Zero states polynomial-like trajectory (ZSPOT) generation

Ki Tak Ahn, Wan Kyun Chung and Youngil Youm

Robotics & Bio-mechatronics Lab., Department of Mechanical Engineering
Pohang University of Science and Technology(POSTECH), Pohang 790-784, Korea
(Tel: +82-54-279-5946; Fax: +82-54-279-5899; Email:{termi, wkchung, youm}@postech.ac.kr)

Abstract: In the area of tracking control, it is important to design not only the controllers but also the trajectories to which a system has to follow. Position in the form of the 5^{th} order polynomial is often used with constraints of initial and final states. Smooth ending with possible minimum time is important for many systems to be away from vibrations or jerky motions. A simple polynomial-like trajectory generation method based on zero final state constraints is suggested and named ZSPOT. The effects of suggested method are shown through experiments in which a system follows an easy and computationally light reference trajectory.

Keywords: trajectory generation, polynomial

1. Introduction

In practice of tracking control, it is important to design not only the controllers but also the trajectories to which a system has to follow. There are some methods for trajectory generation; polynomial, spline, sinusoidal and trapezoidal shape, etc. The 5^{th} order polynomial is often used with constraints of initial and final states. The order 5 comes from the optimal problem which minimizes the integral of jerk square during run time in bio engineering [1], especially for the motion of the human hand and the guide of rehabilitation. But as for the non-bio systems, minimum jerk during full running time is not necessary. Smooth ending with possible minimum time is more important for many systems because vibration or jerky motions may cause some wears for the system. The suggested trajectories are of polynomial nature which is continuous and smooth and require inputs to the actuators which has the same nature. When a system moves something which needs to be carried carefully, this nature is essential. Examples are increased with development of technology in smaller, more accurate systems. The typical polynomial is defined over the real domain. But suggested trajectory is defined only at positive integer value, "polynomial-like" is used instead of "polynomial". The difference between them is shown in section V.

On the base of a polynomial-like trajectory generation method for Hard Disk Drive (HDD) system suggested by H.Uchida and T.Semba [2], the generalized and expanded order of polynomial-like trajectory generation method is shown. The generated trajectory is used as reference set points to be tracked by the controller. The higher the order of the reference polynomial, the faster the system arrived to the final steady state. The constraints for the reference can be expanded to the state of the n^{th} derivative of position. Therefore, not only acceleration(2^{nd} derivative) but also jerk(3^{rd} derivative) and even more minute concepts were considered [3], [4].

Because the suggested method is based on the constraints that all final states are zeros, it is called ZSPOT (Zero States POlynomial-like Trajectory). Without watching control issues behind this method, here shows it only as a new trajectory generation method which is easy to program and can be applied to most systems we handle.

Because the suggested method needs much less computational load than other trajectory generation methods, it is useful to the independent systems which has its own CPU. ZSPOT is applied to the linear motor system, an example of a 2^{nd} order system platform, and shows tracking performances in many cases.

The rest of this paper is arranged as follows. Section II summarizes a polynomial-like trajectory generation method called "reference model generation" from the title of reference paper [2]. In section III it is generalized and expanded to the more minute concept and then named by ZSPOT. Section IV shows the effects of ZSPOT briefly. An experiment on the platform of PID controlled linear motor system for a chip mounting device is performed in section V. The conclusion and future work are discussed in section VI.

2. Reference Model Generation

When an acceleration has a type of a polynomial function of time and in the case of 3^{rd} order polynomial, it can be written as

$$a[n] = c_1 \cdot n + c_2 \cdot n^2 + c_3 \cdot n^3 + a[0] \tag{1}$$

where n is the discrete time.

Then the velocity and position are described as the following simple equations [5], [6].

$$\begin{aligned} v[n] &= v[n-1] + a[n-1] \\ x[n] &= x[n-1] + v[n-1] \end{aligned} \tag{2}$$

These are the form of a double integrator and we define each terms in (2) as follows.

$$\begin{array}{ll} a[n] & acceleration \\ v[n] & velocity \\ x[n] & position \end{array}$$

This research was supported by the National Research Laboratory (NRL) Program(M1-0302-00-0040-03-J00-00-024-00) of the Ministry of Science & Technology, Republic of Korea.

at discrete time n. These can be rewritten after recursive expansion and arrangement with coefficients.

