

Research on Velocity Look-Ahead Algorithm of the 6-DOF Robot

Xiaowei NI^{1,a}, Jianqun LIU^{2,b}, Weiqiang GAO^{3,c}

¹⁻³ School of Electromechanical Engineering, Guangdong University of Technology, GuangZhou, 510006, China

^a157260398@qq.com, ^bjqliu@sina.com, ^c565262314@qq.com

Keywords: Cubic S-Curve; Adaptive Look-ahead Control; Industrial Robot; Continuous Jerk

Abstract: During the 6-DOF (6 degrees of freedom) industrial robot processing continuous short segments, frequent start-stop motions would cause vibration of joint and low playback efficiency. For the purpose of improving machining precision and playback efficiency, a cubic S-curve ACC/DEC (acceleration and deceleration) algorithm which derived with maximum acceleration was proposed. Besides, the adaptive velocity look-ahead algorithm is applied to machine continuous short segments. The velocity-time equation of cubic S-curve is third order derivable, which prevents step change in jerk, avoiding impact and vibration come from playback by industrial robot. The velocity look-ahead algorithm can be used to reduce the motion start-stop counts when the robot plays back and to improve processing stability and efficiency. Simulation result shows that this algorithm is high flexible and reduces the time of playing back significantly.

Introduction

High-speed, high-precision, high-stability is the outstanding feature of modern industrial robot, which are realized by velocity control algorithm. 6-DOF industrial robot gathers 3D machining segments by means of teaching mode. To achieve flexible control on playing back, it needs to use velocity planning on offline path. As we know, frequent start-stop motions would cause vibration of joints, which influences the machining precision and efficiency directly. Furthermore, there are many kinds of velocity control algorithms, linear, exponential and traditional seven-segment S-curve ACC/DEC algorithm are mostly used in robot system [1,2]. Acceleration is discontinuous in linear and exponential ACC/DEC algorithm, which cause flexible impact on robot joints [3,4]. The acceleration of traditional S-curve ACC/DEC algorithm is continuous and the velocity transition is smooth. But the Jerk transform suddenly under the certain stage [5]. It would be the reason for causing impact when the robot is working. And because of the complexity of seven-segment S-curve, it needs to discuss a variety of different situations when it applied to continuous short segments, which enlarges the work remarkably.

So far, many researchers have made deep investigation on velocity look-ahead control algorithm. In [6] and [7], the authors suggested using ACC/DEC control after interpolating to realize high speed manufacturing for continuous short segments. But this method is hard to ensure the relationship of each coordinate axis. In Zhang Bi-tao's master graduate dissertation, a five-segment S-curve ACC/DEC algorithm derived with maximum acceleration was proposed. This algorithm has an advantage of fewer calculation, but he has not consider the situation of the jerk would be oversize when the velocity is recalculated for short segment.

A cubic S-curve ACC/DEC algorithm was proposed in this paper, it keeps the flexibility of acceleration and jerk. The adaptive velocity look-ahead algorithm can adjust the number of the processing segments according to the length of the processed path. Simulation result shows that the

motion start-stop counts and the processing time were reduced after applying this algorithm to the industrial robot system.

The principle of cubic S-curve acceleration and deceleration algorithm

From Fig.1, defining v_s as the start velocity of the offline path, v_e as the end velocity of the offline path, v_m as the maximum allowable velocity of playing back, a_{ac} and a_{de} as the maximum acceleration in acceleration process and deceleration respectively. S_1 , S_2 and S_3 as the displacement of three stages of cubic S-curve respectively, and each stage takes the time T_1 , T_2 and T_3 . And v_m , a_{ac} and a_{de} are known quantity of robot system.

