DOI: 10.21791/IJEMS.2019.1.61.

Speed Variation along an Imposed Linear Trajectory, for Robotic Arms Motion

L. M. Matica[1], Z. Kovendi[2], E. Gergely[3], L. Coroiu[4]

Faculty of Electrical Engineering and Information Technologies, University of Oradea;
Universitatii str., no.1, 410087, Oradea, Romania.
e-mail: lmatica@uoradea.ro[1], zkovendi@uoradea.ro[2], egergely@uoradea.ro[3], lcoroiu@uoradea.ro[4]

Abstract. The paper describes a method of speed (velocity) computation, named mixt profile, during a motion upon an imposed linear trajectory. The method assures an accurate positioning at the end of motion (movement), in a well determinate time lapse. The described method is linked with position vector computation, about a robotic arm.

Keywords: cinematics of robotic arm, vectors and versors, waypoints, acceleration or deceleration of movement, axle steps.

1. About movement of an robotic arm.

About movement command of a robotic arm [1; 2; 3], it is necessary to define direct and inverse cinematics. A cinematics analyze example of a robotic arm, type RRRRRR, is illustrated in fig.1.

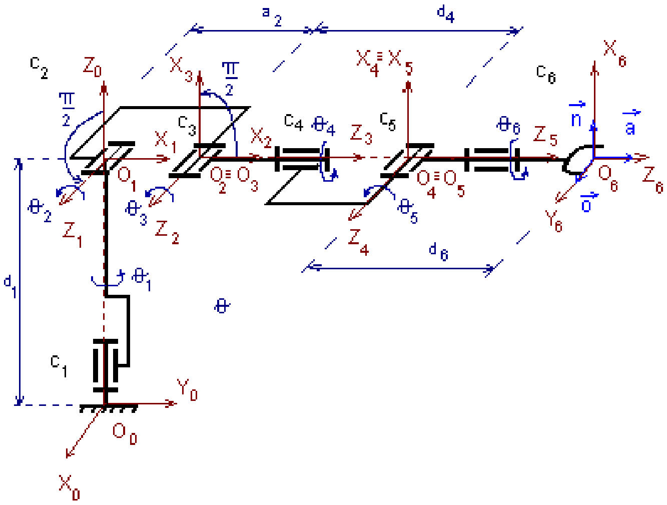


Fig.1. Cinematics analyze of an robotic arm, type RRRRRR.

The notations in fig.1 define several Cartesian coordinate systems with its axles: OX_i ; OY_i ; OZ_i and its origins: O_i (i=1..6). The motion of the robotic arm is determinated by six rotation driving cinematics couples (d.c.c.) named: C_i (index i goes from 1 to 6; i=1..6); the variable parameters of d.c.c. are named: Θ_i (i=1..6) ; there are constant parameters of the robotic arm named: d_1 ; a_2 ; d_4 ; d_6 . The orientation of

DOI: 10.21791/IJEMS.2019.1.61.

the robotic are is defined by the versors (a versor is a vectors with module equal to 1 value): $\vec{n}$; $\vec{o}$; $\vec{a}$ (the versors are identical with the sense and direction of axles OX$_6$; OY$_6$ and OZ$_6$ [3] (the index 6 Cartesian coordinate system has the origin in the tool point of the robotic arm (according with the Denawitt-Hartenberg convention). According to the Denawitt-Hartenberg convention, a Cartesian coordinate system, indexed *i*, is obtained by homogeneous transformations, from previous one, indexed *i-1*. Those homogeneous transformation are (always in this order):

1) rotation around OZi-1 axle;

2) translation along OZi-1 axle;

3) translation along OXi axle;

4) rotation around OXi axle.

For a rotation d.c.c. named C_i, the Cartesian coordinate system indexed i is obtained from Cartesian coordinate system indexed $i-1$ by homogeny transformations defined with formula that we wrote:

$$^{i-1}A_i = Rot(OZ_{i-1}, \theta_i(t) + \beta_i) \cdot Trans(OZ_{i-1}, a_i) \cdot Trans(OX_i, l_i) \cdot Rot(OX_i, \alpha_i) \tag{1}$$

The formula is described by the algorithm*:*

- the parameter $\theta_i(t)$ is the variable angle of the rotation C_i d.c.c. around axle OZ_{i-1}; it has different values at different moments of time, it a variable parameter;
- -the constant angle β_i has a different from zero value if axle OX_{i-1} is not parallel with axle OX_i; it express the rotation (around the axle OZ_{i-1}) of axle OX_{i-1} to obtain the axle OX_i; it depend of the robotic arm construction structure;
- -the constant distances a_i and l_i has a different from zero value if origin O_{i-1} is not identical with origin O_i (its define translations);
- -the constant angle α_i has a different from zero value if the axle OZ_{i-1} is not parallel with axle OZ_i; it express the rotation (around the axle OX_i) of axle OZ_{i-1} to obtain the axle OZ_i; it depend of the robotic arm construction structure.