$$\begin{aligned} v[n] &= v[0] + \sum_{i=0}^{n-1} a[i] \\ x[n] &= x[0] + \sum_{i=0}^{n-1} v[i] \end{aligned} \tag{3}$$

The desired trajectory starts to move at time zero with $a[0], v[0], x[0]$ and ends at time $n = N$. For the steady motion at final time, that is to say, for the smooth settling, the states at final time N are all 0.

$$a[N] = v[N] = x[N] = 0$$

This is the reason why the name "zero states" is included in next section. Using these conditions we can get the states at time n with some tedious development procedures. Any symbolic math program might be very helpful. By substituting equation (1) to (3) and with all zero final state conditions above, the coefficient c_1, c_2, c_3 can be solved [2]. Then, the first input term $a[1]$ is

$$\begin{aligned} a[1] &= c_1 + c_2 + c_3 + a[0] \\ &= \tfrac{-9+N}{N} a[0] - \tfrac{36}{N(N+1)} v[0] - \tfrac{60}{N(N+1)(N+2)} x[0] \end{aligned}$$

where N is the final discrete time which equals to the total sampling number and therefore depends on the sampling time. For a system sampled at 100 Hz and run in 2 seconds, N would be 200. This equation can be converted into a general form as follows.

$$\begin{aligned} a[2] = & \tfrac{-9+N-1}{N-1} a[1] - \tfrac{36}{(N-1)(N-1+1)} v[1] \\ & - \tfrac{60}{(N-1)(N-1+1)(N-1+2)} x[1] \end{aligned}$$

$$\vdots \quad \begin{aligned} a[1] &\rightarrow a[n] \\ a[0] &\rightarrow a[n-1] \\ v[0] &\rightarrow v[n-1] \\ x[0] &\rightarrow x[n-1] \\ N &\rightarrow N-(n-1) \end{aligned}$$

$$\begin{aligned} a[n] = & \tfrac{-9+N-(n-1)}{N-(n-1)} a[n-1] \\ & - \tfrac{36}{(N-(n-1))(N-(n-1)+1)} v[n-1] \\ & - \tfrac{60}{(N-(n-1))(N-(n-1)+1)(N-(n-1)+2)} x[n-1] \end{aligned}$$

Dividing this with constant parts and variable parts, we can rewrite

$$\begin{aligned} a[n] = & (1 + \alpha \cdot K[n-1]) \cdot a[n-1] \\ & + \beta \cdot K[n-1] \cdot K[n-2] \cdot v[n-1] \\ & + \gamma \cdot K[n-1] \cdot K[n-2] \cdot K[n-3] \cdot x[n-1] \end{aligned} \tag{4}$$

where the varying parts

$$K[n-1] = \frac{1}{N-(n-1)}$$

and the constants

$$\begin{bmatrix} \alpha & \beta & \gamma \end{bmatrix} = \begin{bmatrix} -9 & -36 & -60 \end{bmatrix}$$

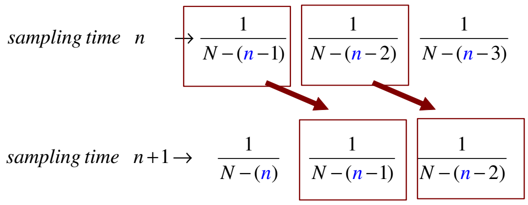


Fig. 1. Updated terms at sampling time:Time Varying part

In the reference [2], constants of several order input cases were shown up to the 7^{th} orders.

$$\begin{aligned} [\alpha\ \beta\ \gamma] &= [\ -9 \quad -36 \quad -60\ ] && 3^{rd}\ order \\ &= [\ -12 \quad -60 \quad -120\ ] && 4^{th}\ order \\ &= [\ -15 \quad -90 \quad -210\ ] && 5^{th}\ order \\ &= [\ -18 \quad -126 \quad -336\ ] && 6^{th}\ order \\ &= [\ -21 \quad -168 \quad -504\ ] && 7^{th}\ order \end{aligned}$$

This reference model generation structure includes states and time-varying terms ahead of just one step. But because two of the three time-varying terms are obtained already at previous step as shown in Fig.1, the new terms to be calculated is just one. All others can be taken from the memory of one step before.

Therefore, for the trajectory generation on the system with its own CPU or real time OS such as hard disk drive, CNC manufacturing machine and mobile robots, this can significantly reduce the computational loads. In addition, the same computational load is required regardless of the order of base polynomials.

