APPLICATION OF THE MODERN TAYLOR SERIES METHOD TO A MULTI-TORSION CHAIN

Georg Fuchs[1], **Jan Kopřiva**[2], **Michal Kraus**[2], **Martin Kozek**[1]

[1]Vienna University of Technology, Institute of Mechanics and Mechatronics,
Division of Control and Process Automation,
1040 Vienna, Wiedner Hauptstrasse 8-10, Austria
[2]Brno University of Technology, Faculty of Information Technology,
Department of Intelligent Systems
612 66 Brno, Božetěchova 2, Czech Republic

georg.fuchs@tuwien.ac.at(Georg Fuchs)

Abstract

In this paper the adoption of a novel high accuracy numerical integration method is presented for a practical mechanical engineering application. It is based on the direct use of the Taylor series. The main idea behind it is a dynamic automatic order setting, i.e. using as many Taylor series terms for computing as needed to achieve the required accuracy. Previous results have already proved that this numerical solver is both very accurate and fast. In this paper the performance is validated for a real engineering assembly. The chosen experiment setup is a multi-torsional oscillator chain which reproduces typical dynamic behavior of industrial mechanical engineering problems. Its rotatory dynamics are described by linear differential equations. For the test series the system is operated in a closed-loop configuration. An analytic solution of the linear differential equations of the closed-loop system for the output variable is obtained with the mathematical software tool Maple and validated by comparison to measurements at the experiment. The performance of the Modern Taylor Series Method is demonstrated by comparing its results to simulation results from conventional fixed-step numerical integration methods from the software tool Matlab/Simulink. Furthermore, the improvement in numerical accuracy as well as stability is illustrated.

Keywords: Simulation, Taylor series, Numerical integration, Matlab/Simulink.

Presenting Author's Biography

Georg Fuchs received the M.S. degree in Mechanical Engineering in 2008 from the Vienna University of Technology. He is presently research assistant at the Institute of Mechanics and Mechatronics at the Vienna University of Technology, pursuing his Ph.D. degree in automatic control engineering. His current research interests are internal combustion engine modeling, simulation and control, numerical linearization and integration methods, and real-time hardware applications.

1 Introduction

In many modern applications in industry simulation has become a powerful tool for engineers in order to predict real system behavior or validate closed loop controller performance without conducting cost-intensive test cycles on appropriate test facilities. Often the specific problem setup requires the application of real-time algorithms running on external hardware units, as for example hardware-in-the-loop systems on automotive test stands. Simulation speed as well as stability of the performing algorithms are critical in such applications.

Today's most common mathematical simulation software packages (e.g. Matlab/Simulink) provide various types of numerical integration methods [1]. These methods differ primarily in the way the solution at the next time step is calculated, knowing the time derivative at the current time step. Variable-step solvers are able to adapt the interval step size dynamically during simulation, depending on the current rate of change of the solution. Fixed-step solvers which are used in real-time systems have a defined fixed step size because they need to calculate the simulation output deterministically for each time step. Increasing the step size, fixed-step algorithms typically become unstable at a certain step size limit. More information about numerical methods and stability can be found in [2]. Finding a numerical integration method which is accurate, fast and also robust regarding stability, would therefore increase the quality of such real-time algorithms drastically.

A very promising approach for such problems is the Modern Taylor Series Method (MTSM) [3] - a special parallel system which has been developed at the Brno University of Technology. This parallel system can be used in a special hardware unit for the acceleration of numerical integration. The main component of the parallel system is a numerical integrator carrying out numerical integration based on the Taylor series. A description of this system can be found in [4].

The Institute of Mechanics and Mechatronics at the Vienna University of Technology deals with modern control engineering methods for different types of industrial applications. In order to compare the performance of the MTSM to other fixed-step solvers a laboratory experiment model of a multi-torsional oscillator has been chosen. This system represents a typical example for such applications as it serves as analogous model for drive train components of internal combustion engines. It can be modeled by linear ordinary differential equations using theoretical modeling techniques and can be simulated on common numerical simulation tools, e.g. Matlab/Simulink. Measurement results for all relevant variables can be gathered without high technical effort. Furthermore, the closed-loop control of this system demonstrates a characteristic use case within automatic control applications.