The formula that we wrote for a translation d.c.c. is ,:

$$^{i-1}A_i = Rot(OZ_{i-1}, \beta_i) \cdot Trans(OZ_{i-1}, d_i(t)) \cdot Trans(OX_i, l_i) \cdot Rot(OX_i, \alpha_i) \tag{2}$$

, where the variable parameter is $d_i(t)$; it is the distance between origin O_{i-1} and origin O_i along axle OZ_{i-1} (the others parameters are identical as it was defined in rel.1).

For any robotic arm, those two formulas (rel.1 and rel.2) make easy the determination of the direct kinematics. It is only a mathematic problem to determine the inverse kinematics formulas.

2. Example of kinematics analyze.

About C1 d.c.c of the RRRRRR robotic arm, fig.1, the direct kinematics formula is:

$$ {}^{0}A_{1} = Rot(OZ_{0},\theta_{1}(t)+\frac{\pi}{2})\cdot Trans(OZ_{0},a_{1})\cdot Rot(OX_{1},+\frac{\pi}{2}) \tag{3}$$

The direct kinematic determine the $+\frac{\pi}{2}$ rotation angle around axle OZ_{0}, the a_{1} translation upon axle OZ_{0} and the $+\frac{\pi}{2}$ rotation angle around axle OX_{1} in concordance with the graphical explications, presented in fig.3:

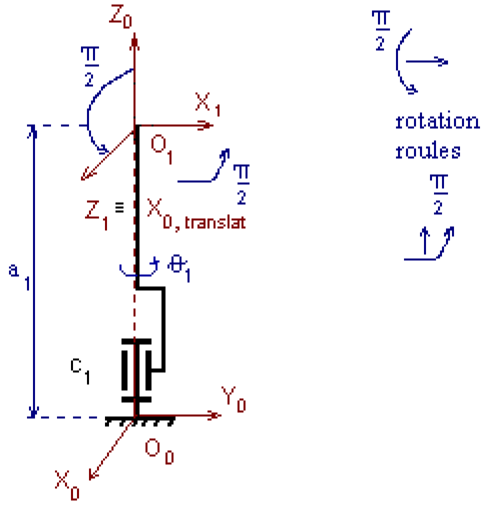


Fig.3. Graphical explication concerning C1 d.c.c.

The described algorithm may be applied for every d.c.c. [3].

3. The location matrix of a robotic arm.

The robotic arm position (location) is defined by the location matrix. It contains axles components of position vector, named $\vec{p}$ (it defines the position of tool centre point, TCP). This matrix contains axles components of orientation versors: $\vec{n}$; $\vec{o}$; $\vec{a}$ (orientation versors have the module equal to 1, its describe only the orientation (about the robotic arm);) fig.9.

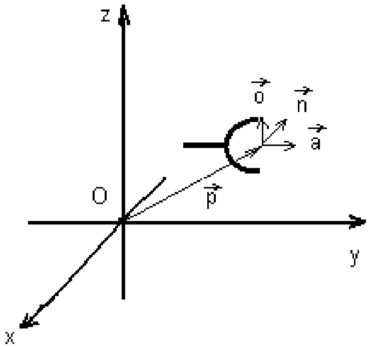


Fig.9. The position and orientations vectors that define the location matrix of an robotic arm.

DOI: 10.21791/IJEMS.2019.1.61.