3. ZSPOT (Zero States POlynomial-like Trajectory)

Although choosing a formula on the base of a few observations or somewhat tedious calculations does not guarantee the validity of the formula, pattern recognition of a sequence is useful to find some rules. Finite differences can help find the pattern for generalizing a formula in section II.

With the concept of finite differences, the reference generation in section II was generalized and expanded up to the any high order as well as more minute physical concepts were shown [3]. All constants in equation (4) were given in the general form. To avoid the confusion due to the character n, refer m to the general discrete sampling time and n to the order of the acceleration polynomial $a(m)$. So $a(m)$ has the highest order term m^n.

With total sampling number N,

$$\begin{aligned} a[m] = & (1 + \alpha \cdot K[m-1]) \cdot a[m-1] \\ & + \beta \cdot K[m-1] \cdot K[m-2] \cdot v[m-1] \\ & + \gamma \cdot K[m-1] \cdot K[m-2] \cdot K[m-3] \cdot x[m-1] \end{aligned} \tag{5}$$

where,

$$K[m-1] = \frac{1}{N-(m-1)}$$

has the constants in the general form of

$$\begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = \begin{bmatrix} -3n \\ -3n(n+1) \\ -n(n+1)(n+2) \end{bmatrix} \tag{6}$$

And the integration terms are

$$\begin{aligned} v[m] &= v[m-1] + a[m-1] \\ x[m] &= x[m-1] + v[m-1] \end{aligned} \tag{7}$$

Through equations (5)∼(7), all smooth and sampled states (acceleration, velocity and position) for general order n are extracted. Because of the all zero final state conditions and the close similarity to the usual polynomial, the trajectory made from this generation method is named as zero states polynomial-like trajectory, ZSPOT. Each profile is used to a reference upon the request of target systems. If a system behaves just like an double integrator owing to one of the model based controllers, all states of ZSPOT are exactly matched to the states of that system [4].
The conditions for this ZSPOT are as follows.
1. Three states(acceleration, velocity and position) are all zeros at final time. When continuous connection between position trajectories is needed, offset and change of the sign can be applied.
2. Minimum order of the acceleration polynomial is three. In double integrator system, control input is proportional to the acceleration and ZSPOT starts with the 3^{rd} order. So position has more than 5^{th} order polynomial-like equation.
3. The higher order ZSPOT has, the more constraints are needed. Let the accelerations be zeros at more steps before final time. $a[N-1] = 0$, $a[N-2] = 0$, etc. This means the higher the order of ZSPOT has, the sooner the system becomes steady.
As mentioned in section I, most of the high technologies today need smooth motion not in full time but just at final stage. Fast moving but steady final ending makes the system more stable and guarantees the accurate motion. Therefore, more minute concept than acceleration at ending can be considered.
Similar to but expanded with acceleration based procedure, we begin with the jerk equation as follows;

$$j[m] = c_1 \cdot m + c_2 \cdot m^2 + c_3 \cdot m^3 + c_4 \cdot m^4 + j[0]$$

where m is the discrete sampling time. The results are

$$\begin{aligned} j[m] &= (1 + \alpha \cdot K[m-1]) \cdot j[m-1] \\ &+ \beta \cdot K[m-1] \cdot K[m-2] \cdot a[m-1] \\ &+ \gamma \cdot K[m-1] \cdot K[m-2] \cdot K[m-3] \cdot v[m-1] \\ &+ \delta \cdot K[m-1] \cdot K[m-2] \cdot K[m-3] \\ &\cdot K[m-4] \cdot x[m-1] \end{aligned} \tag{8}$$

where,

$$K[m-1] = \frac{1}{N-(m-1)}$$

Table 1. Constants upon the order of jerk equation

	Coef. \ Order	4	5	6	7
	Alpha	-16	-20	-24	-28
	beta	-120	-180	-252	-336
	gamma	-480	-840	-1344	-2016
New term →	delta	-840	-1680	-3024	-5040

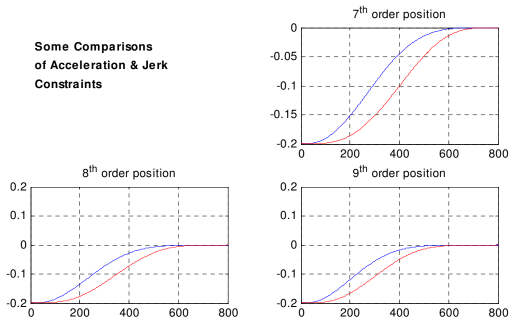


Fig. 2. Comparison of ZSPOT$_{acc}$ and ZSPOT$_{jerk}$

Now we have one more constant than those of ZSPOT based on the acceleration (Table 1). These 4 constants are in the general form as follows using pattern recognition.