The paper is structured as follows: Section 2 contains a detailed description of the MTSM. Section 3 is devoted to the experimental setup which is used for the validation of the analytical solution obtained by Maple and contains the derivation of the differential equations from a mathematical modeling. Section 4 illustrates the application of the MTSM to the differential equations as well as the controller structure and Section 5 provides tests and comparisons of the presented numerical integration methods.

2 Modern Taylor Series Method

The Taylor series method is one of the earliest analytic-numeric algorithms for the approximate solution of initial value problems for ordinary differential equations. Even though this method is not much preferred in literature, experimental calculations have shown and theoretical analyses have verified that the accuracy and stability of the Taylor series method exceeds the currently used algorithms for numerically solving differential equations.

The numerical solution of an ordinary differential equation (1)

$$\dot{y} = f(t, y), \qquad y(t_0) = y_0 \tag{1}$$

is written as the sequence (2)

$$[y(t_0) = y_0], \quad [y(t_1) = y_1], \quad \cdots \quad [y(t_n) = y_n]. \tag{2}$$

The best-known and most accurate method of calculating a new value of a numerical solution of a differential equation is to construct the Taylor series in the form (3).

$$\begin{aligned} y_{n+1} = y_n \quad &+ \quad h \cdot f(t_n, y_n) + \frac{h^2}{2!} \cdot f^{[1]}(t_n, y_n) + \cdots \\ &+ \quad \frac{h^p}{p!} \cdot f^{[p-1]}(t_n, y_n) \end{aligned} \tag{3}$$

where h is the integration step.

The main idea behind the MTSM is an automatic integration method order setting, i.e. using as many Taylor series terms for computing as needed to achieve the required accuracy. The MTSM used in the computations increases the method order automatically, i.e. the values of the terms (4) are computed for increasing integer values of p until adding the next term does not improve the accuracy of the solution (last three terms of Taylor series are equal to zero).

$$\frac{h^p}{p!} \cdot f^{[p-1]}(t_n, y_n) \tag{4}$$

The main problem connected with using the Taylor series (in the form of (3)) is the need to generate higher derivatives $f^{[1]}, f^{[2]}, \cdots$. If it is possible, however, to obtain the terms with higher derivatives, the accuracy of calculations by the Taylor series method is extreme (it is in fact only limited by the type of the arithmetic unit used). A drawback of this method is that $f(t, y)$ has to belong to a special class. Fortunately, this class is large enough to contain the functions that appear in many applications. This is typical, in particular, of the solution of technical initial problems.

2.1 Technical Initial Problems

Technical initial problems are defined as initial problems where the right-hand side functions of the system are those occurring in technical practice, that is functions generated by adding, multiplying and superposing elementary functions. Such systems can be expanded into systems with polynomials on the right-hand sides of the equations. In such a case the Taylor series terms can be easily calculated.

To demonstrate this, Eq. (5) is analyzed.

$$\dot{y} = a \cdot y \cdot \cos t \qquad y(0) = y_0 \tag{5}$$

A simple computation scheme based on equation (1) follows

$$f(t, y) = a \cdot y \cdot \cos t \tag{6}$$

Let $v = \cos t$ then

$$\begin{aligned}
f(t, y) &= a \cdot y \cdot v \\
f^{[1]}(t, y) &= a(f(t, y) \cdot v + y \cdot \dot{v}) \\
f^{[2]}(t, y) &= a(f^{[1]}(t, y) \cdot v + 2 \cdot f(t, y) \cdot \dot{v} + y \cdot \ddot{v}) \\
&\ \ \vdots \\
f^{[p-1]}(t, y) &= a \cdot (\sum_{i=0}^{p-2} f^{[p-2-i]}(t, y) \cdot v^{[i]} \binom{p-1}{i} + \\
&\quad + y \cdot v^{[p-1]}) \quad (p \geq 2)
\end{aligned} \tag{7}$$

where

$$\begin{aligned}
\dot{v} &= -\sin t \\
u &= \sin t \\
v^{[p]} &= -u^{[p-2]} \quad (p \geq 2) \\
u^{[p-1]} &= v^{[p-2]} \quad (p \geq 2)
\end{aligned} \tag{8}$$

Similar constructions can be created for all elementary functions, such as $\exp, \sin, \cos, \tan, \coth, \ln, \sinh, \ldots$. The right-hand side of equations that belong to the technical initial problems can be decomposed in a sequence of simple operations: addition, subtraction, multiplication, division and combination of them. These rules can be applied recursively so that recursive formulas for the derivatives of a function described by combinations of these elementary functions can be obtained.

