AUTOMATYKA/ AUTOMATICS • 2013 • Vol. 17 • No. 2

http://dx.doi.org/10.7494/automat.2013.17.2.271

Wojciech Zwonarz*

Time Stability of Computer Generated Control in Real-time

1. Introduction

This paper is devoted to the analysis of time stability of robot arm controller. The controller is implemented on the PC platform with Windows XP and Matlab/Simulink environment with additional toolbox RT-CON. The described robot is a 3 DOF Stanford Arm. It is powered with high torque, gearless electrical engines. More information about that robot can be found in [1]. Generating control by non RT OS introduces negative phenomenon called jitter. Jitter is undesired deviation from expected sampling time. The histogram of measured sampling times is shown in Figure 1. Additionally, there are external drivers of electrical engines in the main control circuit. Characteristics of drivers are unknown to the author. Unknown are their sampling times and internal dynamics. The analysis of the jitter effect influence on robot period movements was made. An experiment was made with PD and PD extended with neural network controllers. During the experiment a test of the influence of the stressed system on the control generation was made.

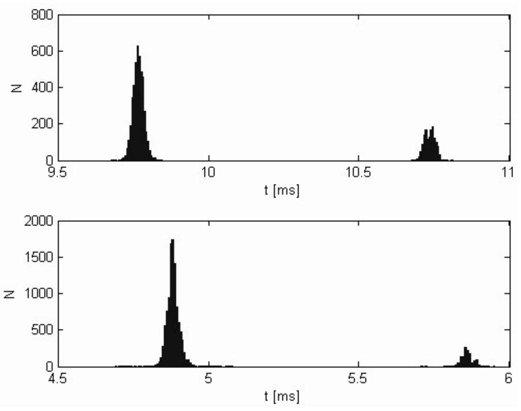

Fig. 1. Jitter in controller circuit, upper figure with set sampling time 10 ms, lower figure 5 ms, CPU not stressed with other jobs

* AGH University of Science and Technology, Krakow, Poland

2. Control generation

Algorithms for robot control are generated by a Matlab/Simulink platform, which is installed on a PC with Windows XP Professional SP3. The computer components are: CPU Intel Pentium 4 2,66 GHz and 1,25 GB RAM. The schema of examined stand is shown in Figure 2.

The Windows operating systems family is very popular on the PC market. Systems from that family are not RT systems, they do not hold hard time criteria. It means that control generated by the computer with one of that OS may by loaded with the jitter. To minimize the deviation from set sampling time specially designed Simulink toolbox was used. RT-CON toolbox is using autonomous system interrupts and does not allow Windows to pre-empt important to control generation tasks.

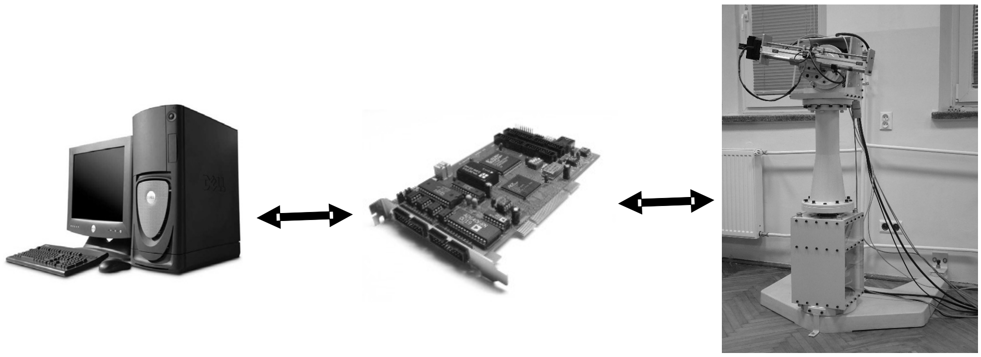

Fig. 2. Test station scheme, PC, RT-DAC extension card and Stanford manipulator

The additional task of the RT-CON toolbox is to provide cooperation between the Matlab/Simulink platform and the RT-DAC extension card. That card is the main input/output interface of the digital generated control. RT-DAC can generate analog output and has build in AD converter. The main part of card is re-configurable FPGA chip. It has implemented 32 bits counter which is counting ticks of card's clock. Thanks to the clock's robustness, accuracy and stability it is possible to measure time between control samplings. The clock frequency is 40 MHz (according to [2]), it gives 25 ns accuracy of measure. Way how the card is working and cooperate with Windows XP is described in [3].

Paper [3] explains why, thanks to internal card architecture, times measures by internal timer can by called sure. The comparison of the system timers and card timer made in the paper shown how unpredictable can system timers be. It is a highly undesirable property during control generation in RT. The paper's author included in it proposals of securing objects before such phenomenons in Windows OS. Applying their solutions, allow for safety use Windows/Matlab platform to control different objects in RT.

