ARBITRARY STATES POLYNOMIAL-LIKE TRAJECTORY (ASPOT) GENERATION AND ITS APPLICATIONS

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
Telephone: (+82-54) 279-5946, Fax: (+82-54) 279-5899
e-mail: {termi, wkchung, youm}@postech.ac.kr

Abstract: A new method of smooth trajectory generation which can replace typical polynomial type trajectory is suggested. A simple, practical, discrete polynomial-like trajectory generation method which is based on zero final state constraints was suggested and named ZSPOT. There is a progress about it by allowing non-zero final constraints and named ASPOT. Without watching control issues behind these methods, they are shown only as a new trajectory generation method which is of discrete form so that easily applicable to most practical systems. ZSPOT is applied to a simulation of mobile-manipulator system and the effect of ASPOT is shown through experiment of a linear motor system.

Keywords: Trajectories, Discrete systems, Mobile, Manipulator, Linear motor

1. INTRODUCTION

Smooth ending with possible minimum time is more important for many systems because vibration or jerky motions may cause possible wears for the system. High jerk (high changes in acceleration, and therefore high changes in force), could cause substantial damage to the dynamic systems and induce unwanted vibrations. When a system moves something which needs careful carrying, this nature is essential. Examples are increased with development of technology in the smaller, more accurate systems. Various methods have been proposed to generates trajectories minimizing the residual vibration problems. SMART(Structural vibration Minimized AcceleRation Trajectory) method which generates smooth control wave forms is one of those methods (Hara *et al.*, 1999; Mizoshita *et al.*, 1996). Most methods are using sinusoidal or polynomial functions of time but have some difficulties that they are complex and require heavy computational burdens.

The trajectories suggested in this paper have the nature of a continuous polynomial function of time with smooth shape. Therefore a system which has to follow this trajectory requires the actuator's input which has the same nature. On the base of a polynomial-like trajectory generation method for HDD system suggested by H.Uchida and T.Semba (Uchida and Semba, 2002), generalized and expanded order of polynomial-like trajectory generation method was developed (Ahn *et al.*, 2003). The trajectory was named ZSPOT (Zero States POlynomial-like Trajectory) because it was designed with constraints that all final states are zeros. The method is based on the time varying state feedback and itself has the dynamics of the multiple integrator systems. This concept regards a trajectory as an independent system. ZSPOT has constraints of arbitrary initial states but zero final states. This means a system which follows ZSPOT should be stop at final time. But, there are many cases in the field which requires change of movement during one given motion, for examples, dynamic obstacle avoidance of a mobile system and moving of an industrial manufacturing system on several set points in possible minimum time, etc. ZSPOT can be used for these cases because the states at changing time can be given as the initial states to a new ZSPOT as shown by experiment in (Ahn *et al.*, 2004*b*). And here shows a simulation result of ZSPOT applied to a mobile-manipulator system so that proves its applicability.

There are another cases in which a system hits an object, jumps in the air. These systems need a trajectory guiding to the non zero final states. A trajectory using the same method applied to the ZSPOT is designed and this can have all non zero final states conditions. It is named ASPOT (Arbitrary States POlynomial-like Trajectory). Because ZSPOT was expanded to handle the general order of position derivatives, ASPOT can handle any non zero acceleration, jerk and even more minute concept, too. ASPOT is applied to the linear motor system, an example of a 2^{nd} order system platform, and shows tracking performances in the case of non zero final states.

Without watching control issues behind these methods, here shows them only as new trajectory generation methods. The rest of this paper is arranged as follows. Section 2 summarizes ZSPOT. In section 3 it is expanded to ASPOT and using ASPOT, modified ZSPOT is shown. Section 4 shows the effects of ASPOT briefly. A simulation on the platform of a differential driven nonholonomic mobile system with two link manipulator is shown using ZSPOT and an experiment on the platform of linear motor system for a chip mounting device is performed using ASPOT in section 5. The conclusion and future work are discussed in section 6.

2. ZSPOT (ZERO STATES POLYNOMIAL-LIKE TRAJECTORY)

ZSPOT is a polynomial-like trajectory generation method with the computational load which is dramatically less than the typical polynomial method. And it is independent of the order of the polynomial.

