The Real-Time Control Algorithm and Control Curves for Servomotors

P. Stodola, J. Mazal, I. Mokrá, and M. Podhorec

***Abstract*—**This article deals with the design and implementation of our real-time system for controlling servomotors. In the first part, the article presents the basic architecture of the designed system. The next part deals with principles of real-time communication between a servo drive and a system control unit and introduces an algorithm for servo position control. Then two types of control curves are introduced: (a) trapezoid and (b) S-curve. The paper compares and evaluates the main parameters and features for both types and presents the real examples of the control curves both in position and velocity axes. The last part is focused on applications of the authors in which the presented system is being presently used.

***Keywords*—**Servomotor, Real-time control, Microcontroller, Control curve, Trapezoid, S-curve.

I. Introduction

Nowadays servomotors can be found in a wide range of applications. Their utilization is common in machines and devices for industrial automation, e.g. in machining, automotive, rubber, food processing, glass, or construction industry for controlling robots, manipulators, manufacturing machines, CNC machines, packing machines, assembly machines, etc.

There are a lot of various types of servomotors, as well as possibilities of their control. At the present time the most common way for controlling servomotors is the real-time principle, i.e. they can be steered according to current requirements or as a reaction to unpredictable circumstances without unacceptable delay causing material, financial or other losses.

This article deals with the design and implementation of our own real-time multi-axis system for controlling servomotors without the necessity of buying a very expensive control system from producers. We have managed to design and construct a reliable system which is used for controlling devices being developed within our research projects [1], [2].

P. Stodola is with the University of Defence, Department of Military Management and Tactics, Czech Republic (email: Petr.Stodola@unob.cz).

J. Mazal is with the University of Defence, Department of Military Management and Tactics, Czech Republic (email: Jan.Mazal@unob.cz).

I. Mokrá is with the University of Defence, Department of Military Management and Tactics, Czech Republic (email: Ivana.Mokra@unob.cz).

M. Podhorec is with the University of Defence, Department of Military Management and Tactics, Czech Republic (email: Milan.Podhorec@unob.cz).

This article considers a topic that has practical implementation and that is of a real concern in number of research or industry applications. There are a lot of papers published in international journals and conferences dealing with utilization and implementation of manipulators, manufacturing machines or robots, e.g. [9], [10], [11], [12], [13], [14]. This paper can provide a unique system as a solution for their controlling.

II. System Architecture

Our system serves for controlling servomotors from the TG Drives company. This company offers its own real-time control system called TG Motion [3] which can be installed on the control unit (computer) with the operating system Windows XP. TG Motion provides a real-time control with precision in miliseconds. However it is very expensive and increases project costs considerably.

The TG Motion architecture is shown in Fig. 1. It is apparent that the link between the control unit and the servo device is conducted directly via a CAN bus or an industrial Ethernet bus EtherCAT. The real-time system TG Motion is installed on the control unit; communication with the control software proceeds via shared memory.

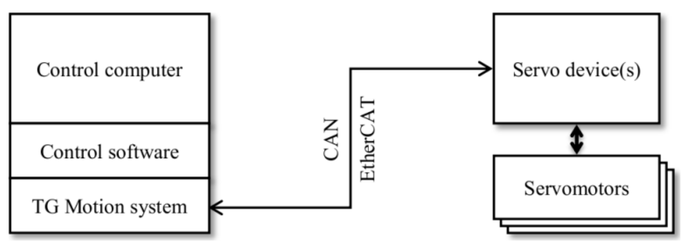

Fig. 1 TG Motion system architecture

Fig. 2 presents the architecture of our system. There is included a microcontroller Stellaris between the control unit and servo device; the microcontroller replaces the TG Motion system completely. Communication between the control software and microcontroller is established on a serial interface RS232 or Ethernet. The microcontroller is connected to the servo device via a CAN bus.

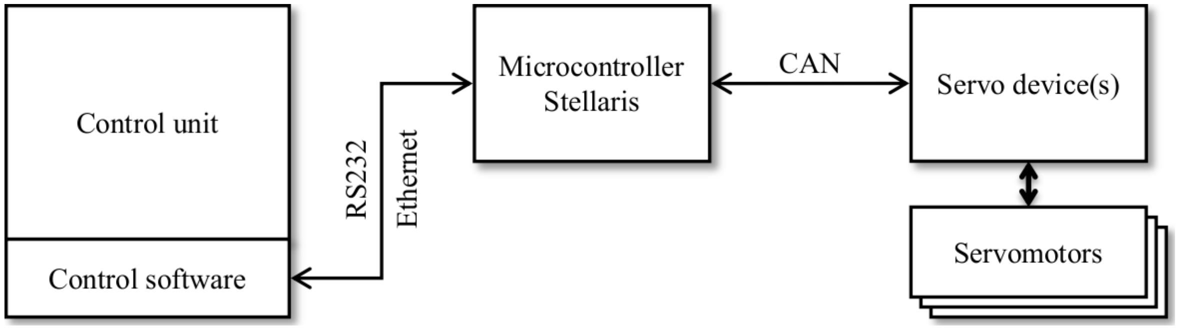

Fig. 2 Architecture of our system

III. Communication protocol

Communication between the microcontroller and servo device on a CAN bus [4] is shown in Fig. 3. In real-time position control commands SYNC are sent to the servo device in regular time intervals t_{sync} (1 to 8 ms according to user setting); up to three servomotors can be connected to one servo device.

