Chapter 5

PATH CONTROL

Dr. Mayez Al-Mouhamed
Professor, Computer Engineeing Department
King Fahd University of petroleum and Minerals

Path control refers to the generation of analytical trajectory passing by an arbitrary number of 6-dimensional points. The objective of this chapter is to present two simple methods for trajectory generation. The first approach consists of time continuous polynomials. The second approach consists of discrete and exact laws. Both methods have interesting characteristics such as continuity of position, velocity, and acceleration. These methods allows defining complex motion functions involving an arbitrary number of via points. The motivation for this problem is to avoid collision between the robot and obstacles by generating a finite number of representative points. subsectionIntroduction Path control refers to the generation of an analytical trajectory passing by an arbitrary number of 6-dimensional points. These trajectory are generated to (1) avoid collision between the robot arm and the various obstacles that may exist in the robot work space, and (2) increasing the motion speed and continuity by removing unnecessary acceleration and deceleration. Collisions may damage both the robot arm and these obstacles. To solve this problem we assume the system is capable of finding a set of representative points, in the work space, whose trajectory represents a candidate path that is collision free.

Generation of representative points of a collision free path can be achieved by using object modelling or visual sensing or both methods.

In any case, the required trajectory is to satisfy a number of constraints that can be summarized in the following:

1. Fast and global motion from an initial point E_I to a final point E_f ,
2. Initial and final velocities should be null,
3. Moderate computational complexity to enable on-line evaluation.
4. Exact convergence for every point of the trajectory.

5.1 Polynomial trajectories

Consider the problem of generating a trajectory starting with a point $P(t=0) = P_0$ and ending with a point $P(t=t_f) = P_f$ and assume both starting and ending velocities $V(t=0) = V_0$ and $V(t=t_f) = V_f$ are to be null. As both velocities are null, the required trajectory is to start from a stopping point (P_0, V_0) and should terminate at another stopping point (P_f, V_f) . These four constraints can be satisfied by using a third order polynomial $P(t)$:

$$P(t) = a_o + a_1 t + a_2 t^2 + a_3 t^3 \tag{5.1}$$

The required polynomial should satisfy the following constraints:

$$\begin{aligned} P(t=0) &= a_0 \\ V(t=0) &= a_1 = 0 \\ P(t=t_f) &= a_0 + a_2 t_f^2 + a_3 t_f^3 = P_f \\ V(t=t_f) &= 2a_2 t_f + 3a_3 t_f^2 = V_f = 0 \end{aligned} \tag{5.2}$$

This leads to the following solution:

$$P(t) = P_0 + \frac{3}{t_f^2}(P_f - P_0)\, t^2 - \frac{2}{t_f^3}(P_f - P_I)\, t^3 \tag{5.3}$$

For

$$t \in [0, t_f]$$

Now, assume a set of representative points $\{E(t_1), E(t_2), ..., E(t_n)\}$ each point $E(t_i)$ for $(i = 1, ..., n)$ consists of a vector in the 6-dimensional space, i.e. vector $E \in R^6$. The vector E denotes the position and orientation of the tool frame of reference that has been discussed in Chapter 3. The vector E could also be considered as a geometric reference for the robot hand.

For convenience, let us assume the following notation for vector E:

$$E(t) = (X_1(t), X_2(t), ..., X_6(t))^t \tag{5.4}$$

The required trajectory should then be decomposed with respect to each scalar component $X_k (k = 1, ..., 6)$ and should satisfy the constraints of X_k. If we partition the original set of points $\{E(t_i)\}$ we obtain $n-1$ segments S_i as follows:

$$S_i = \{E(t_i), E(t_i + 1)\}\; i = 1, ..., n-1 \tag{5.5}$$

If one uses the polynomial function, which is given in Equation 8.3, then each starting or ending point of a segment will be a stopping point. This leads to slow down the motion because of multiple starting and ending both with with zero velocity. To avoid this shortcoming, one needs to find the value of the intermediate velocities between the segments and generate trajectories for each segment such that the mapping of position and velocity is satisfied from a segment to another. If this condition is applied, the trajectory will be continuous function with respect to position and velocity.

To find the trajectory that map positions and velocities, one needs six third-order polynomials for each segment $S_i (i = 1, .., n)$. In other words, each scalar component X_i of vector

E needs to be assigned a polynomial function $P(t)$ to describe its temporal variation along one segment of the trajectory:

$$P(t) = a_o + a_1 t + a_2 t^2 + a_3 t^3 \quad (5.6)$$

where the coefficients a_0, a_1, a_2, and a_3 are to be determined by using the starting and ending points of each segment, i.e. the boundary conditions. In general, the boundary conditions of a segment can be stated by means of initial and final positions and velocities:

Initial position : $P(t=0) = P_0$
Initial velocity : $P^{'}(t=0) = V_0$
Final position : $P(t_f) = P_f$
Final velocity : $P^{'}(t_f) = V_f$

where P_0, V_0, P_f , and V_f denote the position and velocity at the starting and ending points of the segment. It is clear that the position is simply mapped between segments S_{i-1} and S_i by means of specific scalar components of vector E_i which is the end point of segment S_{i-1} and also the starting point of the next segment S_i .

5.1.1 Finding the boundary conditions

In this section, we investigate methods for finding the velocity and acceleration given a set of position points. First, we study a heuristic method to find the velocity given three position points. Second, we approximate a set of three points by means of a third order polynomial for which the second derivative will be evaluated.