$$\begin{bmatrix} \alpha \\ \beta \\ \gamma \\ \delta \end{bmatrix} = \begin{bmatrix} -4n \\ -6n(n+1) \\ -4n(n+1)(n+2) \\ -n(n+1)(n+2)(n+3) \end{bmatrix} \tag{9}$$

where n is the order of jerk polynomial.
The conditions for this jerk based ZSPOT are as follows.
1. Four states(jerk, acceleration, velocity and position) are all zeros at final time. When continuous connection of trajectories is needed, offset and change of the sign will be applied.
2. Minimum order of the base polynomial, jerk in this case, is four. So position has more than 6^{th} order polynomial-like equation.
3. The higher order ZSPOT has, the more constraints are needed. Let the jerks equal 0 at more steps before final time. $j[N-1] = 0$, $j[N-2] = 0$, etc.
For distinction, let the former be ZSPOT$_{acc}$ and the latter be ZSPOT$_{jerk}$.
The comparison of ZSPOT$_{acc}$ and ZSPOT$_{jerk}$ under the condition with the same order position polynomial is shown in Fig.2. The x-axis is the sampling number and the y-axis is the position in meter. The lower one is the position trajectory generated by ZSPOT$_{jerk}$. Both of them satisfy the condition of steady settling at final time. But in the case of ZSPOT$_{acc}$, it seems to require an over action which needs more input to the actuators of the system.
Nevil Hogan proposed that reaching movements of human arms are planned based on a maximum smoothness criterion that is equivalent to minimizing “jerk”, which in turn is the third time derivative of position (or the first time derivative of acceleration) [1]. There are more minute concept. An official term for the first time derivative of jerk is “snap”. The second and third time derivatives of jerk (i.e. the 5^{th} and 6^{th} derivatives of position) would naturally be referred to as “crackle” and “pop”. High jerk (high changes in acceleration, and therefore high changes in force), can cause substantial damage to the dynamic systems, and induce unwanted vibrations. It is also really hard to maneuver on a bus which operates with high jerk. Snap, crackle and pop are less easy concept to have a physical feelings. The ZSPOT method can be expanded to the concepts like snap, crackle,

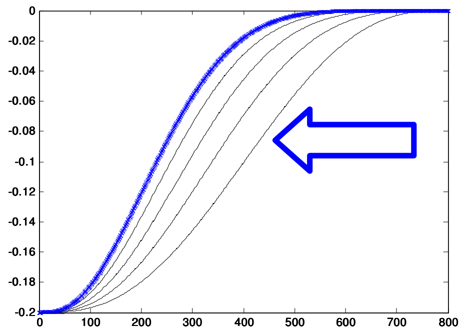


Fig. 3. Trajectory with varying order of ZSPOT$_{acc}$ polynomial

pop and more minute physical concepts in the same way shown above. A expansion to the “snap” was shown in [3].

4. Effects

The trajectories from ZSPOT method show more rapid steady settlings with higher order of reference polynomials (Fig.3). X-axis is of sampling number and y-axis is of position in meter.

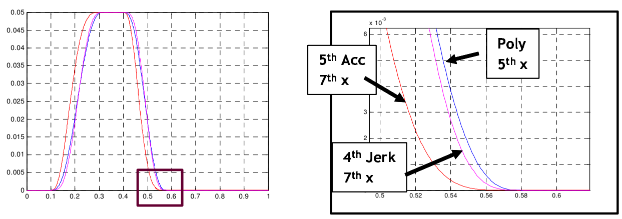
Fig.4 illustrates a comparison between ZSPOTs when two trajectories are joined for an alternating motion. In comparison, the order of position polynomial are set to be equal. Because the minimum order of ZSPOT$_{jerk}$ is 4, 5^{th} ZSPOT$_{acc}$ is used to match with the 4^{th} ZSPOT$_{jerk}$ and the order of positions is 7 in both cases. Fig.4 shows the references including a typical polynomial of the 5^{th} order position for comparison.

When determining the order or the shape of the reference trajectory, considering the required degree of system's steady state is important to reduce the tracking error.