Let's define each terms of below equations as follows.

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

at discrete time m.

When an acceleration has a type of a polynomial function of time m of order three, it can be written as

$$a[m] = c_1 \cdot m + c_2 \cdot m^2 + c_3 \cdot m^3 + a[0] \qquad (1)$$

at discrete time m. If the order of this polynomial is n, $a(m)$ has the highest order term of m^n.

With total sampling number N, ZSPOT generates each states using recursive discrete integration.

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

$$\begin{aligned} v[m] &= v[m-1] + a[m-1] \\ x[m] &= x[m-1] + v[m-1] \end{aligned} \qquad (3)$$

where time varying term,

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

and the constants have the relation with n as follows.

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

In the procedure for inducing equation (2) and (3), sums of powers of consecutive integers compose the base of the procedure. Recursion formula of finite differences helps finding the pattern from those sequences and the constants α, β, γ can be generalized (Ahn *et al.*, 2003). Through ZSPOT equations (2)$\sim$(4), all smooth states (acceleration, velocity and position) are extracted every sampling time. For the steady motion at final time, that is to say, for the smooth ending, ZSPOT has the constraint that all states at final time N are zeros. And this is why this method is named ZSPOT (Zero States POlynomial-like Trajectory). The position trajectories from ZSPOT show more rapid ending with higher order n. (Figure 1)

If a system is sampled at 1000 Hz and run in 2 seconds, N would be 2000. It is obvious that the higher the sampling frequency of the system, the more similar to the continuous polynomial trajectory is generated (Åström and Wittenmark, 1997; Franklin *et al.*, 1990).

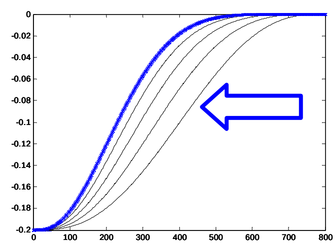

Fig. 1. Trajectory with varying order of ZSPOT_{acc}

Similar to but expanded with acceleration based procedure, a procedure begins with the jerk (the third time derivative of position) 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 result is

$$\begin{aligned} j[m] &= (1+\alpha \cdot K[m-1]) \cdot j[m-1] \\ &+ \beta \cdot K[m-1] \cdot K[m-2] \cdot a[m-1] \\ &+ \gamma \cdot K[m-1] \cdot K[m-2] \cdot K[m-3] \cdot v[m-1] \\ &+ \delta \cdot K[m-1] \cdot K[m-2] \cdot K[m-3] \\ &\cdot K[m-4] \cdot x[m-1] \end{aligned} \tag{5}$$

where,

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

Now there is one more constant than those of ZSPOT based on the acceleration (Table 1). These four constants have the general form as follows (Ahn, 2003).

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

where n is the order of jerk polynomial.

The ZSPOT method can be expanded to the more minute concepts like snap(the 4^{th}), crackle(the 5^{th}), pop(the 6^{th} derivatives of position) and more minute physical concepts in the same way shown above. An expansion to the "snap" was shown in (Ahn *et al.*, 2003). For distinction, let the ZSPOT from the acceleration polynomial be ZSPOT$_{acc}$ and the ZSPOT from the jerk polynomial be ZSPOT$_{jerk}$, etc.

The term which has to be calculated at each sampling time is just one (Ahn *et al.*, 2003). Therefore, for the trajectory generation on the system with its own cheap CPU or real time OS, this can significantly reduce the computational burdens. In addition, the computational load is regardless of the order of base polynomials. The effect on the computational burden was shown by showing very complicated equation of common polynomial in appendix of (Ahn *et al.*, 2004*b*).

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

3. ASPOT (ARBITRARY STATES POLYNOMIAL-LIKE TRAJECTORY)

The procedure to find the general equation of ASPOT is more complex than the case of ZSPOT. Non zero final states terms exist and it makes difficult to find general patterns in the formula.