To map the velocity one needs to find a heuristic approach for evaluating the velocity at the end of each segment. For the first and last segments S_1 and S_n, the starting and ending velocities are to be null, respectively. This means that the first and last points of the trajectory should be stopping points. For the intermediate segments, one may evaluate the velocity as follows:

$$X_k^{'}(t_i) = \frac{X_k(t_{i+1}) - X_k(t_{i-1})}{t_{i+1} - t_{i-1}} \quad (5.7)$$

This equation provides the velocity information which will be used for mapping the segments $S_i - 1$ and S_i that share point E_i. The acceleration is useful for generating smooth trajectories. Initially, we assume a set of position points is given. Let us consider three points $P(t_1), P(t_2)$, and $P(t_3)$ that represent the position at different value of time. The problem is to determine a heuristic for finding the second order time derivative at point $t = t_2$, i.e. the function $P^{"}(t_2)$. Let us consider that the value of $P^{"}(t_2)$ is that of a second order polynomial that passes by the points $P(t_1), P(t_2)$, and $P(t_3)$. In this case, the time polynomial function associated to $P(t)$ will be:

$$\begin{aligned} p(t_1) &= at_1^2 + bt_1 + c \\ p(t_2) &= at_2^2 + bt_2 + c \\ p(t_3) &= at_3^2 + bt_3 + c \end{aligned}$$

Solving this system Equations allow finding the second time derivative of $P(t)$ as follows:

$$P^{"}(t_2) = 2a = \frac{2}{t_3 - t_1}\left(\frac{P(t_3) - P(t_2)}{t_3 - t_2} - \frac{P(t_2) - P(t_1)}{t_2 - t_1}\right)$$

The knowledge of scalar $X_i(t_k - 1)$, $X_i(t_k)$, and $X_i(t_k + 1)$ allows finding the acceleration $X_i^{"}(t_k)$ at point $X_i(t_k)$ by using this heuristic. In the following we assume that each segment of the trajectory is specified by means of position, velocity, and acceleration at the starting and ending times. This allows finding smooth paths that ensure continuity of the trajectory with respect to position, velocity, and acceleration.

5.1.2 Mapping position and velocity

The boundary conditions specify the position, and velocity at both the starting time $t = 0|$ and the ending time $t = t_f$. The boundary conditions are as follows:

Initial position : $P(t = 0) = P_0$
Initial velocity : $P^{'}(t = 0) = V_0$
Final position : $P(t_f) = P_f$
Final velocity : $P^{'}(t_f) = V_f$

The polynomial function $P(t)$ should be at least of degree three in order to satisfy the set of four constraints as stated above. The minimum time polynomial function P(t) is then as follows:

$$\begin{aligned} P(t) &= a_0 + a_1 t + a_2 t^2 + a_3 t^3 \\ P^{'}(t) &= a_1 + 2a_2 t + 3a_3 t^2 \end{aligned}$$

where $P(t)$ and $P^{'}(t)$ denote the position and the velocity, respectively. As a result of the mapping, the position and the velocity will be continuous functions from a segment to another along the whole path which is defined by the set of vectors $\{E_0, E_1, ..., E_n\}$. Each segment requires the computation of six polynomials because each of these polynomials will be assigned to a scalar variable $X_i(i : 1, ..., 6)$ of vector E.

The writing of the boundary conditions at the starting time ($t = 0$) of each segment allows finding the first two coefficients:

$$\begin{aligned} P(t_f = 0) &= P_0 = a_0 \\ P^{'}(t = 0) &= V_0 = a_1 \end{aligned}$$

This leads to the following expressions at the ending time t_f:

$$\begin{aligned} P(t_f) &= P_0 + V_0 t_f + a_2 t_f^2 + a_3 t_f^3 = P_f \\ P^{'}(t_f) &= V_0 + 2a_2 t_f + 3a_3 t_f^2 = V_f \end{aligned}$$

Let denote by ΔP and ΔV the variation in the position and velocity within the same segment:

$$\begin{aligned} \Delta P &= P_f - P_0 \\ \Delta V &= V_f - V_0 \end{aligned}$$

The last two Equations allow writing:

$$\begin{aligned} a_2 + a_3 t_f &= \frac{1}{t_f^2} [\Delta P - V_0 t_f] \\ 2a_2 + 3a_3 t_f &= \frac{\Delta V}{t_f} \end{aligned}$$

Solving the system equations, we obtain solution for a_2 and a_3 , as follows:

$$\begin{aligned} a_2 &= \frac{1}{t^2}\left[3\left(\Delta P - V_0 t_f\right) - \Delta V t_f\right] \\ a_3 &= \frac{1}{t^3}\left[\Delta V t_f - 2\Delta P + 2V_0 t_f\right] \end{aligned}$$

Let $t \in [0, t_f]$ be the time which can be written as function of the sampling period Δt and the number of intermediate points within each segment:

$$t = i.\Delta t \quad i = 1, ..., N \text{ and } t_f = N.\Delta t$$

For each segment and for each scalar component of vector E, the path will be defined by using the following polynomial function:

$$P(i) = a_0 + (a_1\Delta t)i + (a_2\Delta t^2)i^2 + (a_3\Delta t^3)i^3$$

where the parameters a_0 , $a_1\Delta t$, $a_2\Delta t^2$, $a_3\Delta t^3$ are to be evaluated once at the starting of each new segment. Evaluation of these parameters for one scalar variable requires 6 additions and 12 multiplications. To generate one new point of the trajectory, one needs 3 additions and 5 multiplications for each scalar component.