This is easily done by changing only the constants in the equation (6) for ZSPOT$_{acc}$ or in the equation (9) for ZSPOT$_{jerk}$.

First of all, ZSPOT is a trajectory generation system itself. We can take any state that we need. For example, a position state is taken in the experiment of the next section.

Because only constants are changed with desired order, it significantly reduces the computational load. This is a major advantage when ZSPOT is used to a system of its own CPU or of rapid motion. The sample codes in the appendix shows apparent results that it is possible to change the order of trajectory easily without additional load for computing.



(a)Polynomial, ZSPOT$_{acc}$ and ZSPOT$_{jerk}$ (b)Enlarged view

Fig. 4. Comparison plot between three references

Fig. 5. Linear motor system

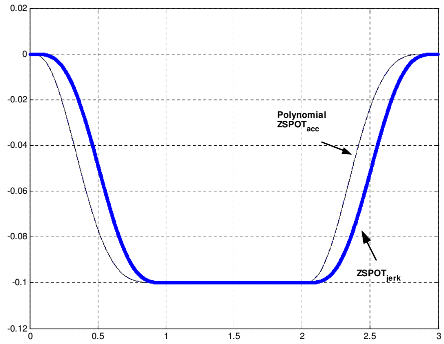


Fig. 6. Reference trajectories

5. Experiments

Some experiments were performed that a system followed given reference trajectories generated by ZSPOT. The system we are dealing with is the one axis precision linear motor system used in the semiconductor chip mounting devices, which is shown in Fig. 5. The linear motor(ANORAD Corp., LEB–S–2–S–NC) is a direct drive motor with no backlash. The control frequency is set to 1000 Hz and the position is measured by a linear encoder whose resolution is 5 μm.

The mathematical model has the 2^{nd} order one and PID control was applied to this system.

The polynomial orders used for the desired trajectory are given as 7^{th} order polynomial function of position. In the experiments, the running time is three seconds to move front and back 10 cm each. It moved 10 cm in one second, stayed in next one second then moved back to the origin. To estimate the effect of different ZSPOTs which have the same order of position polynomial, 5^{th} order ZSPOT$_{acc}$ and 4^{th} order ZSPOT$_{jerk}$ were given. As shown in Fig.6, polynomial and ZSPOT$_{acc}$ (thin lines - they seem almost the same but different when x-scale is magnified.) show faster arrival than ZSPOT$_{jerk}$ case(thick line). It means that although both of them satisfy the constraints of zero final conditions, the former needed to move faster but required more power. During all running times, the motions are continuous and not jerky.

Fig.7 shows the reduction of position tracking error when ZSPOT$_{jerk}$ is applied. This shows tracking performance to

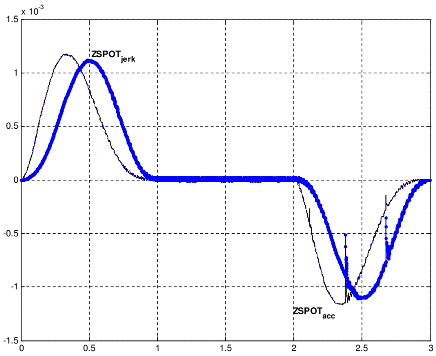
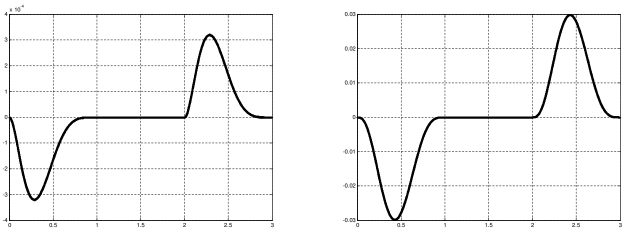


Fig. 7. Errors in position tracking: ZSPOT



(a)polynomial vs ZSPOT$_{acc}$ (b)polynomial vs ZSPOT$_{jerk}$

Fig. 8. Comparison of differences between polynomial and ZSPOTs

the given reference on a system. Although the initial and final condition are satisfied, more errors during the moving sequences made the controller work harder. It is clear that the ideal tracking needs no control. If we can give a system more followable references, cheaper and simpler controllers can survive. This experiment shows the importance of designing not only the controllers but also the trajectories to which a system has to follow.

