Industrial robot optimization for required accuracy and speed

[1]*Peter PAPCUN (2nd year)*
Supervisor: [2]Ján JADLOVSKÝ

[2,3]Dept. of Cybernetics and Artificial Intelligence, FEI TU of Košice, Slovak Republic

[1]peter.papcun@tuke.sk, [2]jan.jadlovsky@tuke.sk

Abstract — **This paper describes design of industrial robot optimization for required accuracy and speed. I began with analysis in article Optimizing industrial robot for maximum speed with high accuracy at conference Modeling of mechanical and mechatronic systems. This article continues with analysis of robot movement and then paper dedicates design of mentioned optimization. Control and optimization of industrial robot's speed and accuracy is very important nowadays, for example in quality welding or application of sealants and many other examples. I am working with industrial robot MELFA RV-2SDB made by Mitsubishi.**

Keywords — **Industrial robot, speed optimizing, accuracy optimizing, required speed, required accuracy**

I. Introduction

This document analyzes robot movement and design robot optimization for required speed and accuracy based on mentioned analysis. Article [1] started with analysis of industrial robot movement.

This article is divided to seven chapters. The first chapter is introduction certainly. The second chapter describes design of movement point-to-point and present types of movements. Next chapter dedicates trajectories and their parametric expression. The fourth chapter describes results of movement point-to-point. The fifth chapter describes optimization methods for required speed and accuracy. The sixth chapter analyzes results of movement point-to-point and describes design of optimizing. The last chapter is conclusion.

II. Design of movement point-to-point

Paper [1] dedicates analysis of industrial robot movements on linear and circular trajectories. These movements use intern functions of robot controller. Article [1] describes data collections from industrial robot for movement analysis. I use this data collection again for movement point-to-point. I remind types of movements from paper [1], because I use same types of movements in method point-to-point:

- **SX** - linear movement in X-axis direction (coordinates Y and Z do not change during the motion)
- **SY** - linear movement in Y-axis direction (coordinates X and Z do not change during the motion)
- **SZ** - linear movement in Z-axis direction (coordinates X and Y do not change during the motion)
- **SA** - linear movement which all coordinates (X, Y, Z) change
- **CX** - circular movement in plane which is parallel with plane YZ
- **CY** - circular movement in plane which is parallel with plane XZ
- **CZ** - circular movement in plane which is parallel with plane XY
- **CA** - circular movement in the whole space (all coordinates is changing during motion)

Data analysis use mentioned tools in paper [1]: application for analyze robot accuracy and application for analyze robot speed. Movement speed does not be limited with any way. Point-to-point movement will be programmed with method, that every point will be exactly defined which robot's effector (endpoint of industrial robot) has to pass. These points will be calculated through parametric equation. Distance between points will be depended on step of iteration. I describe parametric equations of trajectories in the third chapter by this equations will be calculated points of trajectories.

III. Kinematics and trajectories

This chapter describes some equations of trajectories. I write expressions of trajectories in plane with functions and parametric equations. Trajectories are written through parametric equations in space. Parameter t can be time or step of iteration in parametric equations.

Line:

Plane: Function:
$$y = kx + q \qquad (1)$$
k – slope of a line
q – line shift

Parametric equation:
$$x = a + ct$$
$$y = b + dt \qquad (2)$$
[a,b] – point on line
(c,d) – direction vector of line

Space:
$$x = a + dt$$
$$y = b + et$$
$$z = c + ft \qquad (3)$$
[a,b,c] – point on line
(d,e,f) – direction vector of line

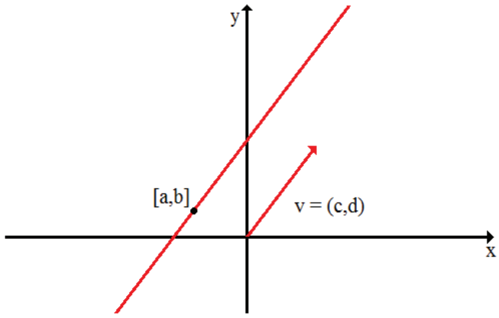

Fig. 1: Line on plane

Circle on plane:

Equation: $(x-a)^2 + (y-b)^2 = r^2$ (4)

Parametric equation: $x = a + r\cos t$

$y = b + r\sin t$ (5)

[a,b] – centre of circle

r – circle radius

Ellipse on plane:

Equation (canonical form):

$$\frac{(x-x_0)^2}{a^2}+\frac{(y-y_0)^2}{b^2}=1 \qquad (6)$$

Parametric equation: $x = x_0 + a\cos t$

$y = y_0 + b\sin t$ (7)

$[x_0, y_0]$ – centre of ellipse

a, b – length of semi-axis

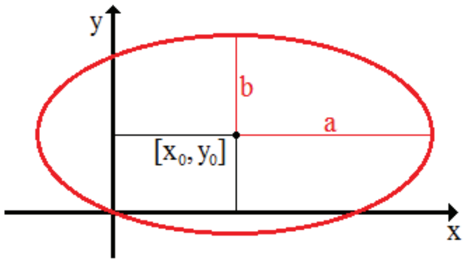

Fig. 2: Ellipse on plane

Circle and ellipse on space:

Parametric equation: $x = x_0 + a\cos t$

$y = y_0 + b\sin t$

$z = z_0 + c\cos t$ (8)

$[x_0, y_0, z_0]$ – centre of ellipse or circle

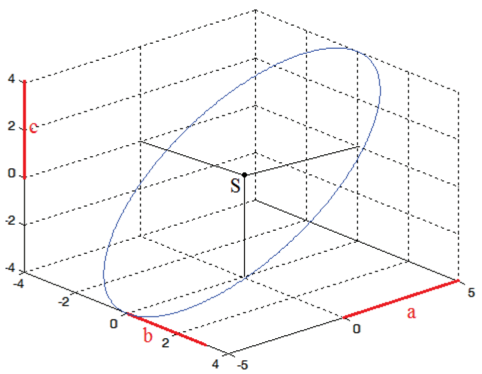

Fig. 3: Ellipse on space

Spiral:

Parametric equation: $x = x_0 + t\cos t$

$y = y_0 + t\sin t$ (9)

Helix:

Parametric equation: $x = x_0 + r\cos t$

$y = y_0 + r\sin t$

$z = kt$ (10)

k – twist

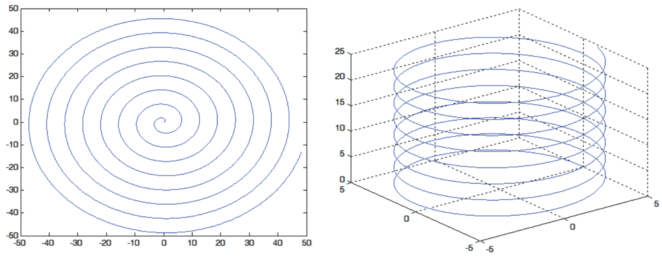

Fig. 4: Spiral on plane and helix on space

Trajectories are many others, but in this part are only some. Many others trajectories can be mathematical expressed, through parametric equation.

IV. Movement Point-to-Point

I test line movement at the first. I determine specific parametric equations of lines, intervals and steps of iteration (t). Simple program (in program language C) calculates points of movement from defined specific parametric equation (3). Step of iterations are determined gradually so that distance between points are 0.01 mm, 1 mm, 5 mm, 10 mm, 25 mm, 5 cm, 1 dm.

If distance was 0.01 mm, than industrial robot was maximally accurate (maximum accuracy is 0.01 mm – sensors restriction), but speed was very low, lower then 1 mm/s. When distance was 1 mm, than industrial robot was maximally accurate and average speed was 61 mm/s, these speeds was in interval from 47.04 mm/s to 71.23 mm/s. If distance was 10 mm, robot accuracy was 0.1 mm and average speed 231 mm/s. The highest distance was 1 dm. When distance was 1 dm, then speeds was higher than was measured in [1], but accuracy was very low (2 mm and near of system centre 2 cm). We can see a summary of results in the following table:

Table 1: Line movement which use method point-to-point

Typ. of mov.	Dista-nce [mm]	Max. deflect. [mm]	Min. speed [mm/s]	Max. speed [mm/s]	Average speed [mm/s]	Measured speed [mm/s]
SX	1	0.00	60.10	64.23	63.57	771.56
	10	0.06	230.53	241.42	237.21	
	100	0.98	761.47	863.15	821.32	
SY	1	0.00	60.61	63.22	62.17	639.54
	10	0.05	233.40	242.46	236.51	
	100	1.02	645.41	701.12	674.74	
SZ	1	0.00	66.51	71.23	68.27	832.18
	10	0.08	203.80	213.07	209.45	
	100	2.45	830.47	915.12	875.24	
SA	1	0.00	47.04	51.71	49.21	882.45
	10	0.10	235.20	259.38	242.32	
	100	18	895.12	952.13	924.21	

Notes to tables 1 and 2:

Typ. of mov. – type of movement

Distance – distance between the nearest points of trajectory

Max. deflect. – the highest measured movement error (deflection)
Min. speed – the highest measured speed
Max. speed – the lowest measured speed
Average speed – average speed of all measured speeds
Measured spd. – the highest average speed measured in [1] (movements which use intern functions of robot controller)

Then I test circular movement. Program (in program language C) calculates points of circular movements, through parametric equations of circular trajectories (7, 8). Iteration steps are determined as in the previous case so that distance between points of trajectory are 1 mm, 5 mm, 10 mm, 25 mm, 5 cm, 1 dm. Measurement results:

Table 2: Circular movement which use method point-to-point

Typ. of Mov.	Distance [mm]	Max. deflect. [mm]	Min. speed [mm/s]	Max. speed [mm/s]	Average speed [mm/s]	Measured speed [mm/s]
CX	1	0.00	61.66	68.84	65.14	606.47
	10	0.09	225.78	236.21	232.29	
	100	8.1	690.68	701.64	696.12	
CY	1	0.00	60.31	68.21	64.61	529.5
	10	0.08	226.82	235.16	231.45	
	100	9.4	681.18	692.38	687.23	
CZ	1	0.00	57.47	65.22	61.11	554.21
	10	0.15	209.24	225.42	216.47	
	100	12.48	573.99	583.85	579.35	
CA	1	0.00	51.57	59.68	56.16	599.51
	10	0.27	226.79	243.53	234.27	
	100	24.17	643.23	672.34	654.58	

We can see in this table, that accuracy with distance 10 mm is worse than in previous case (linear movement) and accuracy with distance 1 dm is not accurate (accuracy 24.17 mm). If distance was 1 dm, average speed was better than was measured in [1], this is same result as in previous case. When we compare rows in tables 1 and 2 with same distance, we can see similar average speeds. This implies that speed depends on distance between points of trajectory regardless of movement type and this distance depends on iteration step.

V. Optimization

Effector speed will be optimized through density of defined points on required trajectory. These points will calculate online in program loop. This loop will be programmed in robot controller. Trajectory will be defined by parametric equation in mentioned loop. New point of trajectory will calculate in each loop. Previous chapter describe iteration step, this step will modify so that robot will have required speed and accuracy. User will be able to define either of parametric equation in function or text file, also equation which is not mentioned in the third chapter.

Industrial robot optimizes for required speed and accuracy will operate by three ways:

1. Optimization for required speed.
2. Maximum speed optimization for required accuracy.
3. Speed optimization for required accuracy and speed.

Optimization for required speed has defined speed as input. Program modify iteration step to value, that effector has required speed regardless of movement accuracy. The first iteration step modifies according point-to-point movement analysis, which we can see in next chapter. If speed does not be equal with defined speed after this modify, than program start with fine tuning of iteration step.

Maximum speed optimization for required accuracy has defined accuracy as input. Program modify iteration step to value, that effector has required accuracy. The first iteration step modifies according point-to-point movement analysis, too. If accuracy does not be equal or better with defined accuracy after mentioned modify, than program reduce iteration step. If accuracy is equal or better with defined accuracy, than program increase iteration step minimally, until effector exceed accuracy. If program get twice same value of iteration step, then robot has maximum speed with required accuracy.