More detailed information about the MTSM can be found in [3]. The MTSM has been implemented in TKSL software [5], which is used for simulation results in this paper.

Some articles that are focused on the MTSM were published last year. Paper [6] describes a new modern numerical method based on the Taylor series method and shows how to evaluate the high accuracy and speed of the corresponding computations. Article [7] deals with the simulation system TKSL that has been created for the numerical solution of first order differential equations using the MTSM.

There are several papers that focus on computer implementations of the Taylor series method in different context - a variable order and **variable step** formulation of the Taylor method for the numerical solution of ODEs (see, for instance, [8]). Another more detailed description of a variable step size version of the Taylor series can be seen in [9]. This paper describes a software implementation of the method as well. The stability domain for several Taylor methods is presented in [10].

3 Application

3.1 Experimental setup

Fig. 1 shows the assembly of the experiment which is used for the validation of the presented integration method. It consists of two torsional oscillator modules (blue casings) which are driven by a separate servo drive unit (black casing).

The system is operated from a Windows PC via a Quanser UPM 1503 power module and Q4 control hardware using Quanser's WinCon control software package for Simulink. For further information on Quanser components and software tools please see [11].

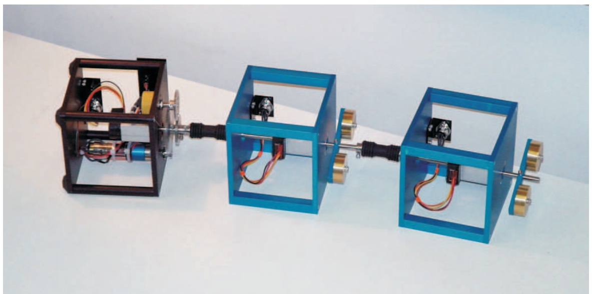

Fig. 1 Multi-torsion chain

3.2 Modeling

The system's dynamical behavior is modeled using a mathematical-physical approach consisting of the electrical and mechanical subsystems.

Servo drive
The servo drive is illustrated in Fig. 2. The goal is to model the torque T, subject to the terminal voltage U and the speed $\dot{\phi}_1$.

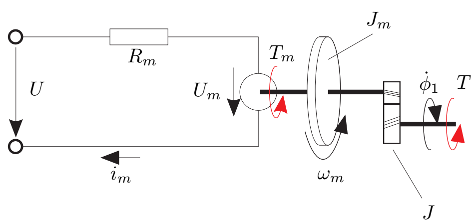

Fig. 2 Servo drive of the multi-torsion chain

Using Kirchhoff's laws and the law of conservation of angular momentum, one obtains the relation

$$J_1\ddot{\phi}_1 = k_g K_m \frac{U - K_m k_g \dot{\phi}_1}{R_m} - T \qquad (9)$$

between the speed $\dot{\phi}_1$, the terminal voltage U and the drive torque T, with k_g being the gear transmission ratio, K_m the gyrator constant of the motor and $J_1 = J_m k_g^2 + J$ the combined moment of inertia of the motor and the gear transmission.

Multi-torsion chain

The two torsional modules each consist of an oscillator and a rubber cuff which serves as connection to the next module, respectively to the servo drive. Fig. 3 shows the two torsional modules schematically. The mass moments of inertia of the two oscillators are denominated J_2 and J_3, the torsional stiffnesses of the rubber cuffs k_{s1} and k_{s2}, and the torsional damping resistances λ_1 and λ_2. Using the law of conservation of angular momentum for the two torsional modules as well as relation (9), one obtains