Each reference shown above is different even between polynomial and ZSPOT$_{acc}$, although they seem to be similar in Fig. 6. The differences can be shown by comparing the tracking outputs of each case (Fig.8). There are differences among three references of the same position order. ZSPOT guides a system in a different way from typical polynomial trajectory does due to its discrete character. The smaller the sampling time at which we give the reference to the system, the more similar ZSPOT$_{acc}$ is to the typical continuous polynomial.

Fig.9 shows one more result that ZSPOT show more rapid steady settle with higher order of reference but requires more energy input to the system actuator.

6. Conclusion

To design an easy and computationally light reference trajectory is as important as to design a good controller for tracking problems. As one of the trajectory generation methods, ZSPOT (zero states polynomial-like trajectory) generation method is suggested. ZSPOT can be applied to any cases in which polynomial natured equations are used to generate a motion profile.

Based on the time-varying state feedback structure, ZSPOT

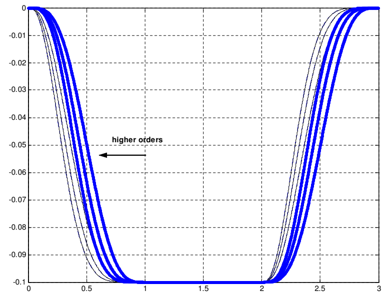


Fig. 9. Tracking with different ZSPOT orders (position order 7∼9)

is generalized and expanded from reference model generation. The simulation and experiment apparently show that it is possible to change the order of trajectory easily without additional calculation load. Just changing to the higher constant related to the order of polynomial makes faster trajectory with keeping a smooth and steady final ending.

Under any steady ending condition in any physical concept, ZSPOT can make a reference state trajectory. It is interesting that the constants in equations (6) and (9) follows the Pascal's Triangle. Therefore, the equation for expanding more minute physical concept like snap can be taken without calculator.

Future works are as follow. Because any initial condition can be applied and ZSPOT methods runs in realtime, it maybe possible to change the trajectory for a new target by putting all present states into a new initial state and reconstructing a trajectory at any time of motion. If arbitrary non-zero final conditions are involved, it maybe called an ASPOT (Arbitrary States polynomial-like Trajectory) method.

References

[1] T.Flash and N.Hogan, "The coordination of arm movements: An experimentally confirmed mathematical model," *JNeuro*, vol. 5, pp. 1688–1703, 1985.

[2] H. Uchida and T. Semba, "Reference model generation using time-varying feedback for model-following control and application to a hard disk drive," in *Proc. 2002 American Control Conference*, pp. 1389–1394, 2002.

[3] K. T. Ahn, W. K. Chung, and Y. Youm, "Reference model generation for tracking and ending in steady final state," in *International conference on control, automation and systems*, pp. 101–106, 2003.

[4] K. T. Ahn, "A trajectory generation for model-following control based on smooth, rapid settling constraints," Master's thesis, (in Korean) POSTECH, 2003.

[5] K. J.Astrom and B. Wittenmark, *Computer-Controlled Systems*. Prentice-Hall,Inc., 3rd ed., 1997.

[6] G. F. Franklin, J. D. Powell, and M. L. Workman, *Digital Control of Dynamic Systems*. Addison Wesley, 1990.

Appendix

For comparison, a polynomial trajectory whose position order is 7 is shown for example.

$$\begin{aligned} a(n) &= c_1 n + c_2 n^2 + c_3 n^3 + c_4 n^4 + c_5 n^5 + a_0 \\ v(n) &= \tfrac{1}{2} c_1 n^2 + \tfrac{1}{3} c_2 n^3 \\ &\quad + \tfrac{1}{4} c_3 n^4 + \tfrac{1}{5} c_4 n^5 + \tfrac{1}{6} c_5 n^6 + a_0 n + v_0 \end{aligned}$$

The position trajectory is

$$\begin{aligned} x(n) = &\tfrac{1}{6} c_1 n^3 + \tfrac{1}{12} c_2 n^4 + \tfrac{1}{20} c_3 n^5 + \tfrac{1}{30} c_4 n^6 \\ &+ \tfrac{1}{42} c_5 n^7 + \tfrac{1}{2} a_0 n^2 + v_0 n + x_0 \end{aligned} \tag{10}$$

at n^{th} discrete sampling.