5.1.3 Mapping position, velocity, and acceleration

In this section we study the design of a polynomial function to generate paths between two points whose boundary conditions includes position, velocity, and acceleration. These conditions imply that the mapping is to be achieved with respect to position, velocity, and acceleration.

The starting time of the trajectory is $t = 0$ and the ending time is $t = t_f$. Denote by $P(t)$ the time polynomial function which represents the position function and $P^{'}(t)$ and $P^{"}(t)$ denotes the velocity and acceleration functions, respectively. Functions $P^{'}(t)$ and $P^{"}(t)$ are obtained by using the time-derivative of function $P(t)$.

The trajectory is only described by means of a set of position vectors $\{E_i : i = 1, ..., n\}$, one needs to find the velocity and the acceleration that should used during the mapping process from a segment to another. For the velocity, we can use the heuristic function which has been presented in the previous Section. In the following we present a heuristic function for finding the acceleration using information on position.

According to the conditions stated above, the boundary constraints specify the position, velocity, and acceleration at the starting time $t = 0$ and at the ending time $t = t_f$. The boundary constraints are as follows:

Initial position	: $P(t = 0) = P_0$
Initial velocity	: $P^{'}(t = 0) = V_0$
Initial acceleration	: $P^{"}(t = 0) = A_0$
Final position	: $P(t_f) = P_f$
Final velocity	: $P^{'}(t_f) = V_f$
Final Acceleration	: $P^{"}(t_f) = A_f$

The polynomial function $P(t)$ should be at least of degree five in order to satisfy the set of constraints stated above. The minimum time polynomial function $P(t)$ is then as follows:

$$P(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5$$

Its first order derivative is given by:

$$P^{'}(t) = a_1 + 2a_2 t + 3a_3 t^2 + 4a_4 t^3 + 5a_5 t^4$$

The second order derivative is:

$$P^{"}(t) = 2a_2 + 6a_3 t + 12a_4 t^2 + 20a_5 t^3$$

At time $t = 0$, the boundary conditions allow identifying the coefficients a_0, a_1 , and a_2 of polynomial $P(t)$, we have:

$$\begin{aligned} P(t=0) &= P_0 = a_0 \\ P^{'}(t=0) &= V_0 = a_1 \\ P^{"}(t=0) &= A_0 = 2a_2 \end{aligned}$$

The position, velocity, and acceleration become:

$$\begin{aligned} P(t) &= P_0 + V_0 t + \frac{A_0}{2} t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \\ P^{'}(t) &= V_0 + A_0 t + 3a_3 t^2 + 4a_4 t^3 + 5a_5 t^4 \\ P^{"}(t) &= S_0 + 6a_3 t + 12a_4 t^2 + 20a_5 t^3 \end{aligned}$$

Let us denote by $\Delta P, \Delta V$, and ΔA the variation in the position, velocity, and acceleration from the starting time until the ending time of the trajectory:

$$\begin{aligned} \Delta P &= P_f - P_0 \\ \Delta V &= V_f - V_0 \\ \Delta A &= A_f - A_0 \end{aligned}$$

Let us rewrite the Equation of the position, velocity, and acceleration with respect to the ending time $t = t_f$, we obtain the following equations:

$$\begin{aligned} P(t) &= P_f = P_0 + V_0 t_f + \frac{A_0}{2} t_f^2 + a_3 t_f^3 + a_4 t_f^4 + a_5 t_f^5 \\ P^{'}(t) &= V_f = V_0 + A_0 t_f + 3a_3 t_f^2 + 4a_4 t_f^3 + 5a_5 t_f^4 \\ P^{"}(t) &= A_f = S_0 + 6a_3 t_f + 12a_4 t_f^2 + 20a_5 t_f^3 \end{aligned}$$

By definition of $\Delta P, \Delta V$, and ΔA we can express the three Equations as follows:

$$\begin{aligned} \Delta P - V_0 t_f - \frac{A_0}{2} t_f^2 &= a_3 t_f^3 + a_4 t_f^4 + a_5 t_f^5 \\ \Delta V - A_0 t_f &= 3a_3 t_f^2 + 4a_4 t_f^3 + 5a_5 t_f^4 \\ \Delta A &= 6a_3 t_f + 12a_4 t_f^2 + 20a_5 t_f^3 \end{aligned}$$

These Equations can be reformulated with respect to the unknown coefficients a_3, a_4, and a_5:

$$\frac{1}{t_f^3}\left[\Delta P - V_0 t_f - \frac{A_0}{2} t_f^2\right] = a_3 + a_4 t_f + a_5 t_f^2 = \alpha$$

$$\frac{1}{t_f^2}\left[\Delta V - A_0 t_f\right] = 3a_3 + 4a_4 t_f + 5a_5 t_f^2 = \beta \frac{\Delta A}{t_f} = 6a_3 + 12a_4 t_f + 20a_5 t_f^2 = \gamma$$

where the terms α, β, and γ are function of the constant parameters $\Delta P, V_0 t_f, \Delta V, A_0 t_f$, and ΔA. By solving the previous system equation, we obtain solution for the parameters a_3 , a_4 , and a_5 as follows:

$$\begin{aligned} 2a_3 &= 20\alpha - 8\beta - \gamma \\ a_4 t_f &= -20\alpha + 7\beta + \gamma \\ 2a_5 t_f^2 &= 12\alpha - 6\beta - \gamma \end{aligned}$$