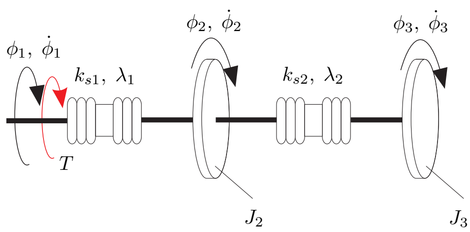

Fig. 3 Torsional modules

$$\begin{aligned}
\ddot{\phi}_1 &= \frac{1}{J_1}[k_{s1}(\phi_2 - \phi_1) + \lambda_1(\dot{\phi}_2 - \dot{\phi}_1) + \\
&\quad + U\frac{K_m k_g}{R_m} - \dot{\phi}_1 \frac{K_m^2 k_g^2}{R_m}] \qquad (10)\\
\ddot{\phi}_2 &= \frac{k_{s1}}{J_2}[\phi_1 - \phi_2] + \frac{k_{s2}}{J_2}[\phi_3 - \phi_2] + \\
&\quad + \frac{\lambda_1}{J_2}[\dot{\phi}_1 - \dot{\phi}_2] + \frac{\lambda_2}{J_2}[\dot{\phi}_3 - \dot{\phi}_2] \qquad (11)\\
\ddot{\phi}_3 &= \frac{k_{s2}}{J_3}[\phi_2 - \phi_3] + \frac{\lambda_2}{J_3}[\dot{\phi}_2 - \dot{\phi}_3]. \qquad (12)
\end{aligned}$$

4 Modern Taylor Series Method Application

If the MTSM is used for the solution of differential equations it is necessary to transform the system of higher order differential equations to an equivalent system of first order differential equations. This methodology can be seen in literature [12]. The transformation (reduction of order) of Eqs. (10), (11), (12) yields:

$$\begin{aligned}
\dot{\phi}_1 &= a \qquad (13)\\
\dot{\phi}_2 &= b \qquad (14)\\
\dot{\phi}_3 &= c \qquad (15)\\
\dot{a} &= \frac{1}{J_1}[k_{s1}(\phi_2 - \phi_1) + \lambda_1(b - a) - \\
&\quad - \frac{K_m^2 k_g^2}{R_m} a] - \frac{K_m k_g}{J_1 R_m}[k_1\phi_1 + k_2\phi_2 + \\
&\quad + k_3\phi_3 + k_4 a + k_5 b + \\
&\quad + k_6 c - K_w w] \qquad (16)\\
\dot{b} &= \frac{k_{s1}}{J_2}[\phi_1 - \phi_2] + \frac{k_{s2}}{J_2}[\phi_3 - \phi_2] + \\
&\quad \frac{\lambda_1}{J_2}[a - b] + \frac{\lambda_2}{J_2}[c - b]. \qquad (17)\\
\dot{c} &= \frac{k_{s2}}{J_3}[\phi_2 - \phi_3] + \frac{\lambda_2}{J_3}[b - c]. \qquad (18)
\end{aligned}$$

Note that the k_i and K_w in Eq. (16) are the controller parameters of the closed-loop setup given in Section 5.1. Therefore, Eqs. (13) - (18) already characterize the closed-loop dynamics.

5 Setup for Comparison

The goal of this paper is to compare results from the MTSM and conventional numerical integration methods from Matlab/Simulink. The variable which is used for comparison is the angle ϕ_3 of the torsional chain (see Fig. 3). For this purpose an analytical solution for a rectangular input signal U (see Section 5.2) is obtained by the mathematical software tool Maple. This solution is validated by comparison to measurements of the real system described in Section 3. The validation is conducted in Section 5.2. The analytical solution from Maple is used for the comparison of the simulation results from the MTSM and Matlab solvers. The computation error is calculated as difference between the analytical solution from Maple and the numerical results from the MTSM and Matlab solvers, respectively.

5.1 Controller

The system is controlled in closed-loop by a state vector feedback controller whose parameters are computed by pole placement. The general structure of such a closed-loop control system with the reference input w and the output y is depicted in Fig. 5.1.

For more information about control design see [13],[14].

5.2 Real system measurements

Fig. 5 shows the measurement of the output of the real system (ϕ_3 in Eq. (12)) and the analytical result from Maple (denoted y^*) to a rectangular reference input signal w with the amplitude of $5\,deg$ and a frequency of $0.5\,Hz$ (blue line). This reference input signal was chosen because it contains high-gradient level changes and is well-suited for the comparison of performance of numerical computation tools. The sampling time for the

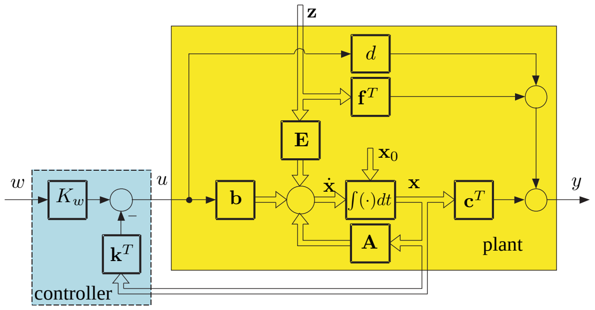

Fig. 4 State vector feedback control structure

measurement on the real system was $0.001\ s$ and for the analytical solution in Maple it was $0.005\ s$. In Fig. 5 the magenta line is the measured signal from the real system (y_m) and the green one is the analytical output signal from Maple (y^*). The reference input signal w is drawn in black.