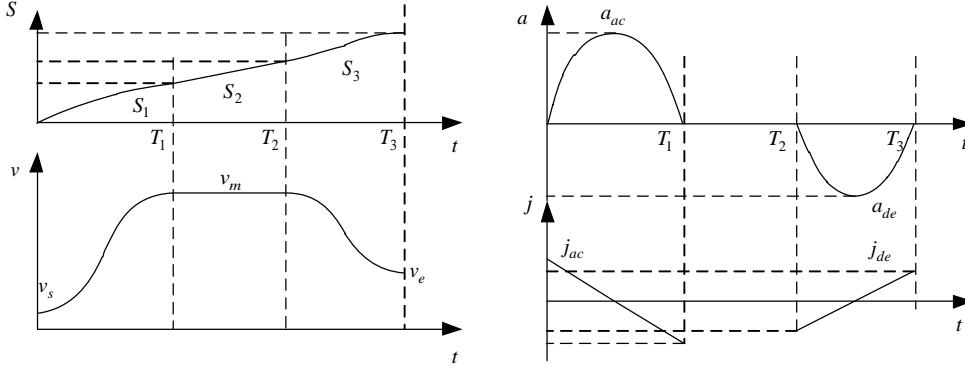


Fig.1. Cubic S-curve acceleration and deceleration mode

By the derivation, we have the equation of velocity-time and displacement-time.

$$v(t) = \begin{cases} -\frac{4a_{ac}}{3T_1^2}t^3 + \frac{2a_{ac}}{T_1}t^2 + v_s & t \in [T_0, T_1] \\ v_m & t \in [T_1, T_2] \\ \frac{4a_{de}}{3(T_3 - T_2)^2}(t - T_2)^3 - \frac{2a_{de}}{(T_3 - T_2)^2}(t - T_2)^2 + v_m & t \in [T_2, T_3] \end{cases} \quad (1)$$

$$S(t) = \begin{cases} -\frac{a_{ac}}{3T_1^2}t^4 + \frac{2a_{ac}}{3T_1}t^3 + v_s \cdot t & t \in [T_0, T_1] \\ v_m \cdot (t - T_1) + S_1 & t \in (T_1, T_2] \\ \frac{a_{de}}{3(T_3 - T_2)^2}(t - T_2)^4 - \frac{2a_{de}}{3(T_3 - T_2)}(t - T_2)^3 + v_m \cdot (t - T_2) + S_2 & t \in (T_2, T_3] \end{cases} \quad (2)$$

The velocity limited by included angle between two adjacent segments

In the case of Fig.2, defining \vec{v}_i as the velocity vector of segment L_i , \vec{v}_{i+1} as the velocity vector of segment L_{i+1} , $\Delta \vec{v}$ as the vector increment from velocity vector \vec{v}_i change to velocity vector \vec{v}_{i+1} . Due to $\Delta \vec{v}$ and \vec{v}_{i+1} are the velocity vector at the same point, they are equal in size ($|\Delta \vec{v}| = |\vec{v}_i| = |\vec{v}_{i+1}|$). By geometric analysis, we have:

$$|\Delta \vec{v}| = 2 \cdot |\vec{v}| \cdot \sin \frac{q}{2} \quad (3)$$

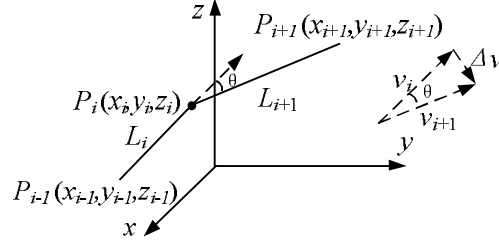


Fig.2. The relationship of velocity vector between two adjacent segment

To calculate the maximum safe linked velocity, set $|\Delta v| = a_m \cdot T_{\Delta v}$, according to Eq.3, we can calculate the maximum allowable velocity of included angle between two adjacent path.

$$veer = \frac{a_m \cdot T_{\Delta v}}{2 \cdot \sin \frac{q}{2}} \quad (4)$$

Where, a_m as the maximum acceleration of robot system, q as the included angle between the velocity vector of two path, T_{DV} as the time constant for veer. Set $T_{DV} = T_C$ (T_C as the interpolation period), $(I_{i,x}, I_{i,y}, I_{i,z})$ as the direction vector of segment i . By geometric analysis, we have:

$$\sin \frac{q}{2} = \frac{1}{2} \sqrt{(I_{i,x} - I_{i+1,x})^2 + (I_{i,y} - I_{i+1,y})^2 + (I_{i,z} - I_{i+1,z})^2} \quad (5)$$