Subsequently, the servo device sends ACTPOS messages containing current positions of all connected servomotors. The microcontroller has to compute new positions of the servomotors requested in the next step and send them to the servo device in time interval t_{pos} via NEWPOS messages. Then the whole cycle is repeated.

Communication is established on a serial interface RS232 or Ethernet and conducted by standardized messages. It does not have to run in the real time; only the microcontroller is responsible for the real-time messaging.

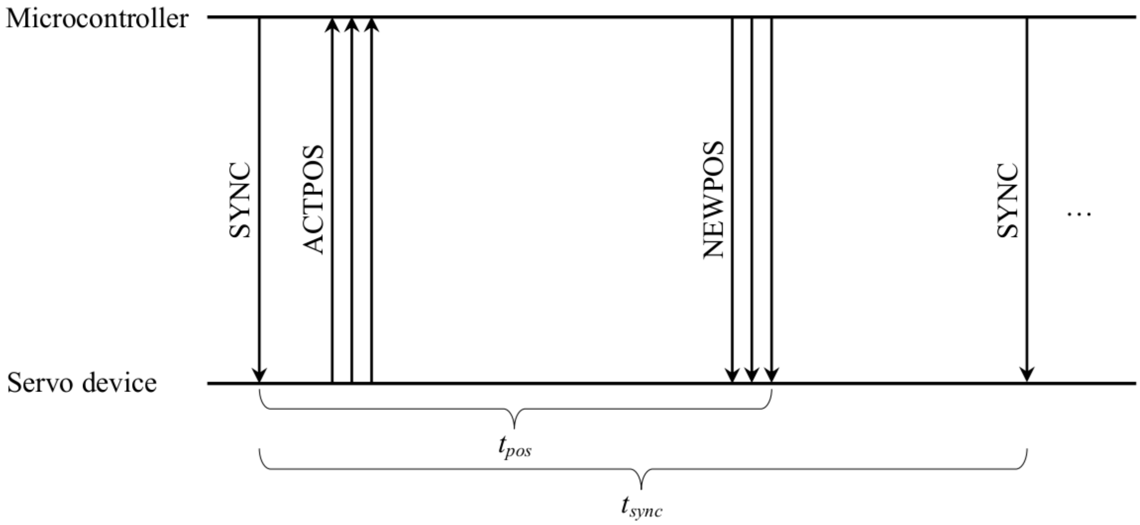

Fig. 3 Communication between the microcontroller and servo device

IV. Control algorithm structure

This part presents a simplified structure of our control algorithm (see Fig. 4) running in the microcontroller Stellaris (particularly the type LM3S8962 with clock frequency 50 Mhz [5]).

The algorithm is based on the above mentioned communication protocol. When the power is on, microcontroller's system variables, parameters, and busses (CAN and RS232 or Ethernet if needed) are initialized and the connection with the servo device is established.

The system timer interrupt regularly generates SYNC commands and sends them to the servo device in time intervals t_{sync} (particularly 4 ms in our application). The main loop of the algorithm waits for setting the *msg* variable signaling arrivals of ACTPOS messages from the servo device (messages are received in the CAN interrupt). After receiving the messages, new requested positions of all servomotors are computed and sent to the servo device via NEWPOS messages. Then the whole process is repeated.