The polynomial function $P(t)$ allows moving from a point specified by P_0, V_0 , and A_0 to another point which is specified by P_f, V_f , and A_f within a time t_f. When a digital computer is used for path control, the generated trajectory can be represented by N spatial points that are defined by:

$$t_f = N\Delta t$$

where Δt is the sampling rate of the trajectory. At step $k, k = 0, ..., N-1$, the position will be evaluated as follows:

$$P(k) = C_0 + C_1 k + C_2 k^2 + C_3 k^3 + C_4 k^4 + C_5 k^5$$

The constants C_0, C_1, C_2, C_3, C_4, and C_5 should be computed only once before generating the trajectory. These parameters are defined as follows:

$$\begin{aligned} C_0 &= P_0 \\ C_1 &= V_0 \\ C_2 &= \frac{A_0}{2} \\ C_3 &= \frac{1}{2}\left(20\alpha - 8\beta - \gamma\right)\Delta t^3 \\ C_4 &= \frac{1}{t_f}\left(-15\alpha + \beta + \gamma\right)\Delta t^4 \\ C_5 &= \frac{1}{2t_f^2}\left(12\alpha + -6\beta - \gamma\right)\Delta t^5 \end{aligned}$$

In the following, we present another approach which is based on the use of discrete law having exact convergence.

5.2 Path control by using discrete trajectory

Recall the problem of moving from a point E_I to another point E_f . Let us design a discrete law for generating trajectories for each of the six scalar components of the effector vector $E = (x_1, ...x_6)^t$. Consider an anti-symmetric law for the acceleration and divide the allocated time T into M slice of times Δt such that $T = M\Delta t$. Three phases will be considered for the acceleration:

1. Phase 1: the system is accelerating with constant value α. In other words, the velocity is a first order polynomial and the position is a second order polynomial.
2. The system acceleration is null, the velocity is constant, and position is a first order polynomial.
3. Phase 3: the system is decelerating with constant value $-\alpha$, the velocity is a first order polynomial, and position is a second order polynomial.

To define the corresponding law, we may choose to accomplish phase 1 and phase 3 with equal number of iterations. Let k ($k \leq M/2$) denotes the number of iterations required to achieve phases 1 and 3, respectively.

The acceleration magnitude α is constant for the first phase. At step $i, i : 1, ..., K$, of phase 1 we can evaluate the acceleration A(i), velocity $V(i)$, and position $P(i)$ as functions of α and the iteration number i:

$$\begin{aligned} \text{Acceleration } A(i) &= \alpha \ \text{ for } i : 1, ..., K \\ \text{Velocity } V(i) &= \alpha.i \\ \text{Position } P(i) &= \alpha.i.(i+1)/2 \end{aligned} \tag{5.8}$$

At the end of phase 1 , we have:

$$A(k) = \alpha, V(k) = V_1 = \alpha.k, P(k) = P_1 = \alpha.k.(k+1)/2 \tag{5.9}$$

In phase 2, the acceleration is nil for N iterations. At step i ($i = 1, ..., N$) we can evaluate the acceleration $A(i)$, velocity $V(i)$, and position $P(i)$ as follows:

$$\begin{aligned} \text{Acceleration } A(i) &= 0 \ \text{ for } i : 1, ..., M \\ \text{Velocity } V(i) &= V_1 \\ \text{Position } P(i) &= P_1 + V_1.i \end{aligned} \tag{5.10}$$

At the end of phase 2, the final conditions are:

$$A(N) = 0$$

$$V(N) = V_2 = \alpha.N$$

$$P(N) = P_2 = P_1 + V_1.N$$

Finally, phase 3 is achieved by means of K iterations during which the system will be decelerating at a constant rate $-\alpha$. At iteration i ($i = 1, ..., k$) we can evaluate the acceleration $A(i)$, velocity $V(i)$, and position $P(i)$ as follows:

$$\begin{aligned} \text{Acceleration } A(i) &= -\alpha \ \text{ for } i : 1, ..., K \\ \text{Velocity } V(i) &= V_2 - \alpha.i \\ \text{Position } P(i) &= P_2 + V_2.i - \alpha.i.(i+1)/2 \end{aligned} \tag{5.11}$$

At the end of phase 3, the final conditions are:

$$A(k) = -\alpha$$

$$V(k) = V_2 - \alpha.k = V_f$$

$$P(k) = P_2 + V_2.k(k+1)/2 = P_f$$

where V_f and P_f are the final velocity and acceleration, respectively. Let us replace the terms V_1 and P_1 in the expression of $V(k)$ and $P(k)$, we obtain:

$$V_f = 0 \tag{5.12}$$

$$P_f = \alpha.K.(K+1)/2 + \alpha.K.N + \alpha.K^2 - \alpha.K.(K+1)/2 \tag{5.13}$$

Simplification of the expression of P_f and setting $P_f = 1$, allow finding the acceleration constant α:

$$\alpha = 1/K.(K + N) \tag{5.14}$$

The value of α so determined allows the system to move from position $P = 0$ to $P_f = 1$ with zero initial and final velocities.