From the ZSPOT$_{acc}$ but with non zero final state conditions, ASPOT$_{acc}$ is shown below.

$$\begin{aligned} a[m] &= (1+\alpha \cdot K[m-1]) \cdot a[m-1] \\ &+\beta \cdot K[m-1] \cdot K[m-2] \cdot v[m-1] \\ &+\gamma \cdot K[m-1] \cdot K[m-2] \cdot K[m-3] \cdot x[m-1] \\ &+\alpha_f K[m-1] \cdot K[m-2] \cdot K[m-3] \cdot a_f \\ &+\beta_f K[m-1] \cdot K[m-2] \cdot K[m-3] \cdot v_f \\ &+\gamma_f K[m-1] \cdot K[m-2] \cdot K[m-3] \cdot x_f \end{aligned} \tag{7}$$

After arranging,

$$\begin{aligned} a[m] &= (1+\alpha \cdot K[m-1]) \cdot a[m-1] \\ &+\beta \cdot K[m-1] \cdot K[m-2] \cdot v[m-1] \\ &+ (\gamma \cdot (x[m-1]-x_f) + \alpha_f \cdot a_f + \beta_f \cdot v_f) \\ &\cdot K[m-1] \cdot K[m-2] \cdot K[m-3] \end{aligned} \tag{8}$$

where,

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

TV means "time varying" and "the number of remained samples to target" in the program code. Three terms, α_f , β_f and K are the function of time. They are changed when time passes. Others are all constants. The procedure of finding patterns for all these terms was shown in appendix of (Ahn *et al.*, 2004*a*). The parameters related to the ongoing and final state conditions are shown below

$$\begin{bmatrix} \alpha \\ \beta \\ \gamma \\ \alpha_f \\ \beta_f \\ \gamma_f \end{bmatrix} = \begin{bmatrix} -3n \\ -3n(n+1) \\ -n(n+1)(n+2) \\ \frac{1}{2}TV(TV+1)n^3 - \frac{3}{2}TV(TV \\ +3)n^2 + ((TV+2)^2+2)n \\ n(n+1)\{6-TV(n-1)\} \\ -\gamma \end{bmatrix} \tag{9}$$

for the n^{th} order acceleration polynomial case. The parameters related with initial states and final position are constant. This property is sustained through all other types of ASPOT.

In the case of ASPOT$_{jerk}$,

$$\begin{aligned} j[m] &= (1+\alpha \cdot K[m-1]) \cdot j[m-1] \\ &+\beta \cdot K[m-1] \cdot K[m-2] \cdot u[m-1] \\ &+\gamma \cdot K[m-1] \cdot K[m-2] \cdot K[m-3] \cdot v[m-1] \\ &+ \begin{pmatrix} \delta \cdot (x[m-1]-x_f) + \alpha_f \cdot j_f \\ +\beta_f \cdot u_f + \gamma_f \cdot v_f \end{pmatrix} \\ &\cdot K[m-1] \cdot K[m-2] \cdot K[m-3] \cdot K[m-4] \end{aligned} \tag{10}$$

The parameters can be taken in the same way as the acceleration based case,

$$
\begin{bmatrix} \alpha & \beta & \gamma & \delta & \{\alpha_f\} & \{\beta_f\} & \gamma_f & \delta_f \end{bmatrix}^T =
\begin{bmatrix}
-4n \\
-6n(n+1) \\
-4n(n+1)(n+2) \\
-n(n+1)(n+2)(n+3) \\
\left\{ \begin{array}{l} -\frac{1}{6}TV(TV+1)(TV+2)n^4 \\ +TV(TV+1)(TV+4)n^3 \\ -\frac{1}{6}TV(11TV^2+69TV+130)n^2 \\ +(TV+4)(TV^2+3TV+6)n \end{array} \right\} \\
\left\{ \begin{array}{l} \frac{1}{2}TV(TV+1)n^4 - TV(TV+9)n^3 \\ -\frac{1}{2}(TV+9)(TV-8)n^2 \\ +(TV^2+9TV+36)n \end{array} \right\} \\
-n(n+1)(n+2)(nTV-TV-12) \\
-\gamma
\end{bmatrix}
\quad (11)
$$

for the n^{th} order jerk polynomial case.

It is important to remember that the minimum order n is three in the case of ASPOT$_{acc}$ and four in the case of ASPOT$_{jerk}$, and so on.