So, the location matrix of the robotic arm, contains the axle components, indexed x; y ; z (along the three axles: OX; OY; OZ) of orientation versors and position vector:

$$G_i = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{9}$$

The relation between location matrix indexed i (refering to index i Cartesian coordinate system), and location matrix indexed i-1, (refering to index i Cartesian coordinate system) is:

$$G_{i-1} = {}^{i-1}A_i \cdot G_i \tag{10}$$

, where matrixs i-1Ai are defined by direct kinematics analyze, for example rel.3-8. According with Denawitt-Hartenberg convention, the matrix G6 is the neutral matrix for multiplication; the relation between location matrix index 0 and index 6 is:

$$G_o = {}^0T_6 \cdot G_6 = {}^0A_1 \cdot {}^1A_2 \cdot {}^2A_3 \cdot {}^3A_4 \cdot {}^4A_5 \cdot {}^5A_6 \cdot G_6 = {}^0A_1 \cdot {}^1A_2 \cdot {}^2A_3 \cdot {}^3A_4 \cdot {}^4A_5 \cdot {}^5A_6 \tag{11}$$

The inverse kinematics (as a result of direct kinematics) compute the d.c.c. parameters, named $\theta_i(t)$, starting with location matrix values.

4. Acceleration, motion on trajectory and deceleration.

The speed (velocity) profile of a motion is important for different reasons. Those reasons may be: speed of motion; time of motion; precision of motion; precision for reach the final motion point.

A linear motion trajectory, for the robotic arm, contains several intermediary positions, named waypoints. If the trajectory type is not required, intermediary points may be determined according to several conditions; for example it ensures no jerking. Another condition could be a certain programed speed.

About a robotic arm, if the trajectory is imposed (linear or circular), it is computed location matrices for intermediary points (waypoints). Considering inverse kinematics, the commands for every d.c.c. of the robotic arm are computed, starting with every location matrix that composes the trajectory. For example, the formulas for compute position vector components of robotic arm type RRRRRR (fig.1), (notations Si; i=1..6, means sine of Θi angle and Ci means cosine of same angle, the others notation are identical with those explained), are:

$$\begin{aligned} p_x = {} & (S_1 \cdot C_2 \cdot S_3 + S_1 \cdot S_2 \cdot C_3) \cdot C_4 \cdot S_5 \cdot d_6 + C_1 \cdot S_4 \cdot S_5 \cdot d_6 - S_1 \cdot C_2 \cdot a_2 + \\ & + (S_1 \cdot S_2 \cdot S_3 - S_1 \cdot C_2 \cdot C_3) \cdot (C_5 \cdot d_6 + d_4) \end{aligned} \tag{12}$$

$$\begin{aligned} p_y = {} & (-C_1 \cdot C_2 \cdot S_3 - C_1 \cdot S_2 \cdot C_3) \cdot C_4 \cdot S_5 \cdot d_6 + S_1 \cdot S_4 \cdot S_5 \cdot d_6 + + C_1 \cdot C_2 \cdot a_2 + \\ & + (C_1 \cdot C_2 \cdot C_3 - C_1 \cdot C_2 \cdot S_3) \cdot (C_5 \cdot d_6 + d_4) \end{aligned} \tag{13}$$

$$p_z = (-S_2 \cdot S_3 + C_2 \cdot C_3) \cdot C_4 \cdot S_5 \cdot d_6 + (S_2 \cdot C_3 + C_2 \cdot S_3) \cdot (C_5 \cdot d_6 + d_4) + S_2 \cdot a_2 + d_1 \tag{14}$$

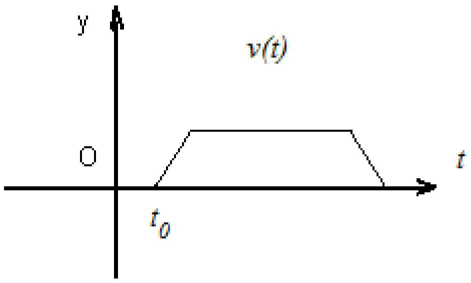
The motion of a robotic arm may contain three stages:

1)-the acceleration from zero motion speed to the imposed motion speed;

2)-the motion with imposed motion speed (constant);

3)-the deceleration from imposed speed to zero.

Commonly, acceleration to the imposed speed depends on the speed profile that was selected (trapezoidal or parabolic), fig.3 and fig.4 (those graphics consider continuous time)



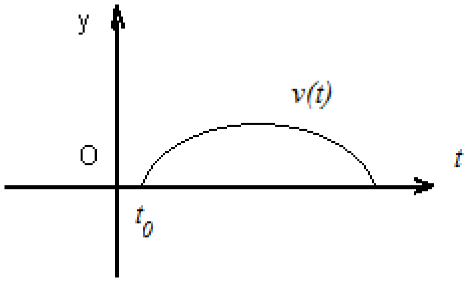


Fig.3. Trapezoidal profile (of speed). *Fig.4. Parabolic profile.*

This paper describes another method about deceleration stage; the method describes another speed profile, named mixt profile of speed, fig.5.