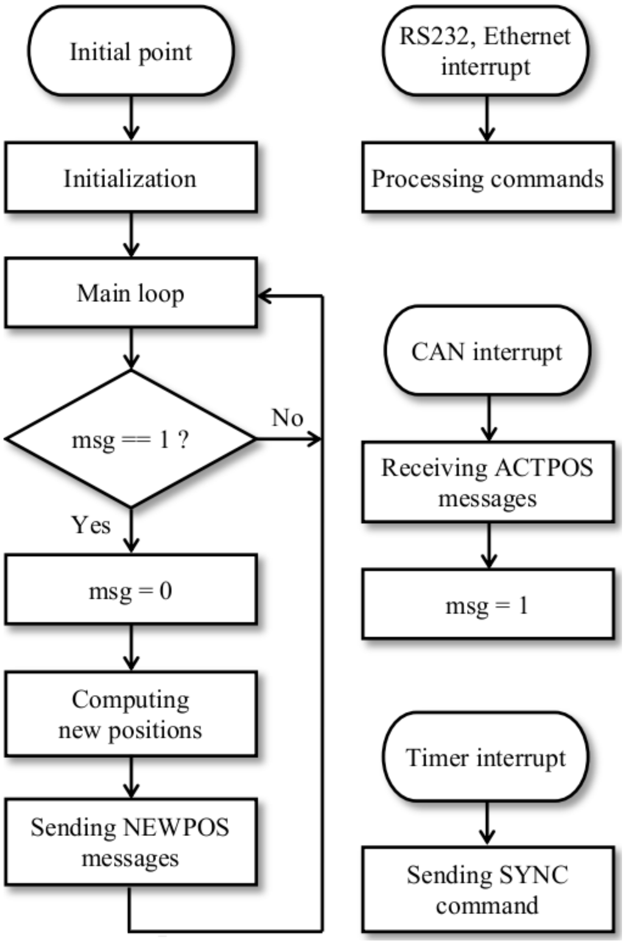

Fig. 4 Structure of the control algorithm

V. Control curves

The key element for real-time controlling is calculation of all servomotors' positions in each step of the main loop of the control algorithm. The aim is to reach the precisely defined target position s_r from the initial position s_0 as fast as possible, providing the maximal values of velocity (v_{max}), acceleration (a_{max}) and deceleration (d_{max}) are not exceeded. Maximum velocity limits the permitted rotational speed of servomotor; maximum acceleration defines the maximum change of rotational speed between two adjacent steps.

The control algorithm is responsible for meeting the conditions mentioned above; this algorithm determines the servomotor(s) position in each step according to the control curve. Trapezoidal curves [6] are used most often in real applications. However, some applications implement S-curves [7] which ensure lower instantaneous change of forces exerted on servomotors. It is of importance in applications with high rotational speed of servomotors [8]. The next text discusses both types of control curves.

A. Trapezoid

The trapezoid is the most common control curve for servomotors used in practice. Fig. 5 presents this curve in a velocity axis. There we can see the conditions graphically: maximum velocity (v_{max}) and acceleration (a_{max} for rising phase, d_{max} for falling phase; both values are mostly the same in practice). Parameter s_0 represents the position of the servomotor in an initial state, i.e. before the demand for the new position received from the control unit. Parameter s_r is the demanded position from the control unit to be reached.

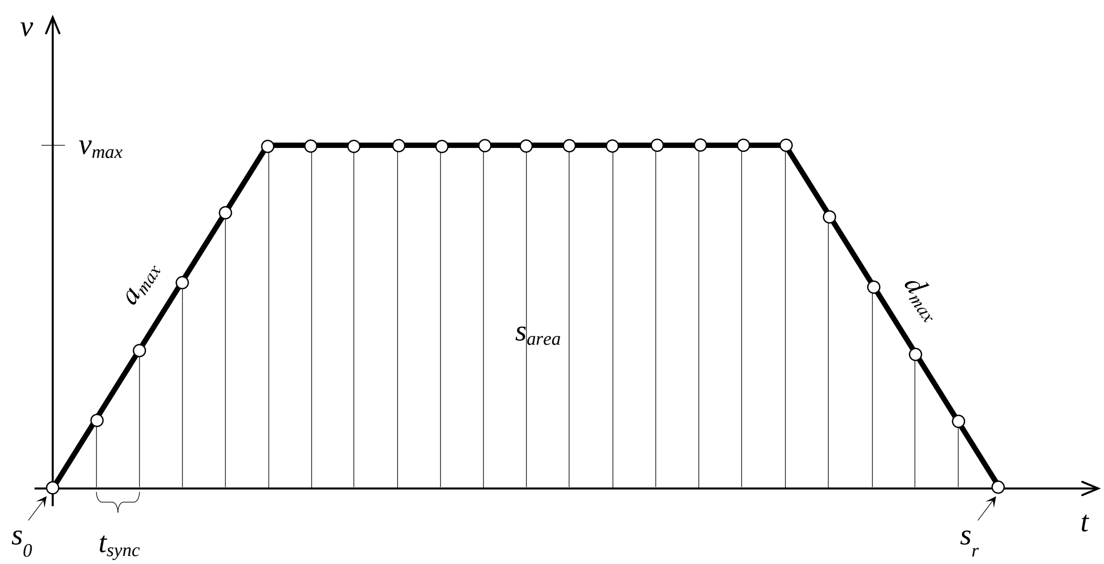

Fig. 5 Trapezoid curve for position control

The area under the curve (s_{area}) is given as an integral of instantaneous velocities in every step and has to be equal to the difference between the initial (s_0) and final positions (s_r) according to the formula (1).

$$s_{area} = |s_r - s_0| \quad (1)$$

Fig. 6 shows the control curve example adopted from our real application in the velocity and position axes (the scale of the position axis is lessened to display the whole curve). During control there were several position requirements from the control unit (called by symbols s_{r1}, s_{r2}, s_{r3}). The curve shows that the algorithm is able to control the servomotor fluently, even if the new position is required at the moment when the servomotor is not idle (i.e. without zero initial velocity).