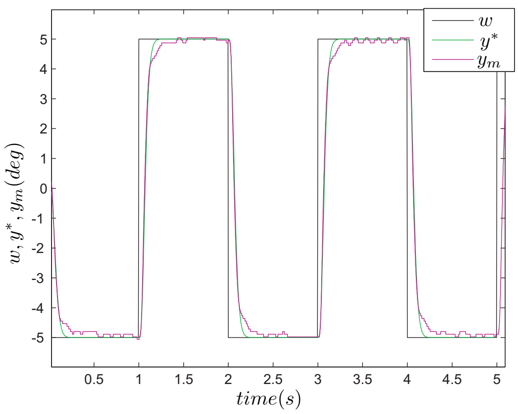

Fig. 5 Results from measurement and analytical solution

From these results it can be seen that the output calculated by Maple approximates the measured output from the real system very well. It is therefore used for the comparison of the numerical integration methods.

5.3 Simulation Results

The results of the fixed-step solvers from Matlab/Simulink (ODE1 - Euler, ODE3 - Bogacki-Shampine, ODE4 - Runge-Kutta, see [1]) and the MTSM (denoted by ODE6) are compared to each other in this Section. The analytical solution from Maple is used for this comparison. The simulation time step is chosen as $0.005s$.

A linear Simulink model that represents the real system behavior is shown in Fig. 6. The plant model consists of the state space system with the input u (voltage) and the output y (angle ϕ_3). The system is controlled by a state vector feedback controller with the feedback gain vector $\mathbf{k}^T$ and the pre-filter K_w. The reference input w for this test series is a rectangular signal with frequency $0.5\ Hz$ and amplitude $5\ deg$.

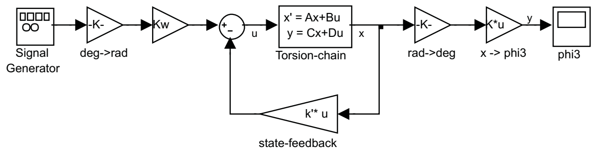

Fig. 6 Simulink model

The analytical solution y^* from Maple and the simulation results y_{sim} from Matlab/Simulink (ODE1, ODE3, ODE4) and the MTSM (ODE6) respectively are shown in Fig. 7.

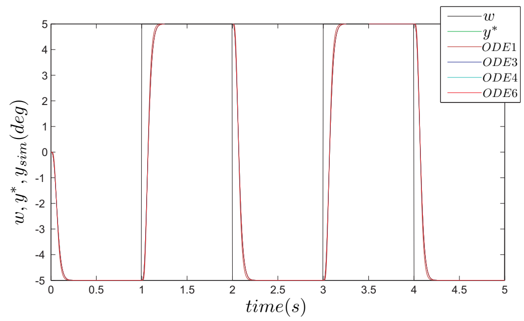

Fig. 7 Simulation results

5.4 Comparison of Results

The computation error has been calculated as the difference

$$e = y^* - y_{sim} \tag{19}$$

between the analytical solution from Maple y^* and the numerical results from Matlab/Simulink fixed-step solvers ODE1, ODE3 and ODE4, and the MTSM, respectively. These errors are shown in Fig. 8.