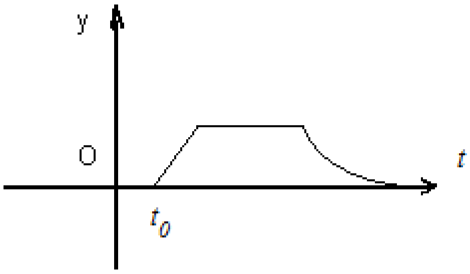


Fig.5. Mixt profile.

4. Acceleration and Deceleration Stages for Mixt Profile of Velocity

The acceleration variation depends of the maximum acceleration possible, on a sample period of time (in a numerical computation system with numerical processor). The numerical process of command computation, about a robotic arm movement, is a discrete one. The variation of robotic arm position, movements speed, acceleration and deceleration values depend of a discrete variable defined by relation: $k \cdot T$, where T is the sampling period of time, and k is the number of the sample periods of time considered (for example, the variable had the value $10 \cdot T$ after ten sampling periods of time from the start of movement).

In the computation described in this paper, it is considered the value of maximum possible acceleration in a sample period of time, named a_{max}. About this new method, the acceleration stage for mixt speed profile, is described by the relation:

$$v(kT) = v_0 + k \cdot a_{\max} \tag{15}$$

DOI: 10.21791/IJEMS.2019.1.61.

The initial value of movement speed is zero, it results $v_0 = 0$. After every sampling period of time, the pozition varies with the values $T \cdot v(kT) = T \cdot k \cdot a_{\max}$; this variation may be computed for every component of the pozition vector, considering the maximum possible acceleration along every axle, named a_M (instead of a_{max}) $k \cdot T \cdot a_M$.

Considering the initial pozition of the robotic arm defined by those components of pozition vector: $p_{0,x}$; $p_{0,y}$; $p_{0,y}$ after *k* sampling periods of time during the acceleration, the components of the position vector (indexed k), have the values:

$$p_{k,x} = p_{0,x} + k \cdot T \cdot a_M$$
$$p_{k,y} = p_{0,y} + k \cdot T \cdot a_M \qquad (16)$$
$$p_{k,z} = p_{0,z} + k \cdot T \cdot a_M$$

Considering the maximum value of motion speed, named v_M , the acceleration ends when $v_M = k \cdot T \cdot a_M$, this special value of k, named k_A , may be computed. The maximum value of motion speed defines the number of sampling periods of time for the acceleration, named k_A:

$$k_A = v_M / T \cdot a_M \qquad (17)$$

The value of kA must be an integer value (the value of kA must be adapted of this rounding, it is the next bigger integer value of the computed value).

It is a different situation for a movement with a programed speed, named vP . Considering its component vP,x ; vP,y ; vP,z ; the described method compute different values for each axle; the kA value is determined by the maximum component (considering each component of programed speed, as previous described in rel.10):

$$k_A = \max(v_{P,x}; v_{P,y}; v_{P,z}) / T \cdot a_M \qquad (18)$$

During the acceleration stage of motion, the position vector components are computed according withprevious considerations (usually, the components of initial speed, v0,x ; v0,y ; v0,z is zero):

$$p_{k,x} = p_{0,x} + k \cdot T \cdot (v_{P,x} - v_{0,x}) / k_A \quad , k = 1..k_A$$
$$p_{k,y} = p_{0,y} + k \cdot T \cdot (v_{P,y} - v_{0,y}) / k_A \qquad (19)$$
$$p_{k,z} = p_{0,z} + k \cdot T \cdot (v_{P,z} - vz) / k_A$$

In the previous relation k is an integer value, and goes from 1 to kA. The method may be applied for any values of initial speed of motion.

It result the logical conclusion: the movement of a robotic arm, with a programed speed is described by the relations (where k starts from k_A and goes till is necessary the deceleration stage, defined by the value named k_D):

$$p_{k+1,x} = p_{k,x} + T \cdot v_{P,x}$$
$$p_{k+1,y} = p_{k,y} + T \cdot v_{P,y} \qquad (20)$$
$$p_{k+1,z} = p_{k,z} + T \cdot v_{P,z}$$