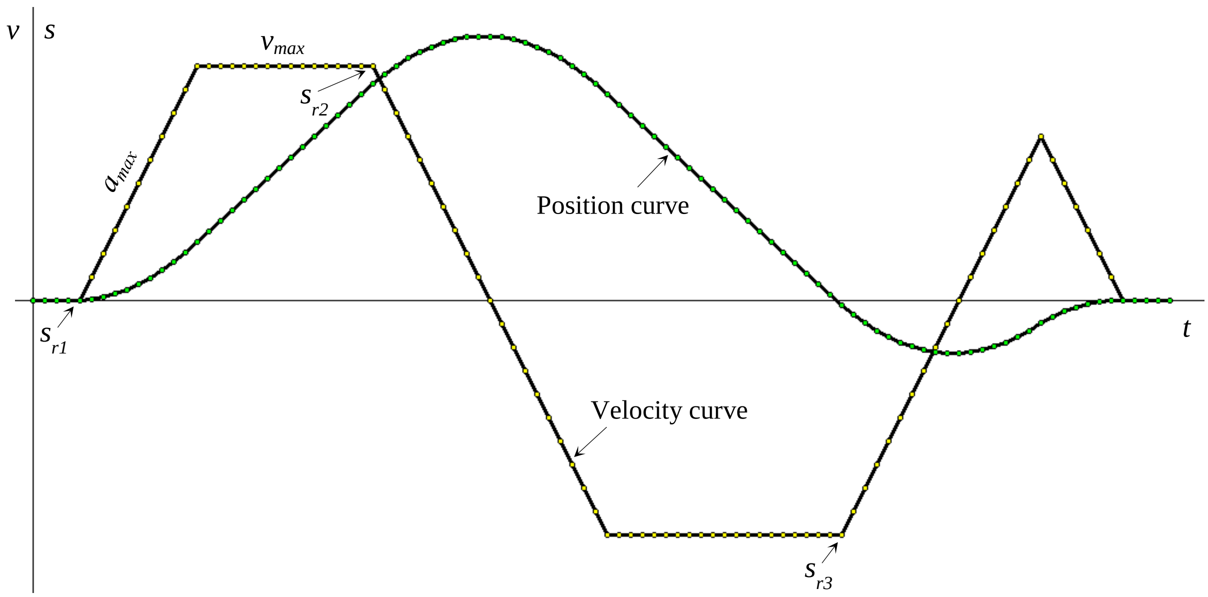

Fig. 6 Example of the trapezoidal control curve in the velocity and position axes

B. *S-curve*

S-curve guarantees the fluent change in the velocity axis. In consequence, the forces exerted on the servomotor in each step are changed gradually. S-curves are sometimes used in applications with high rotational velocities and sharp acceleration. It may cause lower vibrations of parts attended to the servomotor.

Fig. 7 presents the S-curve. Similarly to the trapezoid, both conditions are defined here: maximum velocity (v_{max}) and acceleration (a_{max}). In addition, there is a new condition: maximum change of acceleration, i.e. the third time derivative of the position function called jerk (j_{max}). As in the previous case the formula (1) has to be valid.

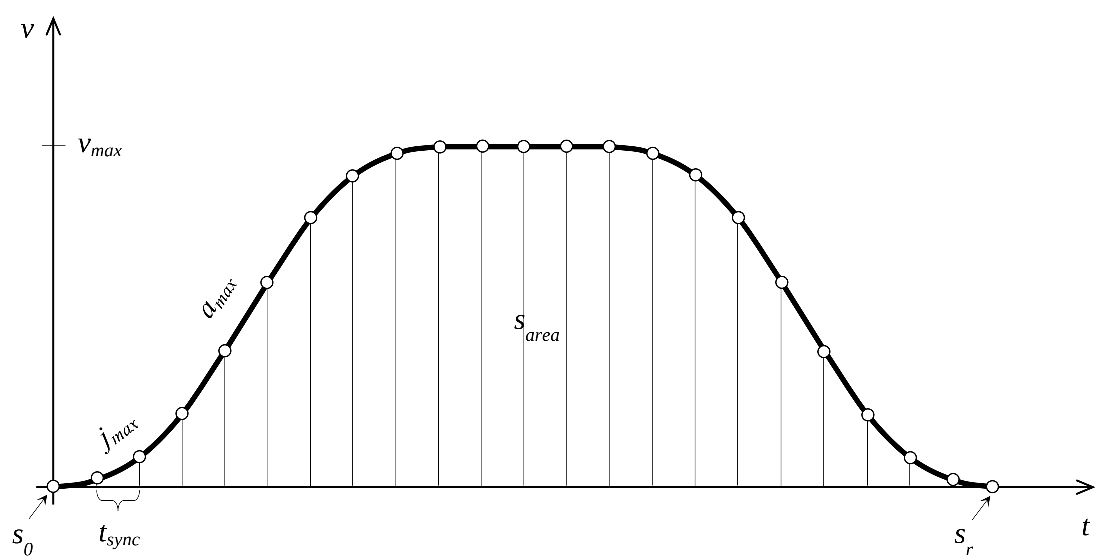

Fig. 7 S-curve for controlling servomotors

Fig. 8 presents the impact of the parameter j_{max} on the curve shape. Different shapes of control curves for various values of jerk are color coded. The lower the value of jerk is, the slower the curve reaches maximum velocity and the longer time duration is. In case the value of jerk is the same as the value of maximum acceleration ($j_{max} = a_{max} \cdot s^{-1}$), the curve becomes trapezoidal.

Fig. 9 shows the possible example of the control curve in the velocity and position axes (the scale of the position axis is lessened again). Position requirements from the control unit are called by symbols $s_{r1}, s_{r2}, s_{r3}, s_{r4}, s_{r5}$.