4. EFFECTS

The trajectories from ASPOT method is shown in figure 2. The final states are all non zero. More rapid steady settling with higher order of reference polynomial is shown. The x-axis is of sampling number and the y-axis is of acceleration, velocity and position from above.

ASPOT makes it possible to design a trajectory for a system to jump, hit or push. These actions need non zero acceleration or velocity at the end of the motions. The main purpose of ASPOT and ZSPOT is to guide smooth and continuous motion. All states generated by ASPOT and ZSPOT are smooth during whole run times. Concentrating on the fact that vibration or jerky motions may cause some wears for the system, suggested method guides the system more stable with less vibrations which may be caused by an abrupt reference like a step signal for quick motion change. By giving the order n higher, it can reduce the time for the system to get a desired states. The maximum value of n depends on the performance of the actuator of the system.

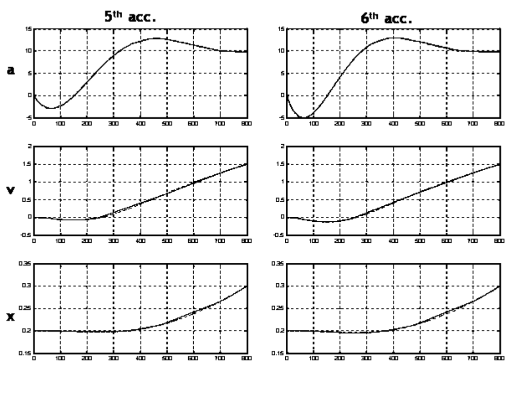

Fig. 2. ASPOT with 5^{th} and 6^{th} order acceleration cases

First of all, ASPOT and ZSPOT are the trajectory generation systems itself. Any state among them can be taken. For example, a position state is taken in the experiment of next section. Because ASPOT is the general case of ZSPOT, modified ZSPOT is obtained by making all final states except position be zeros. Changing $a_f = 0$ and $v_f = 0$ in equation (8) gives following ZSPOT$_{acc}$ equation.

$$
\begin{aligned}
a[m] &= (1 + \alpha \cdot K[m-1]) \cdot a[m-1] \\
&\quad + \beta \cdot K[m-1] \cdot K[m-2] \cdot v[m-1] \\
&\quad + \gamma \cdot (x[m-1] - x_f) \\
&\qquad \cdot K[m-1] \cdot K[m-2] \cdot K[m-3] \\
v[m] &= v[m-1] + a[m-1] \\
x[m] &= x[m-1] + v[m-1]
\end{aligned}
\quad (12)
$$

The effect on the computational burden shown with ZSPOT (Ahn *et al.*, 2004*c*) can be adopted to ASPOT, too. The application of ZSPOT to the case of changing reference point before one motion ends was shown by an experiment (Ahn *et al.*, 2004*b*). It is certain that ASPOT can be applied to the same situation with wider range.

5. SIMULATION AND EXPERIMENT

In order to avoid wheel slippage or mechanical damage during mobile robot navigation, it is definitely necessary to smoothly change driving velocity, acceleration or even jerk, if possible.

A planning methodology for nonholonomic mobile with manipulator is suggested. And because the generated trajectories are of polynomial nature which is continuous and smooth, ZSPOT is applied to the simulation of the same differential driven nonholonomic mobile system with two link manipulator in Fig. 3 (Papadopoulos and Poulakakis, 2001). As the quintic polynomial was used for the desired trajectory, 3^{rd} order ZSPOT$_{acc}$ is applied to generate trajectories for mobile wheels and manipulator links.

In the simulation, the running time is 6 seconds. The initial configuration is given by

$$
\begin{aligned}
&(x_F,\ y_F,\ x_E,\ y_E,\ \phi,\ \theta_1,\ \theta_2)_{initial} \\
&= (0.85,\ 0.67,\ 0.5,\ 0.5,\ -90^\circ,\ -30^\circ,\ -60^\circ)
\end{aligned}
$$