Speed optimization for required speed and accuracy has defined speed and accuracy as input. This optimization works same as optimization for required speed. If robot have required speed after optimization for required speed, then program check accuracy. If accuracy is equal or better with defined accuracy, then optimizing is at the end. When accuracy does not be equal or better with defined accuracy, program informs about this fact. When we would like required accuracy, we can continue with maximum speed optimization for required accuracy.

VI. Analysis of Movement Point-to-Point

We can see graph on figure 5 where are relationship between trajectory points distance and effector speed. Effector speeds are average measured values from movement point-to-point (average values from table 1 and 2).

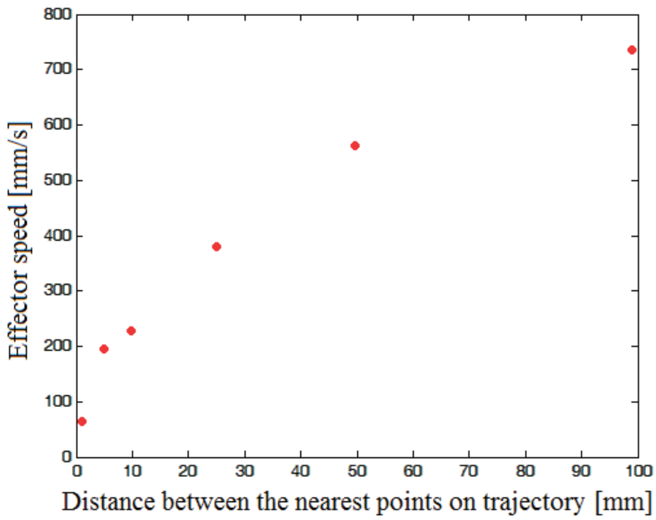

Fig. 5: Relationship between effector speed and trajectory points distance

Values which are in graph (Fig. 5):

Table 3: Values in graph, relationship between speed and distance

Distance [mm]	1	5	10	25	50	100
Speed [mm/s]	61.3	193.1	230.0	380.5	565.6	739.1

I approximate points on graph by method of least squares. I mark effector speed as v and distance between the nearest points on trajectory as s. Result is this function:

$$v = -0.0657s^2 + 13.0036s + 92.6894 \qquad (11)$$

We can see approximated function (11) on figure 6:

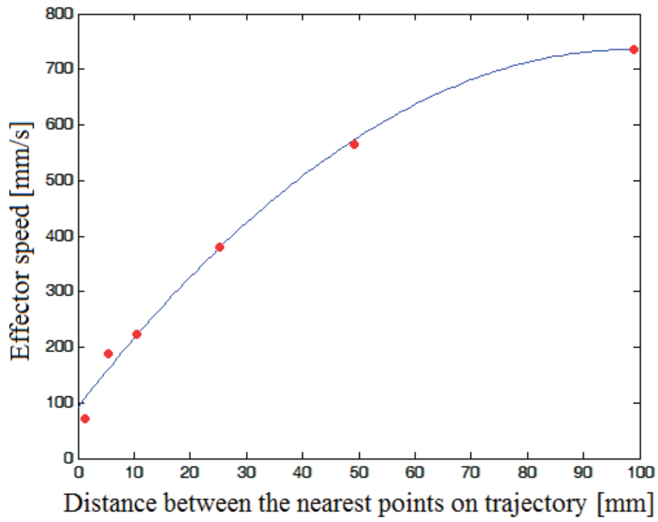

Fig. 6: Relationship between effector speed and trajectory points distance with approximated function

Now, we can define required speed and calculate distance s approximately, by approximated function (11).

I can do similar operation with accuracy. We can see average values of maximum deflection (the highest measured movement error) as accuracy from tables 1 and 2 in table 4:

Table 4: Relationship between trajectory points distance and accuracy

Distance [mm]	1	5	10	25	50	100
Accuracy [mm]	0.00	0.03	0.11	0.59	2.37	9.58

Approximated function to values in table 4 by method of least squares (p – accuracy, s – distance):

$$p = 0.001s^2 + 0.001s + 0.01 \qquad (12)$$

Points from table 4 and approximated function (12):