According to Eq.4 and Eq.5, we can get the final equation of $veer$:

$$veer = \frac{a_m \cdot T_{\Delta v}}{\sqrt{(I_{i,x} - I_{i+1,x})^2 + (I_{i,y} - I_{i+1,y})^2 + (I_{i,z} - I_{i+1,z})^2}} \quad (6)$$

Sometimes the length of next machining segment can be interpolated in less than an interpolation period. In this case, we get the $veer$ according to the length of next machining segment. Set $T_{DV} = L_{i+1}/veer$, and substituted T_{DV} into Eq.4, we have:

$$veer = \sqrt{\frac{a_m \cdot L_{i+1}}{2 \cdot \sin \frac{q}{2}}} \quad (7)$$

The velocity limited by the length of the machining segments

With the decrease of the size of processing paths, the length of the segment before or after the current joint-point might not be enough to accelerate to the maximum velocity. Therefore, it is necessary to consider the length of the segment before or after the current joint-point.

The velocity limited by the length of the segment before joint-point. L_i as the length of the segment before the current joint-point. In a complete S-curve mode, suppose that throughout the segment S_{ac} , the velocity accelerates from v_s to v_m . According to Eq.1 and Eq.2, we have:

$$S_{ac} = \frac{3(v_m^2 - v_s^2)}{4a_{ac}} \quad (8)$$

When $L_i \geq S_{ac}$, the segment of sufficient length to accelerates from v_s to v_m . We have: $v_{m,ac} = v_m$.

When $L_i < S_{ac}$, the segment of insufficient length to accelerates from v_s to v_m . we have to recalculate the maximum velocity:

$$v_{m,ac} = \sqrt{\frac{4}{3} \cdot S_{ac} \cdot a_{ac} + v_s^2}$$

But above this section, we have:

$$v_{m,ac} = \begin{cases} v_m & L_i \geq S_{ac} \\ \sqrt{\frac{4}{3} \cdot S_{ac} \cdot a_{ac} + v_s^2} & L_i < S_{ac} \end{cases} \quad (9)$$

The velocity limited by the length of the segment after joint-point. L_{i+1} as the length of the segment after the current joint-point. In a complete S-curve mode, suppose that throughout the segment S_{de} , the velocity decelerates from v_m to v_e . According to Eq.1 and Eq.2, we have::

$$S_{de} = \frac{3(v_m^2 - v_e^2)}{4a_{de}} \quad (10)$$

When $L_i \geq S_{de}$, the segment of sufficient length to decelerates from v_m to v_e . We have: $v_{m,de} = v_m$.

When $L_i < S_{de}$, the segment of insufficient length to decelerates from v_m to v_e . So we have to recalculate the maximum velocity:

$$v_{m,de} = \sqrt{\frac{4}{3} \cdot S_{de} \cdot a_{de} + v_e^2}$$

But above this section, we have:

$$v_{m,de} = \begin{cases} v_m & L_i \geq S_{de} \\ \sqrt{\frac{4}{3} \cdot S_{de} \cdot a_{de} + v_e^2} & L_i < S_{de} \end{cases} \quad (11)$$

The design of adaptive velocity Look-Ahead algorithm

From the last two sections above, the linked velocity of continuous segments can be calculated:

$$v_{link} = \min\{v_m, v_{eer}, v_{m,ac}, v_{m,de}\} \quad (12)$$

The method to confirm the linked velocity is presented in this section below:

- 1) According to the limits of included angle between two adjacent segments and the length of current machining segment, the maximum velocity of look-ahead can be calculated;
- 2) Calculating the distance to decelerate from the velocity which had been calculated by step 1 to the minimum velocity of the robot system. And this distance would be regarded as the look-ahead distance to calculate the number of look-ahead segment. And then we can get the interval of look-ahead;
- 3) Start from the last segment in the interval of look-ahead, set the end velocity of this segment equal to the minimum velocity of the robot system. According to the length of each segment, the start velocity of each segment in the section of look-ahead can be calculated. Comparing the size of the start velocity and the included-angle velocity, taking the smaller one to regard as the end velocity of the above segment. By that analogy, the end velocity of current machining segment can be calculated finally.
- 4) Comparing the size of the end velocity, included-angle velocity and the velocity limited by

the length of current segment, taking the smallest one to regard as the end velocity of current machining segment. By that analogy, the start velocity and the end velocity of each segment can be calculated.