Equation (10) is put in the trajectory part of the program code. All coefficients c_1, c_2, c_3, c_4 and c_5 are calculated before action of the system. But this computation is complicated and required some computational burdens as shown below.

$$c_1 = -\frac{\left(\begin{array}{l} 30u_0N^8 - 420u_0N^7 + 180v_0N^7 + 2405u_0N^6 \\ -2340v_0N^6 + 420x_0N^6 - 7020u_0N^5 \\ +12420v_0N^5 - 5040x_0N^5 + 10868u_0N^4 \\ -33660v_0N^4 + 24780x_0N^4 - 8400u_0N^3 \\ +48600v_0N^3 - 63000x_0N^3 + 2520u_0N^2 \\ -35280v_0N^2 + 86520x_0N^2 + 10080v_0N \\ -60480x_0N + 16800x_0 \end{array}\right)}{\left(\begin{array}{l} (N-1)(N-2) \\ \left(2N^4 - 24N^3 + 109N^2 - 210N + 140\right) N^3 \end{array}\right)}$$

$$c_2 = 20\frac{\left(\begin{array}{l} 6u_0N^8 - 78u_0N^7 + 48v_0N^7 + 411u_0N^6 \\ -576v_0N^6 + 126x_0N^6 - 1089u_0N^5 \\ +2796v_0N^5 - 1386x_0N^5 + 1502u_0N^4 \\ -6840v_0N^4 + 6195x_0N^4 - 1008u_0N^3 \\ +8770v_0N^3 - 14175x_0N^3 + 252u_0N^2 \\ -5544v_0N^2 + 17304x_0N^2 + 1344v_0N \\ -10584x_0N + 2520x_0 \end{array}\right)}{\left(\begin{array}{l} (N-1)(N-2) \\ \left(2N^4 - 24N^3 + 109N^2 - 210N + 140\right) N^4 \end{array}\right)}$$

$$c_3 = -20\frac{\left(\begin{array}{l} 10u_0N^8 - 120u_0N^7 + 90v_0N^7 + 575u_0N^6 \\ -990v_0N^6 + 252x_0N^6 - 1350u_0N^5 \\ +4335v_0N^5 - 2520x_0N^5 + 1582u_0N^4 \\ -9315v_0N^4 + 10080x_0N^4 - 840u_0N^3 \\ +10080v_0N^3 - 20160x_0N^3 + 140u_0N^2 \\ -5040v_0N^2 + 20748x_0N^2 + 840v_0N \\ -10080x_0N + 1680x_0 \end{array}\right)}{\left(\begin{array}{l} (N-1)(N-2) \\ \left(2N^4 - 24N^3 + 109N^2 - 210N + 140\right) N^5 \end{array}\right)}$$

$$c_4 = 5\frac{\left(\begin{array}{l} 30u_0N^7 - 330u_0N^6 + 288v_0N^6 + 1415u_0N^5 \\ -2880v_0N^5 + 840x_0N^5 - 2835u_0N^4 \\ +11160v_0N^4 - 7560x_0N^4 + 2576u_0N^3 \\ +20160v_0N^3 + 26460x_0N^3 - 840u_0N^2 \\ +16632v_0N^2 - 44100x_0N^2 - 5040v_0N \\ +34440x_0N - 10080x_0 \end{array}\right)}{\left(\begin{array}{l} (N-1)(N-2) \\ \left(2N^4 - 24N^3 + 109N^2 - 210N + 140\right) N^5 \end{array}\right)}$$

$$c_5 = -7\frac{\left(\begin{array}{l} +6u_0N^6 - 60u_0N^5 + 60v_0N^5 + 225u_0N^4 \\ -540v_0N^4 + 180x_0N^4 - 360u_0N^3 \\ +1800v_0N^3 - 1440x_0N^3 + 200u_0N^2 \\ -2520v_0N^2 + 4260x_0N^2 + 1200v_0N \\ -5400x_0N + 2400x_0 \end{array}\right)}{\left(\begin{array}{l} (N-1)(N-2) \\ \left(2N^4 - 24N^3 + 109N^2 - 210N + 140\right) N^5 \end{array}\right)}$$

where N is the total sampling number. The system was sampled at 1000Hz and ran in one second per one movement in the experiment, so N was 1000.

When one wants to try different order polynomial trajectory, tedious pre-computation for getting new coefficients and long code which has to be typed again are needed.

In the case of ZSPOT, only one change of order n in equation (6) or (9) was needed. Once ZSPOT code is made, testing a lot of trajectories is easy.

Of course, all calculations of coefficients above can be reduced if every movement starts in zero acceleration and zero velocity.