This method can be used to generate trajectory between any pair of points E_0 and E_f in the effector space. Let $P(i)$ be the position as generated in any of the previous three phases, the effector vector $E(i)$ can be defined using the scalar $P(i)$, we have:

$$E(i) = E_0 + P(i).(E_f - E_0) \tag{5.15}$$

where $i \in [1, k]$ in phase 1 and 3, and $i \in [1, N]$ in phase 2. The scalar function $P(i)$ has been used for all the components of vector E because it allows controlling the rate of incrementation from zero up to $E_f - E_0$. The maximum velocity V_E in the E-space is then modulated by V_max of the normalized trajectory:

$$V_{max} = \alpha.K \tag{5.16}$$

and

$$V_E = \alpha.K.(E_f - E_0) \tag{5.17}$$

5.2.1 Time control of the trajectory

When the system is moved between two points the required motion time is given by

$$T = M.\Delta t = (2K + N)\Delta t \tag{5.18}$$

where Δt is the system period between the generation of two successive points $E(i)$ and $E(i + 1)$.

Since the number of iterations M is a time parameter. The value K and N can be obtained from M by any assumption such as:

$$K = \frac{N}{2} = \frac{M}{4} \tag{5.19}$$

This assumes that the system will be accelerating or decelerating during 25% of the motion time and moving at constant speed during 50 of motion time. In this case, we determine α as follows

$$\alpha = 16/3M^2 \tag{5.20}$$

The overhead of evaluating the position p(i) will be as follows:

Phase 1: for $i = 1, ..., k$

$$\begin{aligned} V &= V + A \\ P &= P + V \end{aligned}$$

Phase 2: for $i = 1, ..., N$

$$P = P + V$$

Phase 3: for $i = 1, ..., k$

$$\begin{aligned} V &= V + A \\ P &= P + V \end{aligned}$$

Therefore, the overhead in phases 1 and 3 is two additions and only one addition in phase 2. The trajectory is time controlled by using the total number of iterations M which is required to reach the target point E_f. In the following section, a method for speed-control of the trajectory will be discussed.

5.2.2 Speed control

Speed control of the trajectory consists of specifying the upper limit of the speed which corresponds to the speed of phase 2. Two equations are to be considered:

$$\alpha = \frac{1}{K(K+N)} \tag{5.21}$$

$$X_{max} = \alpha K.(X_f - X_0) \tag{5.22}$$

where X is the component of vector E on which the total increment $X_f - X_0$ is the largest among all the components of E. The control parameter is the maximum speed X_{max} . Let us suppose that M steps are required and K and N can be predetermined from M:

$$K = \frac{M}{4} \text{ and } N = \frac{M}{2} \tag{5.23}$$

The system becomes:

$$\alpha = \frac{16}{3M^2} \tag{5.24}$$

$$X_{max} = \frac{\alpha}{M}(X_F - X_I) \tag{5.25}$$

We deduce the number of steps:

$$M = E\left[\frac{4}{3}\frac{X_F - X_I}{X_{max}}\right] \tag{5.26}$$

where the function $E[.]$ defines the smallest integer that is greater than this expression. The procedure will be identical to the previous function with the exception that M and α should be computed according to Equations (8.27) and (8.29).

This method has the advantage to offer a better trajectory control and avoids possible motor saturation or under speed motion. In addition it presents a simplified form because it deals with motion velocity that is independent from motion magnitude.

5.2.3 Mapping position and velocity

Trajectory passing by arbitrary points consists of considering the continuity of speed and positions. Assume the following boundary conditions:
Starting point:

$$\begin{aligned} P(t_0) &= P_0 \\ P'(t_0) &= V_0 \end{aligned}$$

Ending point:

$$\begin{aligned} P(t_f) &= P_f \\ P^{'}(t_f) &= V_f \end{aligned}$$

The boundary constraints can be satisfied if we divide the timing of the discrete second derivative, or acceleration, into three phases. In the first phase, the acceleration will be α for k iterations. In the second phase, the acceleration will be zero for N iterations. In the third phase, the acceleration will be β for k iterations. Note that during the first and third phases the system will be accelerating ar decelerating depending on the values of α and β . The total number of iterations will then be $2K + N$. This low introduces two variables α and β that depend on the boundary conditions.

For the first phase, the acceleration, velocity, and position are defined as follows:

$$\begin{aligned} P^{"}(i) &= \alpha \\ P^{'}(i) &= V_0 + \alpha i \\ P(i) &= P_0 + V_0 i + \alpha i(i+1)/2 \end{aligned}$$

On completion of K iterations we obtain:

$$\begin{aligned} P^{"}(K) &= \alpha \\ P^{'}(K) &= V_0 + \alpha K = V_1 \\ P(K) &= P_0 + V_0 K + \alpha K(K+1)/2 = P_1 \end{aligned}$$

For the second phase, the acceleration, velocity, and position are defined as follows:

$$\begin{aligned} P^{"}(i) &= 0 \\ P^{'}(i) &= V_1 \\ P(i) &= P_1 + V_1 i \end{aligned}$$

On completion of N iterations we obtain:

$$\begin{aligned} P^{"}(K) &= 0 \\ P^{'}(K) &= V_1 = V_2 \\ P(K) &= P_1 + V_1 M = P_2 \end{aligned}$$

For the third phase, the acceleration, velocity, and position will be evaluated as follows:

$$\begin{aligned} P^{"}(i) &= \beta \\ P^{'}(i) &= V_2 + \beta . i \\ P(i) &= P_2 + V_2 i + \beta . i.(i+1)/2 \end{aligned}$$

On completion of N iterations we obtain:

$$\begin{aligned} P^{"}(K) &= \beta \\ P^{'}(K) &= V_2 + \beta K = V_f \\ P(K) &= P_2 + V_2 M = P_f \end{aligned}$$

In the following we determine the values of α and β that satisfy the boundary conditions.