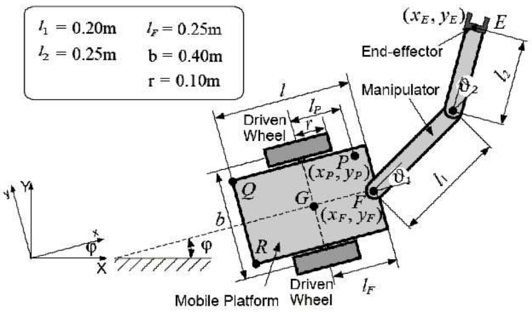

Fig. 3. Simulation model - a mobile with manipulator

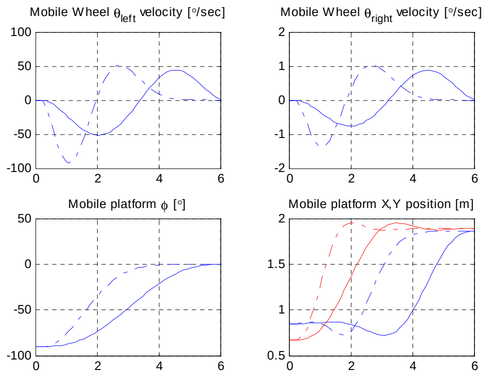

Fig. 4. Comparison for different orders of position trajectory

The final configuration is then given by

$$(x_F,\ y_F,\ x_E,\ y_E,\ \phi,\ \theta_1,\ \theta_2)_{final}$$
$$= (1.86,\ 1.89,\ 2,\ 2,\ 0°,\ -102.5°,\ 135°)$$

Through the kinematic modeling for this differentially driven mobile platform with two link manipulator , the angular velocities of the left and right wheels are calculated and plotted in Fig.4. To estimate the effect of ZSPOT with respect to the order of polynomial, 7^{th} order ZSPOT$_{acc}$ was simulated again. As shown in two upper graphs of Fig.4, higher order case(dash line) shows more maximum velocity but faster arrival than lower order case(line). The changes of orientation and cartesian coordinate of position of mobile platform are also plotted in the two lower graphs. Higher order case(dash line) shows more faster arrival than lower order case(line). During all running time, the motions are continuous and not jerky.

The system for the experiments of ASPOT is the one axis precision linear motor system used in the semiconductor chip mounting devices, which is shown in figure 5. The linear motor(ANORAD Corp., LEB–S–2–S–NC) is a direct drive motor without 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 (Kim and Chung, 2001) and PID control was applied to this system. When controlling the system to follow the given ASPOT trajectory as exact as possible, the shape of the acceleration state generated by ASPOT is proportional to the force of the system. To check this fact, a force sensor(ATI Industrial Automation, Delta SI-660-60) is attached on the system body. Figure 6 shows non zero final position, velocity and acceleration made by ASPOT. That is, the system sustain force at final time. Position is given as the reference set points to the system. It moves $0.2m$ in $3sec$. Because any ASPOT can generate the required states, ASPOT$_{jerk}$ is used for zero jerk at final time. During all running time, the motions are continuous and not jerky.

Fig. 5. Linear motor system

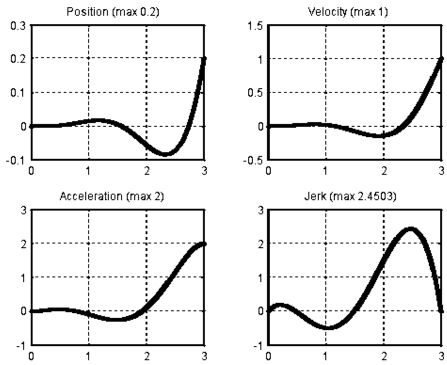

Fig. 6. Simulated trajectories of states using ASPOT$_{jerk}$

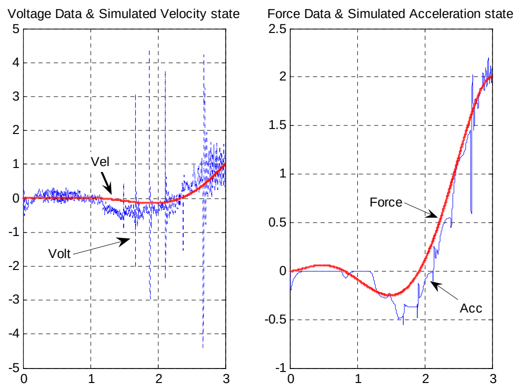

Fig. 7. Comparison of measured data with states from ASPOT

Figure 7 shows the plots of the simulated velocity and acceleration state data with measured input voltage and force data. Thick line is the simulated and thin line is the measured. Voltage input to the motor has close shape to the velocity state from ASPOT. After changing the sign and multiplying inertia factor, force data shows very close shape to the acceleration from ASPOT. This means that tested system is one of the 2^{nd} order systems.