The start of deceleration stage is defined by the value kD. The speed variation is described by the relationship, where aD is the deceleration:

$$v(kT) = v_P - T \cdot a_D(kT) = v_P - b \cdot k^2 \ , k = 1 ... k_D \tag{21}$$

In the previous relation, the value a_D is a variable value and b is a constant value adapted of desired characteristics about robotic arm motion. The deceleration decreases at the motion end. Considering the condition: $0 = v_P - b \cdot k_D{}^2$ and components of speed for each axle, it results the number of sampling period of time for the deceleration, named k_D:

$$k_D = \sqrt{\frac{\max(v_{P,x}; v_{P,y}; v_{P,z})}{b}} \tag{22}$$

The resulting speed profile is named mixt profile, fig.5 (the graphic considers continuous time). The described method ensures a better precision about stop point proximity. Typically, for a displacement with precise positioning at the end of a robotic arm, it can't be specified the time needed; the mixt profile of speed specifies exactly the time needed for the motion with precise positioning at the end.

The described method, named mixt profile (of speed), was implemented at a flexible welding cellule (for the manufacture of mining machinery), and the agreed motion characteristics (with the beneficiary) were ok.

For example, considering a linear trajectory and same orientation of robotic arm (along the movement), considering the values of programed speed: vP = $5\sqrt{2}$ mm/s; vP,x = 3mm/s; vP,y = 4mm/s; vP,z = 5mm/s; aM = 25mm/s2 and T = 10^{-2} s; an example of computation determines $k_A = 20$ of sampling periods of time for acceleration process:

$$k_A = \max(3;4;5) mm/s / (10^{-2} s \cdot 25 mm/s^2) = 20 \tag{23}$$

After the determination of k_A, the computation about the waypoints of the linear trajectory gets typical. Applying the difference analyzing algorithm[3] (about movement on a linear trajectory) the speed for each axle differs with values: δv_x = 3/20 mm/s; δv_y = 4/20 mm/s; δv_z = 5/20 mm/s; for each sampling period of time, and the position differs with values: $\delta p_x = 10^{-2} \cdot 3/20$ mm; $\delta p_y = 10^{-2} \cdot 4/20$mm; $\delta p_z = 10^{-2} \cdot 5/20$mm.

The number of sampling period of time for deceleration, considering b = 5mm/900s, is:

$$k_D = \sqrt{\frac{\max(3;4;5) mm/s}{5 mm/900 s}} = 30 \tag{24}$$

The computation of waypoints coordinates (during the deceleration) involves speed values:

$$\begin{aligned} v_x(kT) &= v_{P,x} - \frac{v_{P,x}}{k_D^2} \cdot k^2 = 3 - \frac{3}{900} \cdot k^2 \ , k = 1 ... k_D \\ v_y(kT) &= v_{P,y} - \frac{v_{P,y}}{k_D^2} \cdot k^2 = 4 - \frac{4}{900} \cdot k^2 \\ v_z(kT) &= v_{P,z} - \frac{v_{P,y}}{k_D^2} \cdot k^2 = 5 - \frac{5}{900} \cdot k^2 \end{aligned} \tag{25}$$

For each sampling period of time, the position differs with values:

$$\begin{aligned} &\mathrm{p}_x = T \cdot v_x(kT) \qquad , k = 1 \ldots k_D \\ &\delta \mathrm{p}_y = T \cdot v_y(kT) \\ &\delta \mathrm{p}_z = T \cdot v_z(kT) \end{aligned} \tag{26}$$

The previous example considered a linear trajectory.

A circular trajectory imposes the computation of waypoints on spherical coordinates (applying the difference analyzing algorithm) and conversion on Cartesian coordinates of those values.

5. Conclusions

The advantages of mixt profile (of motion velocity) are: the best precision to reach the end point of motion, minimum time of acceleration up to programed motion speed and exact determination of motion time.

The method may have others diverse applications, about motion on a linear or circular trajectory; for example about turning or milling process.

References

[1] L.M.Matica – H. Oros (2017) *Speed Computation for Robotic arms Motion Followed by Accurate Positioning*. International Journal of Computers, Communications & Control,http://univagora.ro/jour/index.php/ijccc/article/download/2785/1061 , accessed January 2018.

[2] L.M. Matica – Z. Kovendi (2011) *Structure Analysis for an Robotic arm*. Journal of Computer Science and Control Systems. Conference on Engineering of Modern Electric Systems 2011, Oradea. 4 (1) p. 89.

[3] T. Horsch – B.Juttler: *Cartesian Spline Interpolation for Robotic arms*. University of Technology, Departament of mathematics, Darmstadt, Germany (http://www.ag.jku.at/pubs/csi98.pdf)