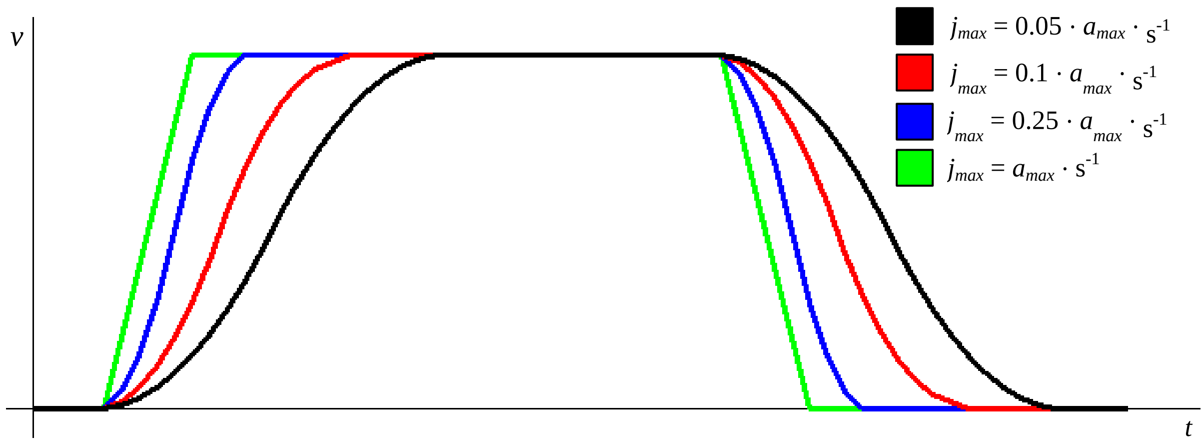

Fig. 8 Impact of jerk on the S-curve shape

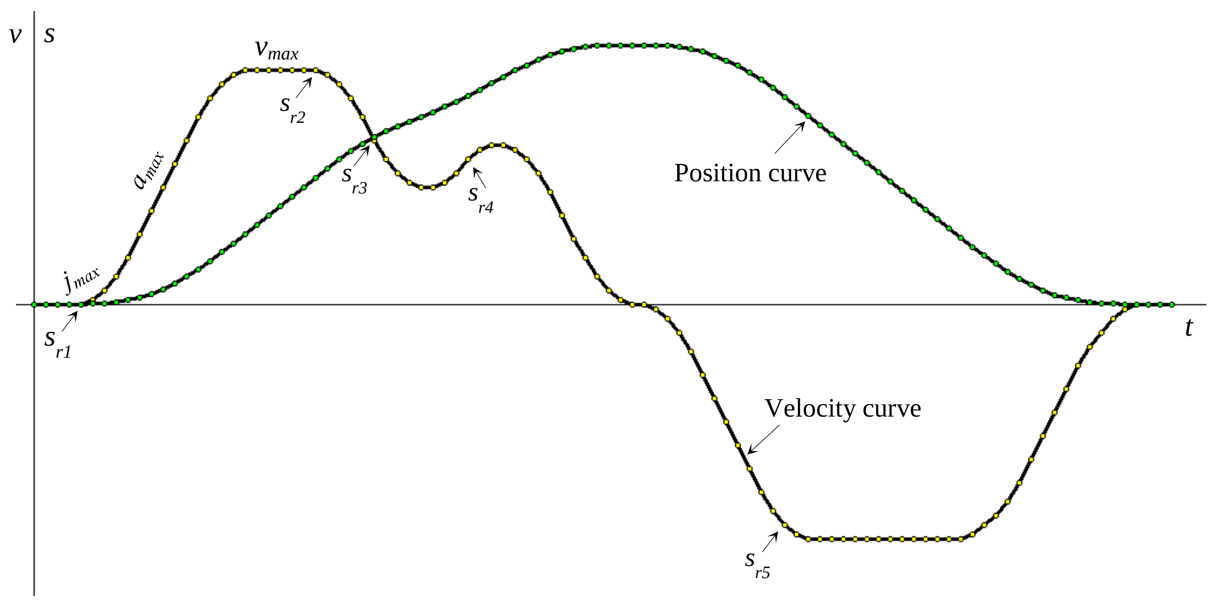

Fig. 9 Example of the control S-curve in the velocity and position axes

VI. Control Curves Evaluation

This section discusses both above mentioned types of control curves in terms of computational complexity, time duration and forces exerted on the servomotor.

A. Computational complexity

Control curves are computed on the microcontroller Stellaris with clock frequency 50 MHz. Generally, the algorithm for computing the trapezoid is relatively computational undemanding. Computing S-curves is more complex; however, we have developed the algorithm using the pre-computed tables saved in the microcontroller operation memory. This algorithm is also relatively very computational undemanding.

Table 1 presents the minimum, maximum and average runtime for computing both types of control curves in every step. Values were measured during the common operation in one of our applications (specifically in the Lafeta application – see chapter VII). The table shows that the S-curve average runtime is only about 45 % higher. The value of jerk did not have an impact on the results.