It is clear that a system doing the ideal tracking needs no additional control. If a system has a smooth reference enough to follow, then cheap and simple controllers are sufficient.

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 was suggested for the case of zero final states conditions. ASPOT (arbitrary states polynomial-like trajectory) deals successfully with this constraint and allows arbitrary final state conditions.

Based on the recursive time-varying state feedback structure, these SPOTs can reduce computational burdens. The simulation and experiment apparently show that it is possible to reduce run time by easily changing the order of base polynomial without additional calculation load. Just changing to the higher constant related to the order of polynomial makes faster trajectory with keeping smooth and not jerky profile.

SPOTs are able to handle physically more minute concepts freely. Using some symbolic processes to induce general formula, the n^{th} derivative of position can be taken into account. The bigger or the smaller systems are, the more useful ASPOT (or ZSPOT) is.

Future works are remained. As shown in section 5, ASPOT can be used to identify the order and parameters of a system. Its application to more systems has to be followed.

ACKNOWLEDGMENT

This research was supported in part by a grant(02-PJ3-PG6-EV04-0003) of Ministry of Health and Welfare, by the International Cooperation Research Program (M6-0302-00-0009-03-A01-00-004-00) of the Ministry of Science and Technology, by the National Research Laboratory (NRL) Program (M1-0302-00-0040-03-J00-00-024-00) of the Ministry of Science and Technology, and by a grant(M1-0214-00-0116) of the Ministry of Science and Technology Republic of Korea.

REFERENCES

Ahn, Ki Tak (2003). A trajectory generation for model-following control based on smooth, rapid settling constraints. Master's thesis. (in Korean) POSTECH.

Ahn, Ki Tak, Wan Kyun Chung and Youngil Youm (2003). Reference model generation for tracking and ending in steady final state. In: *ICCAS'03, International Conference on Control, Automation and Systems*. pp. 101–106.

Ahn, Ki Tak, Wan Kyun Chung and Youngil Youm (2004*a*). Arbitrary states polynomial-like trajectory (aspot) generation. In: *IECON'04, The 30th Annual Conference of the IEEE Industrial Electronics Society, on CD-ROM WC4-4*.

Ahn, Ki Tak, Wan Kyun Chung and Youngil Youm (2004*b*). Zero states polynomial-like trajectory (zspot) generation. In: *ICAM'04, The Fourth International Conference on the Advanced Mechatronics*. pp. 120–125.

Ahn, Ki Tak, Wan Kyun Chung and Youngil Youm (2004*c*). Zero states polynomial-like trajectory (zspot) generation. In: *ICCAS'04, International Conference on Control, Automation and Systems*. pp. 1587–1592.

Åström, Karl J. and Björn Wittenmark (1997). *Computer-Controlled Systems*. 3rd ed.. Prentice-Hall,Inc.

Franklin, Gene F., J. David Powell and Michael L. Workman (1990). *Digital Control of Dynamic Systems*. Addison Wesley.

Hara, S., T. Hara, L. Yi and M.Tomizuka (1999). Novel reference signal generation for two degree-of-freedom controllers for hard disk drives. **5**(IATM), 73–78.

Kim, Bong Keun and Wan Kyun Chung (2001). Performance tuning of sliding mode controllers : Structural analysis approach. In: *Proc. 2001 American Control Conference*. pp. 1513–1518.

Mizoshita, Y., S. Hasegawa and K. Takaishi (1996). Vibration minimized access control for disk drives. *IEEE Trans.Magnetics* **32**(3), 1793–1798.

Papadopoulos, E. and I Poulakakis (2001). Planning and obstacle avoidance for mobile robots. In: *Proc. 2001 IEEE Int. Conf. on Robotics and Automation*. pp. 3967 –3972.

Uchida, Hiroshi and Tetsuo Semba (2002). 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.