The overall speed equation is given by:

$$P_f = V_0 + K(\alpha + \beta) \tag{5.27}$$

This leads to find the sum of α and β:

$$\alpha + \beta = \frac{V_f - V_0}{K} = \frac{\Delta V}{K} \tag{5.28}$$

The overall position equation is as follows: ;

$$P_f = (2K + N)V_0 + (\alpha + \beta)K(K + 1)/2 + \alpha K(k + N) + P_0$$

We assume M is the total number of iterations, we have:

$$M = 2K + N \tag{5.29}$$

Let us assume the number of iteration M is distributed as 25% for the first and third phases and 50 for the second phase:

$$K = \frac{M}{4} \; N = \frac{M}{2}$$

According to the above partition, we have:

$$\Delta V = \frac{M}{4}(\alpha + \beta) \tag{5.30}$$

$$\Delta P = V_0 M + (\alpha + \beta)\frac{M(M + 4)}{32} + \alpha\frac{3M^2}{16} \tag{5.31}$$

To find the constant α and β , we have the final velocity and position equations. The final velocity V_f can be expressed as follows:

$$V_f = V_0 + (\alpha + \beta)K$$

This leads to finding the sum α and β , we have:

$$\alpha + \beta = \frac{4\Delta V}{K} \tag{5.32}$$

On the other hand, the expression of the final position allows writing:

$$\Delta P = V_0 M + (\alpha + \beta)\frac{M(M + 4)}{32} + \alpha\frac{3M^2}{16}$$

Let us replace the term $\alpha + \beta$ by its expression in the equation of P_f , we obtain:

$$\alpha = \left[\Delta P - V_0 M - \frac{(M + 4)\Delta V}{8}\right]\frac{16}{3M^2} \tag{5.33}$$

The discrete function is completely determined by finding the acceleration values α and β . The evaluation of the position $P(i)$ and velocity $V(i)$ requires initializing the position and velocity to the values P_0 and V_0 , respectively. Depending on the current phase, the evaluation process will be as follows:

Phase 1: for $i = 1, ..., k$

$$V = V + \alpha$$
$$P = P + V$$

Phase 2: for $i = 1, ..., N$

$$P = P + V$$

Phase 3: for $i = 1, ..., k$

$$\begin{aligned} V &= V + \beta \\ P &= P + V \end{aligned}$$

Therefore, two additions are required in phases 1 and 3 and only one addition is required in phase 2. The trajectory is time controlled because of the given total total number of iterations, M. However, the designed scalar discrete function $P(i)$ cannot be used here for all the components of the effector vector E because of different initial conditions. The function should then be applied on each individual component of E which means that the overhead of using this law is 12 additions for the first or third phases and 6 additions for the second phase. The designed law allows then to move from a point specified by E_0 and $E_0^{'}$ to a the target point specified by E_f and $E_f^{'}$. As the second derivative is either constant or null, the generated trajectory is of second order.

5.2.4 Mapping position, velocity, and acceleration

In the previous section we have studied a method that ensures continuity of position and velocity. For this we have found that a simple law on the acceleration allows satisfying the boundary conditions. Smoother discrete trajectories can be found by considering constraints on the position, velocity, and acceleration. Minimum complexity can be obtained when. Assume the following boundary conditions:
Starting point:

$$\begin{aligned} P(t_0) &= P_0 \\ P^{'}(t_0) &= V_0 \\ P^{"}(t_0) &= \alpha_0 \end{aligned}$$

Ending point:

$$\begin{aligned} P(t_f) &= P_f \\ P^{'}(t_f) &= V_f \\ P^{"}(t_f) &= \alpha_f \end{aligned}$$

The boundary constraints can be satisfied if we divide the timing of the discrete second derivative, or acceleration, into three phases. In the first phase, the acceleration will be a first order polynomial starting with α_0 , which is given, and ending with α_1 . In the second phase, the acceleration will linearly changes from α_1 to α_2. In the third phase, the acceleration linearly changes from α_2 to α_f . Note that during the first, second, and third phases the system will be accelerating ar decelerating depending on the values of α_0 , α_1 , α_2 , α_f . The total number of iterations will then be M=3K, where K is the number of iteration for each of the three phases of the trajectory. This low introduces two variables α_1 and α_2 that depend on the boundary conditions.