Control curve	Maximum runtime		Minimum runtime		Average runtime	
	Cycles	μs	*Cycles*	μs	*Cycles*	μs
Trapezoid	227	4.54	445	8.90	267	5.34
S-curve	254	5.08	567	11.34	388	7.76

Table 1 Computational complexity measurement

B. Time duration

It is clear that from the time duration point of view S-curves need more time to finish the task depending on the value of parameter j_{max}. Table 2 shows several examples of time duration (number of cycles) for three different tasks. Values of parameters of these tasks are in Table 3.

Tasks	Maximum jerk (j_{max}) $[\mathbf{n} \cdot \mathbf{t_{sync}}^{-3}]$				
	1	*2*	*3*	*4*	*5*
Task 1	295	-	291	-	-
Task 2	70	65	62	61	-
Task 3	40	32	25	23	22

Table 2 Time duration for three different tasks

Tasks	Parameters		
	$a_{max}\,[n \cdot t_{sync}{}^{-2}]$	$v_{max}\,[n \cdot t_{sync}{}^{-1}]$	$s_r\,[n]$
Task 1	5	200	50,000
Task 2	10	100	5,000
Task 3	20	500	2,000

Table 3 Values of parameters for tasks in Table 2

C. Force exertion

As already mentioned, the advantage of the S-curve consists in lower changes of forces exerted on the servomotor in order to reach the succeeding position according to the control curve in the next step. It causes the smaller vibrations of parts firmly fixed to the servomotor.

Fig. 10 presents the force exertion for the case of the trapezoid. There are apparent leaps of forces here (see the green color).

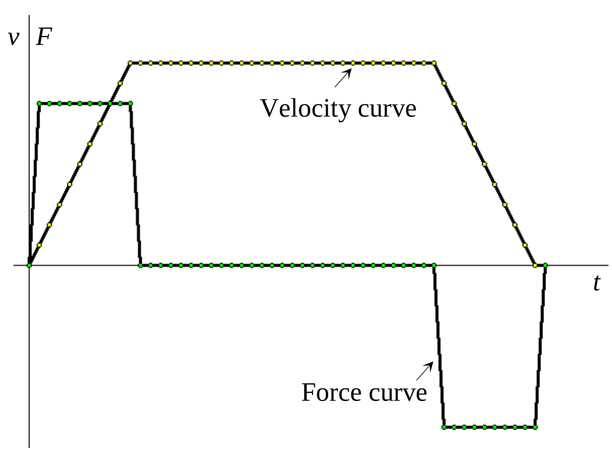

Fig. 10 Force exertion for the case of the trapezoid

Analogically, Fig. 11 shows the force exertion for the S-curve. There we can see the fluent changes of the force exerted.

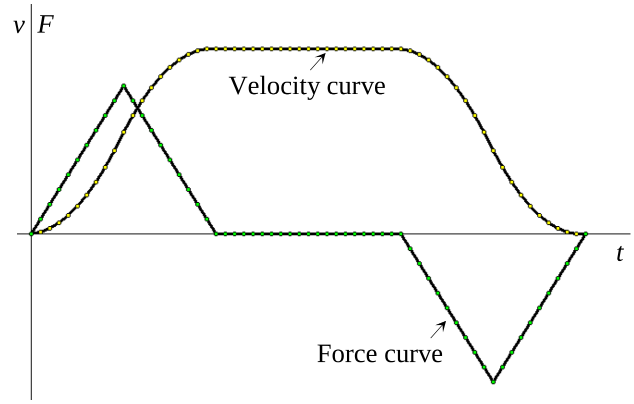

Fig. 11 Force exertion for the case of the S-curve

VII. POSSIBILITIES OF SYSTEM APPLICATIONS

The control system with the above mentioned control curves is being used at the University of Defence for controlling servomotors in real time in several applications as follows:

- **Lafeta**: a mobile wireless remote-controlled stabilized carriage for personal weapons. The carriage contains two servomotors for motion of a gun in two axes.
- **UGV vehicle TCX-G1**: an unmanned ground experimental vehicle for reconnaissance purposes with the possibility of target surveillance and destruction [1]. The vehicle contains a robotic platform with communication, sensor, control and weapon systems.
- **UGV vehicle for radiation reconnaissance**: an unmanned ground vehicle for radiation and chemical measurement. The vehicle carries several radiation and chemical sensors.

The experimental autonomous vehicle TCX-G1 has been developed at the University of Defence since 2007. The vehicle has been designed especially for reconnaissance purposes with the possibility of automatic searching for targets and their destruction [1], [2]. The vehicle is shown in Fig. 12.

The microcontroller Stellaris operates the five servomotors (connected to the two servo devices) in the real time (see Fig. 12). The used servomotors are as follows:

- Rotation of front wheels.
- Rotation of the sensor and weapon platform.
- Horizontal rotation of the camera system.
- Vertical tilting the camera system.
- Vertical tilting the weapon system.