To the 3D continuous segments shown in Fig.3, the flow chart of adaptive look-ahead control is shown in Fig.4.

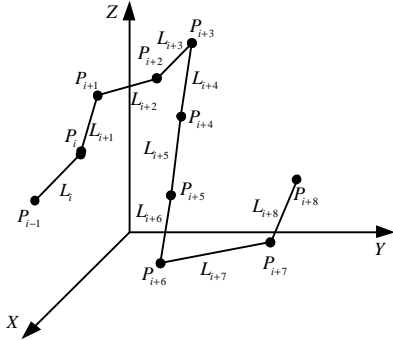


Fig.3. 3D continuous segments

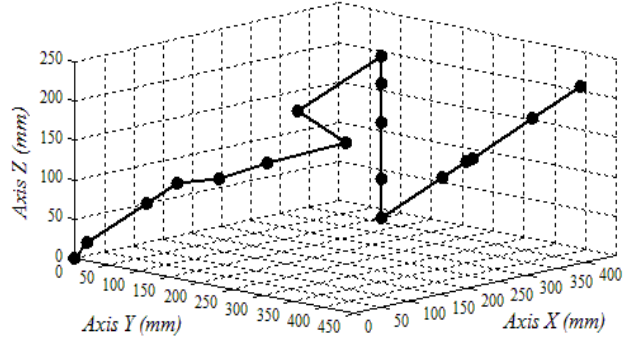


Fig.5. 3D machining segments in simulation

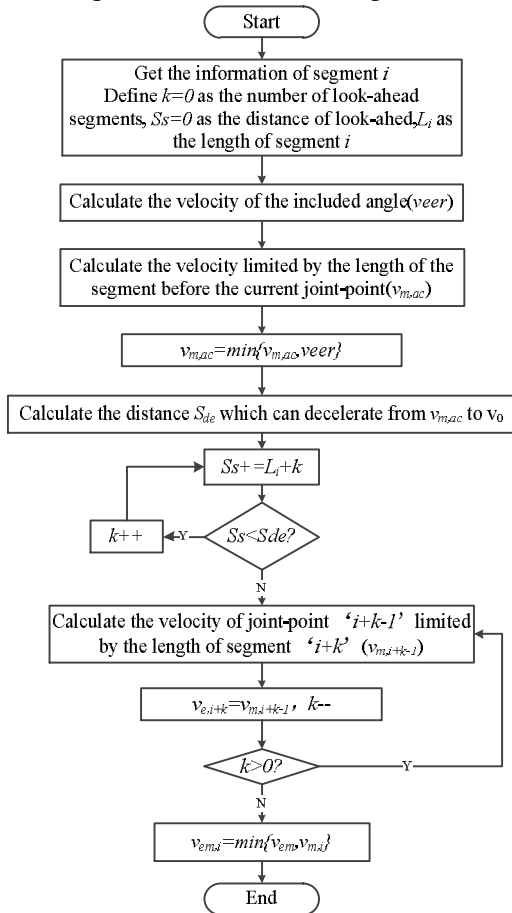


Fig.4. Flowchart of Look-ahead Algorithm

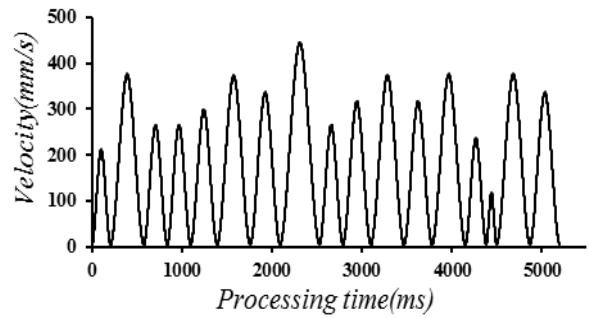


Fig.6. Velocity curve without Look-ahead Control

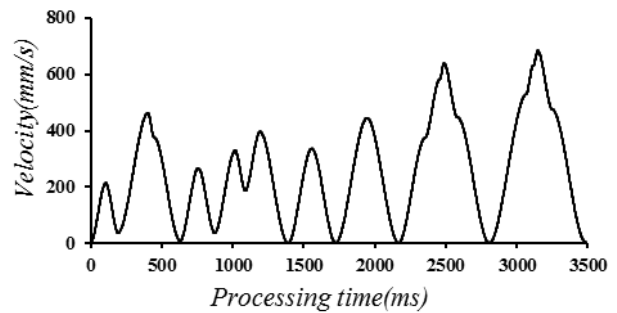


Fig.7. Velocity curve with Look-ahead Control

Experiment simulation

In the simulation, the parameters of the robot system are defined as the follow: the maximum velocity $v_m=1000\text{mm/s}$, the maximum acceleration and deceleration $a_{ac}=a_{de}=3000\text{mm/s}^2$, the interpolation period $T_c=1\text{ms}$. The 3D machining segments are gathered by the teaching mode of the industrial robot, as shown in Fig.5.

The velocity curve without look-ahead has shown in Fig.6. The end velocity and start velocity

are equal to the minimum velocity of the system. It took 5.199s when playing back. The motion starts and stops frequently, which against the stability of processing. Fig.7 shows the velocity curve with look-ahead. The overall speed was improved obviously, which took 3.485s, improving one third of efficiency. The results indicate that, the adaptive look-ahead algorithm on account of cubic S-curve acceleration and deceleration is a flexible velocity control algorithm, which reduces the motion start-stop counts, avoids the vibration and the impact on robot joints and reduces the processing time greatly.

Conclusions