The actuators in the tested object are microprocessor drivers of the robot electrical engines. Because there are digital, drivers have their own sampling times and internal dynamics. That parameters are unknown to the author but the destination of the drivers (and the results of the experiment, described in the next parts of the paper) suggest that their values are negligible. Additionally, we can say their samplings times are precise and jitter is also negligible. At the same time, due to the fact at they are digital, can demonstrate awkward phenomenons following to unpunctuality of the control generation by the PC. At extreme case, when the change frequency of the set point is high, jitter can lead to object's undesirable behaviours.

3. Operating system unpunctuality

During tests of Windows XP with Matlab/Simulink platform punctuality, two sampling times were concerned: 10 ms and 5 ms. In addiction, both tests were performed with CPU in two states. In the first state, the processor was free from jobs other then control generation, in the second state the processor was in stress (external jobs obtained 30~40% processor time). Load used to be measured with system internal marker. The histogram of measured sampling times is shown in Figures 1 and 3, on vertical axis number amount of samplings were marked.

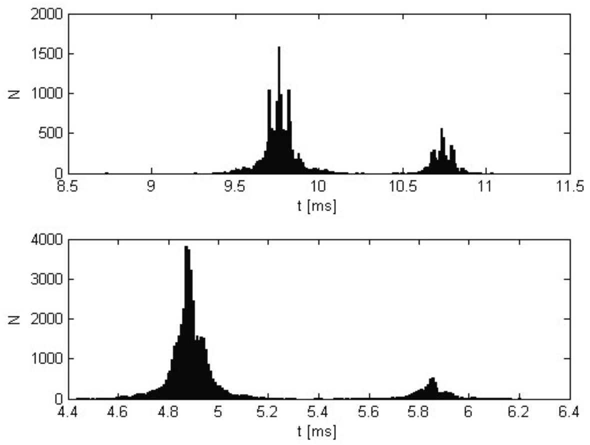

Fig. 3. Jitter in the control circuit, upper figure with set sampling time 10 ms, lower figure 5 ms, processor loaded in 40%

We can notice that Matlab/Simulink using large reserves of PC resources can hold sampling time constant, therefore the times are not the same as set once. Situation changes when CPU is in stress. In that case we observe the presence of time samples much distant from set points. In addiction, in both processor states we notice a large amount of samples moved in time with about 1 ms. A basic statistic analysis was performed, its results are presented in Table 1.

The results of the analysis allow to observe that despite the amount of samples moved in time with large deviation, average sampling times are almost the same as the expected values. We can, also, notice that for the 10 ms sampling period, the relative displacement of extremal sampling times is almost twice less than for the 5 ms sampling period. For both CPU load states, average sampling times and standard deviation do not change radically.

Table 1
Time deviation statistic

Sampling time	5 ms		10 ms	
CPU load rate	~0%	~40%	~0%	~40%
Average sampling time	4.99987 ms	4.99989 ms	9.99965 ms	9.99981 ms
Standard deviation	0.31762 ms	0.32287 ms	0.41363 ms	0.42610 ms
Minimal sampling time	4.69000 ms (–6.19%)	4.43260 ms (–11.34%)	9.67740 ms (–3.22%)	8.72580 ms (–12.74%)
Maximal sampling time	5.95140 ms (19.03%)	6.20320 ms (24.06%)	10.81060 ms (8.10%)	11.04219 ms (10.42%)

4. Period movement repeatability

To test object reactions on the unpunctuality of Windows XP, the experiment with robotic manipulator was performed. The robot was supposed to execute a previously planed period motion. Changes of the set motion period and its influence on the object were measured in real time. To eliminate the influence of robot inertia changes on trajectory repeatability, only the robot's column was moving. Rest of the robot's links were set in constant positions in a way providing the symmetry of the object. As referential trajectory the sinusoid was picked up. It is period, continues and the differential signal. It allowed to check how Windows XP and the robot manipulator are able to maintain signals stability in time.

During tests performances of two controllers were checked. The first one was classic PD controller and the second used to be PD+NN controller (with neural adaptive controller wither described in [1, 4]).

The sinusoid execution time was measured separately for positive and negative parts of the referential signal. This was done to prevent the results against the impact of the possible asymmetry of the friction forces. As referential signals were selected sinusoids with angular speeds equal ω = 5,6 rad/s and amplitudes respectively equal $\pi/6$, $\pi/8$. Both signals are rapidly alternating but do not require using high torques by electrical engine, what could introduce non-linearities to its characteristic (based on [5]). Simple calculation shows that periods of referential signals are respectively 1256.63 ms and 1047.17 ms, their positive (and of course negative) parts take 628.31 ms and 523.59 ms.