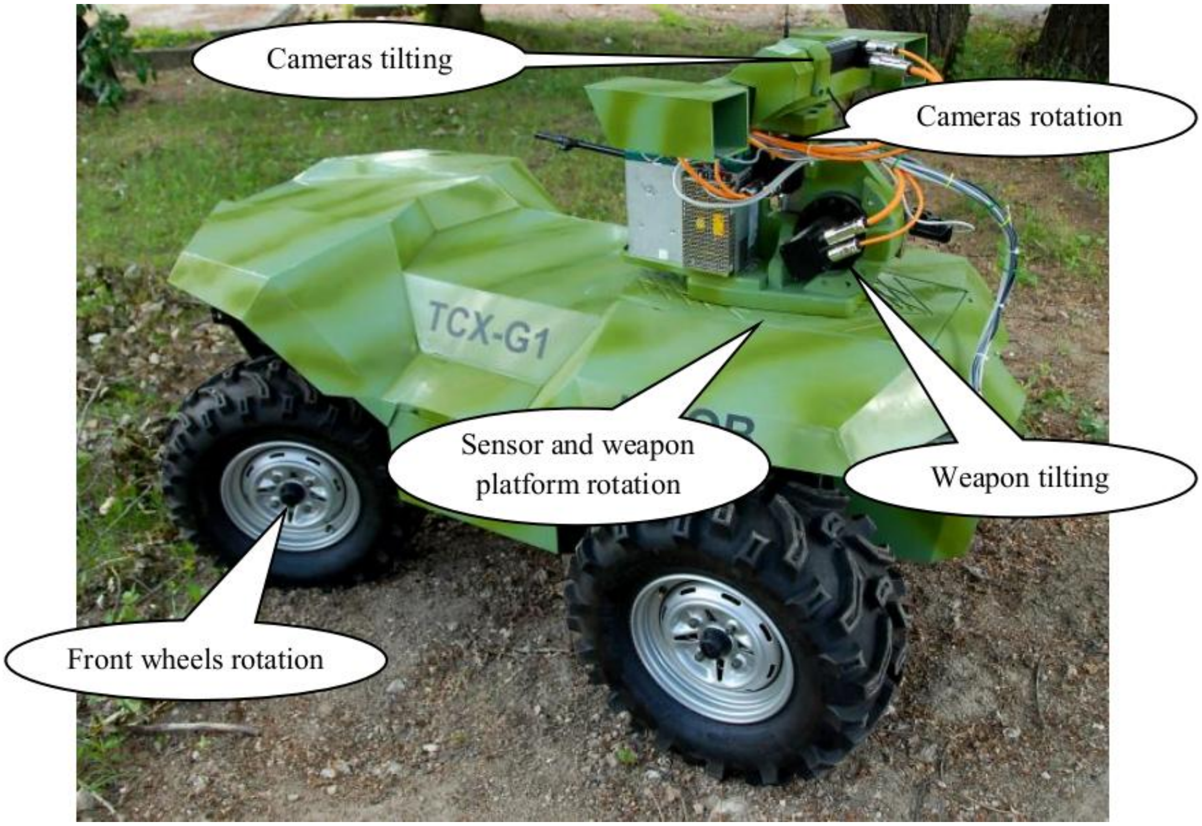

Fig. 12 Experimental autonomous ground vehicle TCX-G1

The second unmanned ground vehicle is presented in Fig. 13. The vehicle is used for radiation and contamination reconnaissance. It is equipped with a hybrid engine and semi-autonomous control (it follows waypoints set by an operator in advance).

This vehicle contains two servomotors controlled by the presented principle – the first for motion of the vehicle, the second for the front wheels rotation (see Fig. 13).

VIII. Conclusion

The article designs the real-time system for controlling servomotors via the microcontroller Stellaris. This system replaces the very expensive TG Motion system from a producer and at the same time it provides the same functionality.

Areas of its utilization are really wide; there are a lot of applications for it. The system can be used for controlling motion parts of robots, manipulators, CNC machines, assembly machines, etc. Reliability and precision of it was successfully verified when implemented into our experimental ground vehicle being developed and tested at the University of Defence and also in other applications.

The article presents two types of control curves: trapezoid and S-curve. S-curves should ensure more fluent forces exerted on servomotors in the phase of their acceleration or deceleration. An advantage consists in lesser oscillations of rotary parts when the servomotor is stopped rapidly (e.g. the barrel of a gun while tilting).

Perspectives of our next research in this area can be seen in measurement of an impact of a control curve type on vibrations of parts firmly attached to the servomotor. We will deal with this issue in the Lafeta application where we will measure vibrations by a precise accelerometer attached to a gun barrel.

Acknowledgment

This work has been supported by University of Defence under PASVR project.