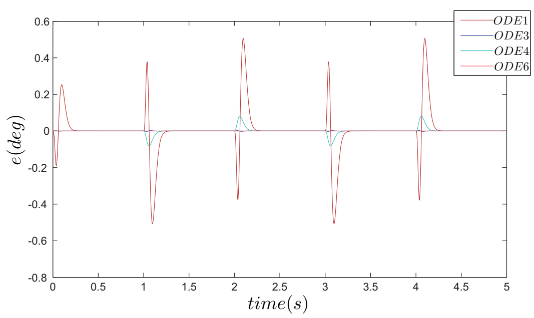

Fig. 8 Error between analytical and numerical solutions

ODE1 produces the relatively largest error, as can be seen in the plot. ODE4 delivers more accuracy, but the error is still comparatively high. Obviously, ODE6 and ODE3 produce the best results. Therefore, Fig. 9 separately shows the absolute errors for ODE3 and ODE6 compared to each other. Especially in the regions of input signal steps the ODE6 solution yields considerably better results than ODE3.

Additionally, in Fig. 10 the difference between the analytical solution and ODE6 is shown.

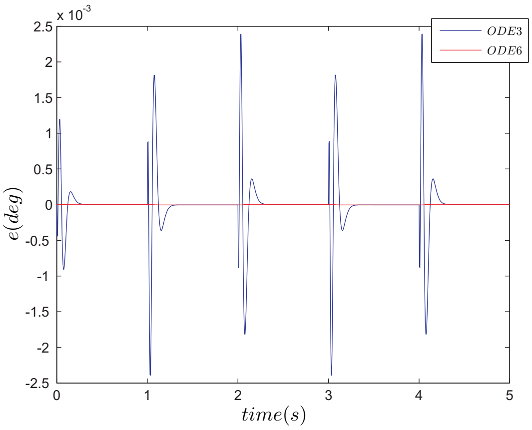

Fig. 9 Error between analytical and numerical solutions for ODE3 and ODE6 solvers

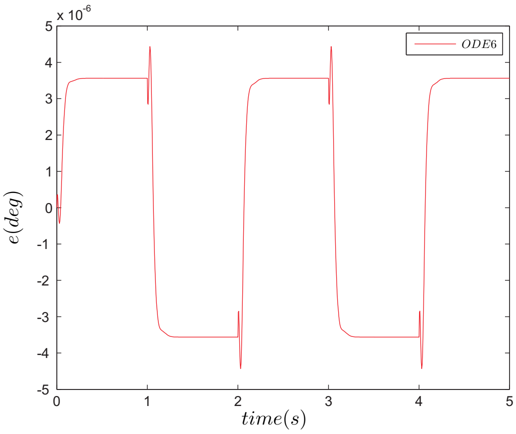

Fig. 10 Error between analytical and numerical solution for ODE6 solver

Tab. 1 and Tab. 2 show the mean absolute error

$$\bar{e}_{abs} = \frac{1}{n} \sum_{i=0}^{n} |y_i^* - y_{sim,i}| \tag{20}$$

and the mean relative error

$$\bar{e}_{rel} = \frac{1}{n} \sum_{i=0}^{n} \frac{|y_i^* - y_{sim,i}|}{y_i^*} \tag{21}$$

between the analytical solution from Maple and each solver ODE1-ODE6 for different frequencies ($0.25\,Hz, 0.5\,Hz, 0.75\,Hz$ and $1.0\,Hz$) of the rectangular input signal w, with n the number of time steps. The results from the table confirm the results that are shown in Fig. 8 and 9.

5.5 Stability comparison

The quality of results of simulations using one of the mentioned numerical fixed-step integration methods from Matlab/Simulink is depending on the step size. If

Tab. 1 Absolute error of numerical methods for different frequencies of input signal w

f	Solver error - absolute			
[Hz]	ODE1	ODE3	ODE4	ODE6
0.25	0.0239	8.6097e-5	0.0033	3.4532e-6
0.5	0.0431	0.0067	1.5218e-4	3.3982e-6
0.75	0.0713	0.0056	0.0117	3.3137e-6
1	0.0910	3.1738e-4	0.0150	3.2607e-6

Tab. 2 Relative error in percent of numerical methods for different frequencies of input signal w

f	Solver error - relative			
[Hz]	ODE1	ODE3	ODE4	ODE6
0.25	1.91	0.12	0.41	1.5787e-4
0.5	3.12	0.12	0.81	1.5916e-4
0.75	4.12	0.63	1.19	1.6093e-4
1	6.15	0.14	1.8	1.6241e-4

the step size is increased the results will show unstable behavior once a critical step size has been reached. For the ODE3 solver in the presented application this step size was found to lie between $h = 0.020\,s$ and $h = 0.025\,s$. Fig. 11 shows the results of the ODE3 and ODE6 solvers for a step size of $h = 0.023\,s$. Significant indication of unstable behavior for the ODE3 solver can clearly be determined by the oscillations around the corners of the input signal. For further information on numerical stability please see [15].