For the first phase, the acceleration, velocity, and position are defined as follows:

$$\begin{aligned} \alpha(i) &= \frac{\alpha_1 - \alpha_0}{K} i + \alpha_0 = A_1 i + \alpha_0 \\ V(i) &= V_0 + Aaub1i(i+1)/2 + \alpha_0 i \\ P(i) &= P_0 + V_0 i + \frac{A_1}{2}\left[i(i+1)(2i+1)/6 + i(i+1)/2\right] + \alpha_0 i(i+1)/2 \end{aligned}$$

On completion of K iterations we obtain:

$$\begin{aligned}\alpha(K) &= \alpha_1 \\ V(K) &= V_1 = V_0 + A_1K(K+1)/2 + \alpha_0 K \\ P(K) &= P_1 = P_0 + V_0K + \frac{A_1}{2}\left[2K(K+1)(2K+1)/6K(K+1)/2\right] + \alpha_0 K(K+1)/2\end{aligned}$$

For the second phase, the acceleration, velocity, and position are defined as follows:

$$\begin{aligned}\alpha(i) &= \frac{\alpha_2-\alpha_1}{K}i + \alpha_1 = A_2 i + \alpha_1 \\ V(i) &= V_1 + A_2 i(i+1)/2 + \alpha_1 i \\ P(i) &= P_1 + V_1 i + \frac{A_2}{2}\left[i(i+1)(2i+1)/6 + i(i+1)/2\right] + \alpha_1 i(i+1)/2\end{aligned}$$

On completion of N iterations we obtain:

$$\begin{aligned}\alpha(K) &= \alpha_2 \\ V(K) &= V_2 = V_1 + A_2K(K+1)/2 + \alpha_1 K \\ P(K) &= P_2 = P_1 + V_1K + \frac{A_2 >}{2}\left[K(K+1)(2K+1)/6 + K(K+1)/2\right] + \alpha_1 K(K+1)/2\end{aligned}$$

For the third phase, the acceleration, velocity, and position will be evaluated as follows:
On completion of N iterations we obtain:

$$\begin{aligned}\alpha(i) &= \frac{\alpha_3-\alpha_2}{K}i + \alpha_2 = A_3 i + \alpha_2 \\ V(i) &= V_2 + A_3 i(i+1)/2 + \alpha_2 i \\ P(i) &= P_2 + V_2 i + \frac{A_3}{2}\left[i(i+1)(2i+1)/6 + i(i+1)/2\right] + \alpha_2 i(i+1)/2\end{aligned}$$

On completion of K iterations we obtain:

$$\begin{aligned}\alpha(K) &= \alpha_3 \\ V(K) &= V_f = V_2 + A_3K(K+1)/2 + \alpha_2 K \\ P(K) &= P_f + P_2 + V_2K\frac{A_3}{2}\left[K(K+1)(2K+1)/6 + K(K+1)/2\right] + \alpha_2 K(K+1)/2\end{aligned}$$

In the following we determine the values of V_1, V_2 , and V_3 as function of α_0 , α_1 , α_2 , α_3 and V_0 , we obtain the following expression for the velocities:

$$\begin{aligned}V_1 &= V_0 + (\alpha_1 - \alpha_0)(K+1)/2\alpha_0 K \\ V_2 &= V_0 + (\alpha_2 - \alpha_0)(K+1)/2 + (\alpha_0 + \alpha_1)K \\ V_f &= (\alpha_f - \alpha_2)(K+1)/2 + (\alpha_0 + \alpha_1 + \alpha_2)K + V_0\end{aligned}$$

Denote by $\Delta V = V_f - V_0$ the overall variation in the velocity. The unknown parameters are $alpha_1$ and α_2 , using the expression of the final velocity V_f we obtain:

$$\alpha_1 + \alpha_2 = \frac{1}{K}\left[\Delta V - \alpha_3\frac{K+1}{2} - \alpha_0\frac{K-1}{2}\right]$$

Similarly, we evaluate the expression of the positions P_1 , P_2 , and P_f as follows:

$$\begin{aligned} P_1 &= V_0K + \frac{\alpha_1-\alpha_0}{2}\left[K(K+1)(2K+1)/6 + K(K+1)/2\right] + \alpha_0 K(K+1)/2 \\ P_2 &= 2V_0K + P_0 + \frac{1}{2}(\alpha_2-\alpha_0)\left[K(K+1)(2K+1)/6 + K(K+1)/2\right] \\ &\quad +(\alpha_0+\alpha_1)K(K+1)/2 + (\alpha_1-\alpha_0)K(K+1)/2 + \alpha_0K^2 \\ P_f &= P_0 + 3V_0K + \frac{1}{2}(\alpha_3-\alpha_0)\left[(K+1)(2K+1)/6 + (K+1)/2\right] \\ &\quad +(\alpha_0+\alpha_1+\alpha_2)K(K+1)/2 + (\alpha_2+\alpha_1-2\alpha_0)K(K+1)/2 + (2\alpha_0+\alpha_1)K^2 \end{aligned}$$

Let us replace the term $\alpha_1 + \alpha_2$ by its expression in the equation of P_f , after rearranging the terms we obtain:

$$P_f = P_0 + (2K-1)V_0 + (K+1)V_3\frac{(5K-1)(K-1)}{6}\alpha_0 - \frac{(K+1)(K+2)}{12}\alpha_3 + K^2\alpha_1$$

As the parameters $P_0, V_0, \alpha_0 P_3, V_f, \alpha_f$ are known, the term α_1 can then be determined by using this equation:

$$\alpha_1 = \frac{1}{K_2}\left[\Delta P - (2K-1)V_0 - (K+1)V_3 - \frac{(5K-1)(K-1)}{6}\alpha_0 + \frac{(K+1)(K+2)}{12}\alpha_3\right]$$