To reduce the vibration causing by the motion starting and stopping frequently, an adaptive look-ahead algorithm on the base of cubic S-curve was proposed in this paper. We had analyzed and derived the displacement-time and velocity-time equation with the parameter of maximum acceleration. By setting up mathematical model and programming, the result shows that this algorithm is flexible, which avoids the impact causing by step change in acceleration and jerk. The motion start-stop counts and the machining time decreases obviously after using adaptive look-ahead algorithm for robot system.

Acknowledgements

This work was supported by the Science and Technology Planning Project of Guangdong Province (No.2015B010101013, No.2015B010102012), "Blue Ocean Talent Plan" Innovative and Entrepreneurial Teams Project of Nanhai District of Foshan, the University Industry Science Partnership (UNISPAR) Project of Guangdong Province and Ministry of Education of P.R. China (No.2013B090500125, No.2013B090500037), and the Postgraduate Demonstration Base Project of Department of Education of Guangdong Province (No.2013JDXM29). Meanwhile, as the first author, I would like to express my gratitude to all those who have helped me during the writing of this thesis.

References

- [1] Zhongqian Zheng, Xingfei Wang, Song Li. et al.: Journal of Mechanical & Electrical Engineering, Vol.31(2014)No.4, pp.425-430. (In Chinese)
- [2] Rongli Gai, Hu Lin, Liaomo Zheng. et al.: Journal of Chinese Computer Systems, Vol.30(2009) No.6, pp.1067-1071. (In Chinese)
- [3] Xingui Guo, Congxin Li: Journal of Shanghai Jiao Tong University, Vol.37(2003) No.2, pp.205-207. (In Chinese)
- [4] Haihong Pan, Wei Yang, Lin Chen. et al.: China Mechanical Engineering, Vol.21(2010) No.2, pp.190-195. (In Chinese)
- [5] Lin Wang, Jianfu Cao: Int J Adv Manuf Technol, Vol.63(2012), pp.705-717.
- [6] JEON J W: Electronics Letters, Vol.36(2000) No.8, pp.766-768.
- [7] Hongbin Leng, Yijie Wu, Xiaohong Pan: Mechanical Engineering, Vol.45(2009) No.6, pp.73-79. (In Chinese)