Results of trajectory tracking by PD and PD+NN controllers were comprised in Figure 6. The comparison shows in what way quality of control during single experiment. As the quality measure square error integer was used. We can assume that PD+NN controller minimizes quality measure much better then classic PD controller.

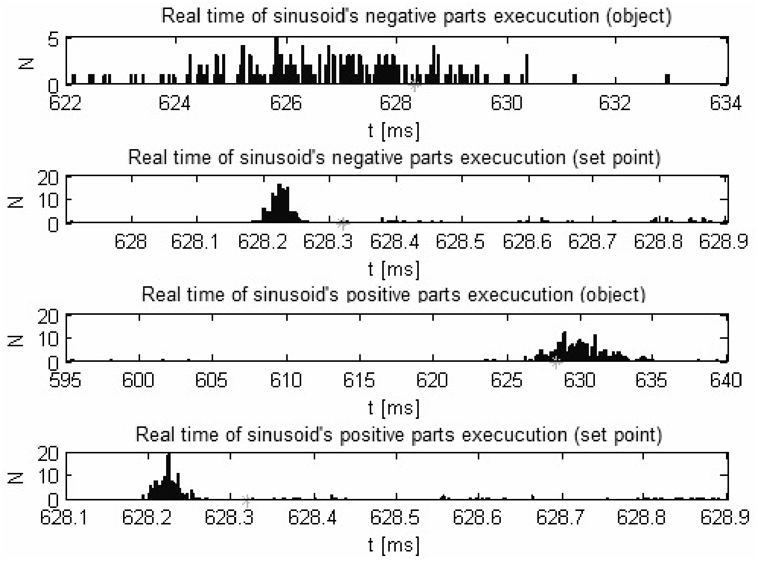

Fig. 4. Real time of sinusoid execution, 5 rad/s, PD controller

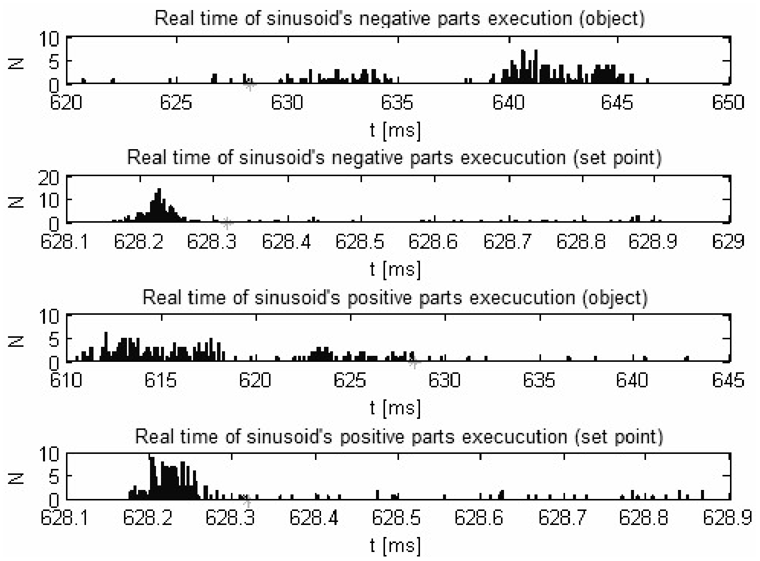

Fig. 5. Real time of sinusoid execution, 5 rad/s, PD+NN controller

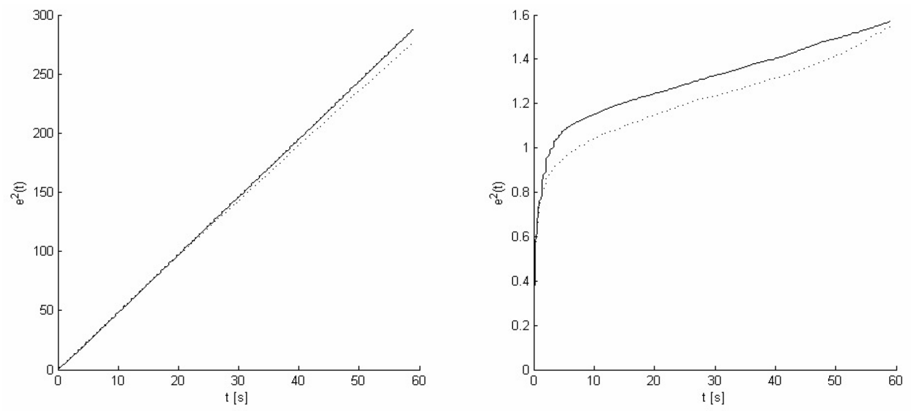

Fig. 6. Comparison of sample control's error. On the left, the figure PD controller, on the right the figure PD+NN controller. By the dotted lines marked measures with loaded CPU