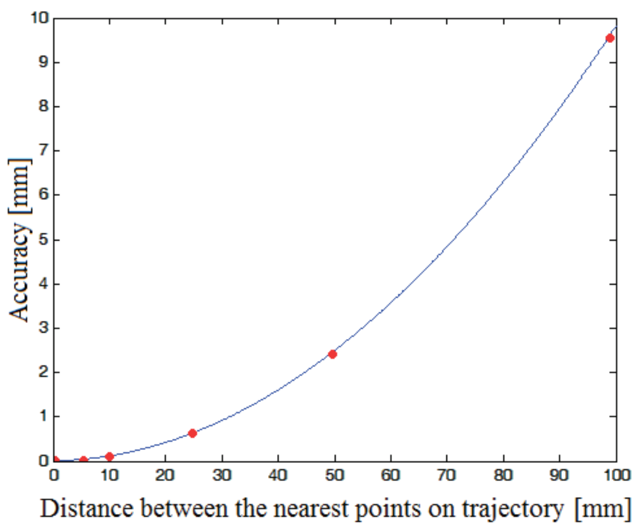

Fig. 7: Relationship between accuracy and trajectory points distance with approximated function

We can calculate distance s approximately from required accuracy by approximated function (12).

I described three method of optimization in previous chapter. I mentioned in each of them, that initial state of iteration step (the first iteration step modify) program calculate from analysis movement point-to-point. Optimization for required speed and speed optimization for required accuracy and speed use function (11) and maximum speed optimization for required accuracy use function (12). Program calculates distance between the nearest points on trajectory mentioned as s from functions (11) and (12). Trajectory is specified by parametric equation, so I can use formula (13) for calculate length of curve.

$$l = \int_0^{t_{\max}} \sqrt{x'(t) + y'(t) + z'(t)}\,dt \qquad (13)$$

When I know length of curve l and distance s, than I can calculate iteration step k (t_{max} is maximum value of parameter t from parametric equation of curve):

$$k = t_{\max} : \frac{l}{s} = \frac{t_{\max} s}{l} \qquad (14)$$

Now, I have calculated iteration step, this step will use first at optimizing effector speed and accuracy as initial state of iteration step. Next, program start tuning of iteration step, according procedure described in chapter five. Concept fine tuning mean modify iteration step about value t_{max}/l.

VII. Conclusion

I designed method of speed and accuracy optimization in this paper. At the first I analyzed robot movement by method point-to-point. Then I designed optimization. When I was designing optimization, I suggested concept initial state of iteration step. I can calculate initial step based on the measured values from analysis of movement point-to-point. This calculation is described in the sixth chapter. My research will continue with realization of this optimization design, also I would like design and realize diagnostic tool which will control speed and accuracy.

Acknowledgment

This work has been supported by the Scientific Grant Agency of Slovak Republic under project Vega No.1/0286/11 Dynamic Hybrid Architectures of the Multiagent Network Control Systems.

References

[1] JADLOVSKÝ, J. – PAPCUN, P. 2012. Optimizing Industry Robot for Maximum Speed with High Accuracy. In: Procedia Engineering. No. 48 (2012), p. 533-542. - ISSN 1877-7058
[2] PAPCUN, P. – ČOPÍK, M. – ILKOVIČ, J. 2012. Riadenie robota integrovaného v pružnom výrobnom systéme. In: ElectroScope. Vol. 2012, no. 2 (2012), p. 1-9. - ISSN 1802-4564
[3] MOSTÝN, Vladimír – KRYS, Václav: Mechatronika průmyslových robotů, Ostrava, 2012, ISBN 978-80-248-2610-3
[4] BOŽEK, Pavol – BARBORÁK, Oto – NAŠČÁK, Ľubomír – ŠTOLLMANN, Vladimír: Špecializované robotické systémy, Bratislava, 2011, ISBN 978-80-904766-8-4
[5] Instruction manual, CRnQ/CRnD Controller, Mitsunishi Electric, Ratingen, Germany, 2010.
[6] Product leaflets, RV-2SDB, Mitsubishi Electric, Ratingen, Germany, 2010.
[7] PAPCUN, P. 2011. Control of robot integrated in flexible production line, diploma thesis, Košice, Slovakia, 2011.