The discrete function is completely determined by finding the acceleration values α_1 and α_2 . The evaluation of the position $P(i)$, velocity $V(i)$, and acceleration $\alpha(i)$ requires initializing the position P, velocity V, and acceleration α to the values P_0, V_0 , and α_0 respectively:

$$P = P_0 V = V_0 \alpha = \alpha_0$$

Three constant should also be evaluated before hand:

$$\begin{aligned} A_1 &= \frac{\alpha_1-\alpha_0}{K} \\ A_2 &= \frac{\alpha_2-\alpha_1}{K} \\ A_3 &= \frac{\alpha_f-\alpha_2}{K} \end{aligned}$$

Depending on the current phase, the evaluation process will be as follows:

Phase 1: for $i = 1, ..., K$

$$\begin{aligned} \alpha &= \alpha + A_1 \\ V &= V + \alpha \\ P &= P + V \end{aligned}$$

Phase 2: for $i = 1, ..., K$

$$\begin{aligned} \alpha &= \alpha + A_2 \\ V &= V + \alpha \\ P &= P + V \end{aligned}$$

Phase 3: for $i = 1, ..., k$

$$\begin{aligned} \alpha &= \alpha + A_3 \\ V &= V + \alpha \\ P &= P + V \end{aligned}$$

Therefore, three additions are required in either phases 1, 2, or 3. This shows that the discrete law that maps the position, velocity, and acceleration has low on-line overhead compared with a polynomial approach for the same problem. The trajectory is time controlled because of the given total total number of iterations M and the number of iterations K for each phase:

$$K = \frac{M}{3}$$

Unfortunately, the designed scalar discrete function $P(i)$ cannot be used here for all the components of the effector vector E because of different initial and final conditions. The function should then be applied on each individual component of E which means that the overhead of using this law is 18 additions for any phase. The designed law allows then to move from a point specified by E_0 , $E_0^{'}$, and $E_0^{"}$ to a the target point specified by E_f , $E_f^{'}$, and $E_0^{"}$. As the second derivative is linear function of the time, the generated trajectory is of third order, i.e. function of K^3 .

5.2.5 Example of programming and optimization

1. Consider the time evolution of each component of E separately.

2. Evaluate the speed as follows:

$$V1 = \frac{X_2 - X_0}{t_2 - t_o} \; ; \; V2 = \frac{X_3 - X_1}{t_3 - t_1} \tag{5.34}$$

3. Use the filter function $p(i)$ as follows:
   $X(i) \leftarrow XI + P(i).(X_F - X_I)$
   Then

$$\begin{aligned} X_M &= X_F \\ V_M &= V_F \end{aligned} \tag{5.35}$$

   The total execution time:

$$T = t_{i+2} - t_{i-1} = M.\Delta T \tag{5.36}$$

   Since the number of iterations is:

$$M = T/\Delta T \tag{5.37}$$

   Let us determine the number of iterations for each part. First the total position increments:

$$\Delta = \sum_{i=1}^{n} |x_i - x_{i-1}| = \sum \Delta_i \tag{5.38}$$

   For each part of the trajectory the number of iterations M_i can be proportional to its increment Δ_i/Δ:

$$\frac{M_i}{M} = \frac{\Delta_i}{\Delta} \tag{5.39}$$

   In addition every Δ_i is divided into four segments (Acc., constant speed, deceleration), since M_i is to be a multiple of 4.

Exercises

1. It is desired to move from position vector $E_I = (200, 300, 400)^t$ to position vector $E_F = (600, 500, -100)^t$ in a time of 1 second.
   (a) Find the expression of the required three polynomials.
   (b) Find a discrete scalar law from $x_I = 0$ to $x_f = 1$ with M iterations.
   (c) Show how to use the discrete scalar law for moving from E_I to E_f within 1 second.

2. Compare the complexity of the polynomial approach with that of the discrete scalar law in the problem of moving between two vector points.

3. Using your result of problem 8.1, write a program to generate the trajectory between points E_I and E_f . Show the plot of the resulting trajectory together with the speed and acceleration.

4. Consider a set of vector points $E = \{E_o, E_1, ..., E_n\}$ each is that of dimensional vector-points and divide this set into n segment s_i:

$$s_i = \{E_< i - 1 >, E_i\}$$

   Design a polynomial function which can be used for all the scalar component of $E_i = X_1, ..., X_6{}^t$. The function should allow moving from E_{i-1} to E_i and ensure position, velocity, and acceleration continuity between consecutive segments. Use the set of constraints in Section 8.2. Velocity can be evaluated in a heuristic way.

5. Recall the problem 4, Design a discrete scalar law for motion generation between any pair of points in a defined segment s_i . To move between two points the acceleration parameter α and β should be evaluated for each scalar component of the vector E. The discrete law is to ensure continuity of position and velocity.

6. Write a program to execute the instruction:

$$\text{MOVE TO } E_f \; VIA \;\; E_1, E_2, ..., E_{n-1}.$$

   Choose either the polynomial or the discrete methods for path control. The program should partition the trajectory into segments and call a routine to process the motion within a segment. Output of the program should show the plot of the position, velocity, and acceleration versus time.