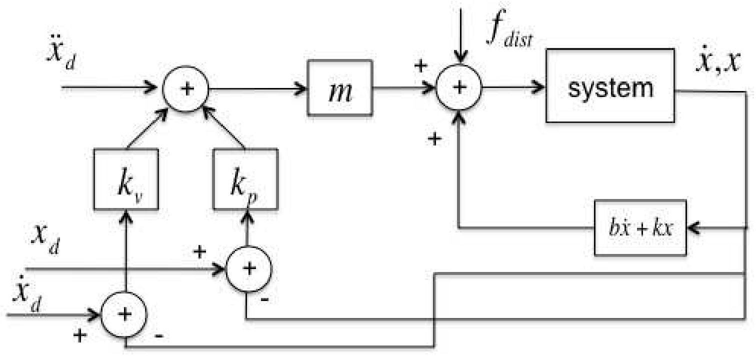

Figure 4: Feedback control diagram for trajectory following.

If we combine the above equation with the simplified canonical equation of motion $\ddot{x} = f$ [1] we will obtain the following equation:

$$\ddot{x} = \ddot{x}_d + k_v \dot{e} + k_p e \tag{19}$$

or

$$\ddot{e} + k_v \dot{e} + k_p e = 0 \tag{20}$$

where $\ddot{e} = \ddot{x}_d - \ddot{x}$. Notice that this again second order differential equation, hence we can determine the behavior of the error trajectories $e(t)$ by setting the coefficients of the equation, based on the cases outlined at the beginning of this handout. This equation captures the behaviour of the system in the error space. If our model is perfect (we know exactly m, b and k) the controller will follow the trajectory perfectly. In practice two things often happen: one is that the knowledge of our model is not perfect and that our system can be affected by some external disturbances. If the presence of external disturbances the closed loop behavior of our system will have the following form:

$$\ddot{e} + k_v \dot{e} + k_p e = f_{dist} \tag{21}$$

[1] Any second order system can be rewritten to this form by simply grouping the other parameters of the system into f.

where f_{dist}. The simplest kind of disturbance is when the f_{dist} is constant. In such case when the system is in steady-stage (i.e. all derivatives are zero), we will have

$$k_p e = f_{dist}$$

i.e. the value of the steady state error will be $e = \frac{f_d ist}{k_p}$. From here we can see that higher the gain k_p two lower the error will be. In order to eliminate the steady-state error, the control law is typically modified to incorporate so called intergral term and the control law then becomes

$$f' = \ddot{x} + k_v \dot{e} + k_p e + k_i \int e dt$$

This resulting control law is ofter called **PID** control law which stands for ”proportional, intergral, derivative” control law. It is one of the most commonly used control strategies applicable to a large variety of problems.

3 Mobile Robots

Control of mobile robots topics covered in class:

- pose to point control of differential drive robot (slides)
- pose-to-pose control of differential drive robot in the polar coordinates (slides)
- line following

4 Time-varying coordinates of Rigid Body

Here we will make a slight digression from the control discussion and return to the issue of generating trajectories. In the previous examples we shown how to generate trajectories from a single joint angle, assuming some initial and final conditions and some constraints on the velocities. We have shown that if we want to motion to be smooth (no velocity discontinuities), we need to specify the trajectory as a polynomial. Another strategy we had was the generating the trajectories as linear segments with parabolic blends, which generated so called trapezoidal velocity profiles and enable the joing variable to move at maximal (on some constant) velocity during pre-specified time.

We will try to generalize these ideas for genera rigid bodies, which have 6 degrees of freedom. Suppose that you have initial pose $g_0 = (R_0, T_0)$ and you want to generate trajectory which interpolates between initial and final pose $g_1 = (R_1, T_1)$. The idea of interpolation is the following to generate intermediate poses it any instance of time

$$g(t) = (1-s)g_0 + sg_1$$

where $s \in [0,1]$. Note at the beginning where $s = 0$ the pose is g_0 and at the end $s = 1$ we will have g_1. If g_0 is a rigid body pose which is represented by a matrix, we cannot simply multiply the matrix by s as the resulting matrix would not longer represent a rigid body pose and it is part corresponding to the rotation matrix which is problematic. This brings about the issues of parametrization which we discussed previously. Given that any rotation can be express using its exponential coordinates ω as $R = e^{\hat{\omega}}t$ we can just interpolate between ω_0 and ω_1, while the translation component can be interpolated linearly.

In situations where we want the 3D rigid body follow a path, which is specified in terms of the waypoints, we need to again worry about smooth transitions between waypoints and use higher order polynomial (as opposed just line) for interpolation. The common choice of such polynomials are so called splines, which are specified in terms of their waypoints, although the final curve may not pass through them.