Table 2
Sinusoid's execution time deviation statistic

Sinusoid type	$\omega = 6\left(\frac{\text{rad}}{\text{s}}\right)\ A = \frac{\pi}{8}$				$\omega = 5\left(\frac{\text{rad}}{\text{s}}\right)\ A = \frac{\pi}{6}$			
Controller	PD		PD+NN		PD		PD+NN	
Measure source	object	SP	object	SP	object	SP	object	SP
Average time of sinusoid's negative part	536.41	523.60	522.96	523.58	638.93	628.30	626.78	628.30
	527.11	523.59	×	×	636.32	628.30	627.38	628.30
Std. dev. time of sinusoid's negative part	8.93	0.45	4.36	0.45	5.44	0.19	1.83	0.18
	3.01	0.44	×	×	9.27	0.20	1.92	0.20
Max time of sinusoid's negative part	546.83	524.07	541.48	524.04	646.27	628.90	632.91	628.90
	535.92	524.10	×	×	666.13	628.93	631.94	628.96
Min time of sinusoid's negative part	513.57	522.90	503.84	523.00	620.73	628.16	622.13	627.91
	511.50	522.89	×	×	618.61	627.95	621.25	628.07
Average time of sinusoid's positive part	511.02	523.57	523.91	523.59	617.99	628.30	629.35	628.31
	520.23	523.58	×	×	620.46	628.30	628.87	628.31
Std. dev. time of sinusoid's positive part	9.31	0.45	7.48	0.45	6.09	0.17	4.86	0.18
	3.39	0.44	×	×	9.45	0.16	4.58	0.20
Max time of sinusoid's positive part	539.97	524.14	547.08	524.04	642.70	628.87	639.26	628.89
	538.95	524.14	×	×	644.26	628.91	639.34	628.99
Min time of sinusoid's positive part	500.34	522.98	468.70	523.00	610.49	628.18	595.37	628.19
	512.05	522.92	×	×	591.26	628.10	591.82	628.02

The histograms of measured sinusoids' halves times are shown in Figures 4 and 5, respectively for the control circuit with PD and PD+NN controller. The sampling period was set on 5 ms. Ideal executions times are marked with the stars. In Table 2 the results of the statistical study of the measures were collected. Upper parts of rows contain measures from the unloaded CPU, lower contains measures when the PC processor was in stress, all values are told in ms. The analysis of quality marker allowed to assume that PD+NN controller would maintain constant times of period trajectories much better then classic PD controller. The statistical analysis of measures showed that thesis is true. The PD+NN controller maintain stiff times of sinusoid's halves much better then classic PD controller.

The weak side of neural controller showed to be performance during work with loaded processor. Controller performance on stressed processor were not able to follow by referential, faster trajectory. We have to assume that theneural controller should not be used in the stressed processor state.

5. Summary

Measurements taken during the experiment on the real robotic manipulator proves, that Windows XP with, Matlab/Simulink platform can not be treated as hard real-time operating system. Therefore, toolbox RT-CON allow to treat Windows XP as soft-time criteria OS.

The operating system was not able to hold constant sampling period, extreme values of sampling time differ a lot from the set ones. The difference reached up to 1 ms, what takes a big part in the 5 ms sampling period. In the same time, the average sampling time differs from the set one on the 4th decimal place. Despite that long time measures of period movements showed discrepancies between the supposed and measured period of movement.

Experiment performed with robotic manipulator show that jitter on the observed level has minor influence on robot behaviour. Despite the same jitter in the main control circuit, using two different controllers change robot performance radically. We can assume that for the examined object more important then holding stiff sampling period is controller type in main control circuit.

References

[1] Marchewka D., *Algorytmy sterowania robotem za pomocą silników niskoobrotowych o dużym momencie napędowym.* Ph.D. Thesis. Department of Automatics, AGH, Kraków 2006.

[2] RT-DAC4/PCI Board, Users Guide, http://www.inteco.com.pl.

[3] Kołek K., Turnau A., *FPGA as a part of MS Windows control environment.* Proceedings of the International Multiconference on Computer Science and Information Technology, 2006, pp. 401–406.

[4] Lewis F.L., Campos J., Semic R., *Neuro-Fuzzy Control of Industry Systems with Actuator Nonlinearities*. SIAM, Philadelphia, 2002.

[5] Sadatoshi Kato., *Recent development of NSK Direct-Drive Motors*. No. 6, Japan 1999, pp 16–24.

[6] Żabiński T., *Sterowanie systemami mechatronicznymi w czasie rzeczywistym – podejście klasyczne i inteligentne.* Ph.D. Thesis. Department of Automatics, AGH, Rzeszów 2006.