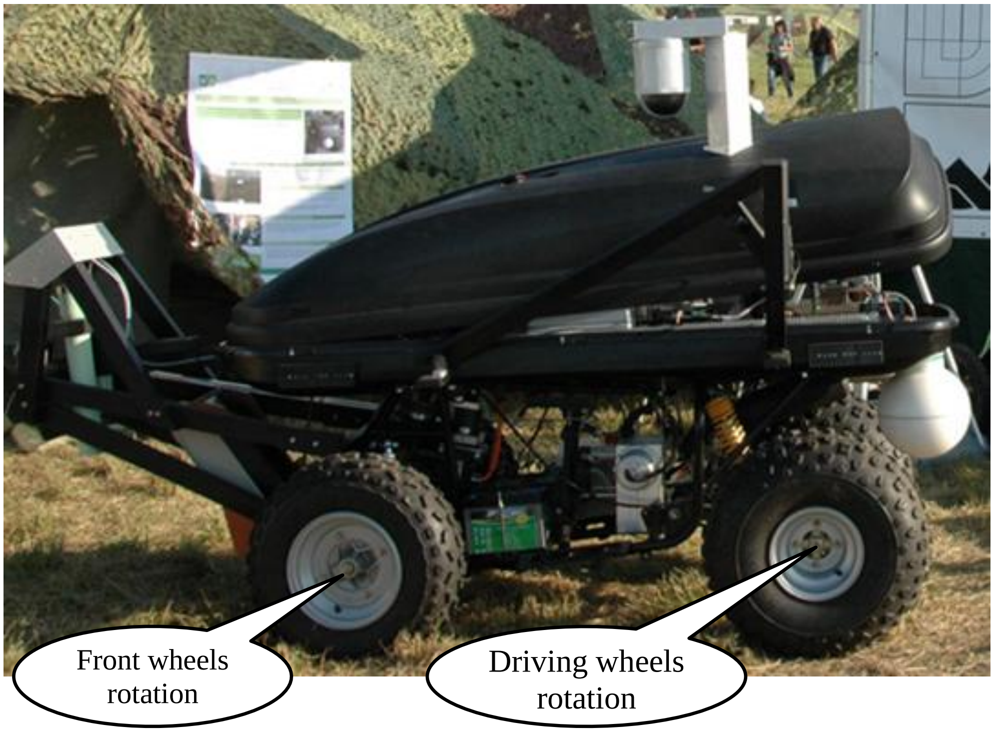

Fig. 13 Unmanned ground vehicle for radiation reconnaissance

REFERENCES

[1] P. Stodola, J. Mazal, "Optimal Location and Motion of Autonomous Unmanned Ground Vehicles", in *WSEAS Transactions on Signal Processing*, vol. 6, no. 2, 2010, pp. 68-77, ISSN 1790-5052.

[2] P. Stodola, "Extended Motion Model of Autonomous Ground Vehicle", in *International Journal of Mathematics and Computers in Simulation*, vol. 5, no. 1, 2011, pp. 28-35, ISSN 1998-0159.

[3] *TG Motion: Virtual PLC v 6.2*, Brno: TG Drives, 2005.

[4] *Sdrive: UNI Interface Board Manual*, ver. 7.51, Brno: TG Drives, 2009.

[5] *Stellaris LM3S8962 Microcontroller: Data Sheet*, Wild Basin: Texas Instruments, 2011.

[6] J. T. Hetzel, *Trapezoid: the trapezoidal distribution*, 2012, available at: http://cran.r-project.org/.

[7] D. Denison, *Log(ist)ic and simplistic curves*, Raymond Hickey (ed.), Motives for Language Change, Cambridge: Cambridge University Press, 2003, pp. 54-70, ISBN 978-0-521-79303-2.

[8] PMC, *Velocity profiles – trapezoidal versus S-curve*, Motion Control for Robotics & Machine Automation, 2000.

[9] A. T. Hasan, "Under-Actuated Robot Manipulator Positioning Control Using Artificial Neural Network Inversion Technique", in *Advances in Artificial Intelligence*, vol. 2012, 6 p, ISSN 1687-7470.

[10] I. Hassanzadeh, M. Darabi, S. Hassanzadeh,"Rapid Prototyping of a Manipulator Mechanism Using Hardware in the Loop (HIL) Simulators and Comparing the Results", in *WSEAS International Conference on Applications of Electrical Engineering*, Prague, 2006, pp. 191-196. ISBN 960-8457-42-4.

[11] A. A. Zaher, M. A. Zohdy, "Robust Motion Control of Biped Walking Robots", in *WSEAS Transactions on Systems and Control*, vol. 4, no. 12, 2009, pp. 613-624, ISSN 1991-8763.

[12] R. S. Lin, S. L. Chen, J. H. Liao, "Advanced Curve Machining Method for 5-Axis CNC Machine Tools", in *International MultiConference on Engineers and Computer Scientists*, Hong Kong, 2011, ISBN 978-988-19251-2-1.

[13] F. Terneus, F. Reyes, E. Lebano, "A family of hyperbolic and exponential-type controllers", in *International Journal of Mathematics and Computers in Simulation*, vol. 3, no. 4, 2009, pp. 159-169, ISSN 1998-0159.

[14] O. Hachour, "The proposed hybrid intelligent system for path planning of Intelligent Autonomous Systems", in *International Journal of Mathematics and Computers in Simulation*, vol. 3, no. 3, 2009, pp. 133-145, ISSN 1998-0159.

Petr Stodola was born in Brno, the Czech Republic, in 1979. He graduated in electrical engineering in 2003 at Military Academy in Brno. He finished his Ph.D. study in electrical engineering in 2006 at the University of Defence in Brno.

He has been working at the University of Defence in Brno in the Department of Military Management and Tactics since 2006 as a pedagogic and scientific worker. He has published more than 80 scientific works.

Dipl. Eng. Petr Stodola, Ph.D. deals mainly with automation processes, optimization algorithms and operational research.