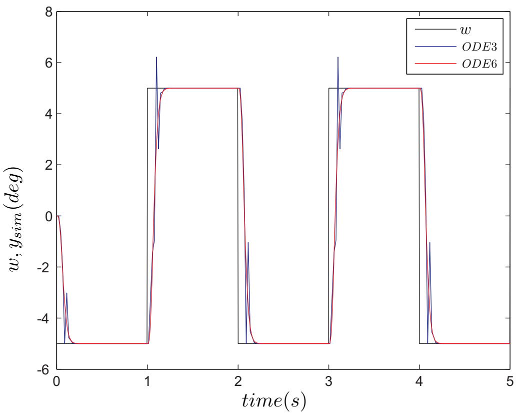

Fig. 11 Stability test of numerical solution

Using the MTSM, no significant qualitative decline in the performance can be determined for increasing step sizes. It is possible to achieve high accuracy and stability, even if the step size increases, because the MTSM changes the order of the method (ORD) automatically - for increasing step sizes the method order increases as well. Another advantage of this method is that for nearly stationary input signals where the gradients are small, it does not include higher orders of Taylor series terms. For high frequency input signals with large gradients, the MTSM calculates all higher order terms it needs for sufficient accuracy. This behavior is shown in a screenshot of TKSL in Fig. 12. The violet line represents results of simulation using the MTSM and the blue line shows the corresponding order of the method (ORD). As can be seen, in areas around the corners of the input signal the MTSM uses higher orders for achieving a stable and accurate solution.

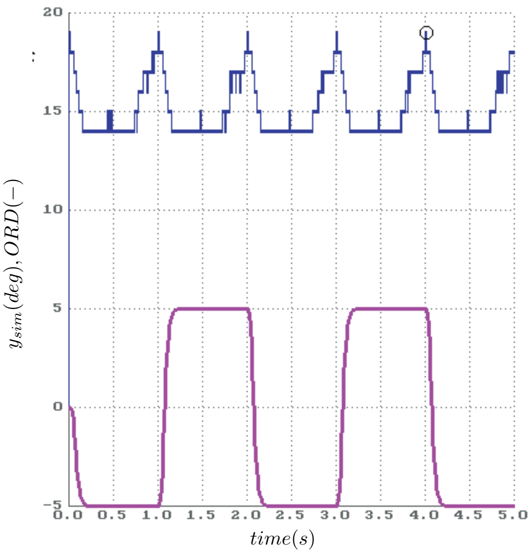

Fig. 12 The value of ORD changes during the computation (blue line)

6 Conclusion

In this paper the application of a novel numerical integration method to a real experimental setup was presented. The main advantage of this method is the dynamical adaption of the number of Taylor series terms during simulation. Before evaluating the performance of the method, measurement results were obtained on the real experiment. An analytical solution for a specific input was generated using the mathemetical software package Maple and validated with the measurements. The MTSM was then compared to the analytical solution and existing fixed-step numerical integration methods from Matlab/Simulink. The results showed that the performance of the new method achieved the most accurate results and also proved a significantly higher robustness concerning increased step sizes compared to conventional numerical integration methods.

When regarding the online computational effort compared to other fixed-step numerical solvers, the MTSM has essential advantages: For nearly stationary input signals where the gradients are small, it does not include higher orders of Taylor series terms and is therefore computationally very efficient. For high frequency inupt signals with large gradients, it calculates all higher order terms it needs for sufficient accuracy. This fact enables it to be very fast and also accurate.

7 References

[1] Web sites, Matlab Simulink solvers, March 29 2010. `http://www.mathworks.com/access/helpdesk/help/toolbox/simulink/ug/f11-69449.html`.

[2] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, 3rd edition, September 2007.

[3] J. Kunovský. *Modern Taylor Series Method*. FEI-VUT Brno, 1994. Habilitation work.

[4] J. Kunovský, , V. Kaluža, M. Kraus, and V. Šátek. Parallel computations based on analogue principles. In *Proceedings of Eleventh International Conference on Computer Modelling and Simulation*, pages 111–116. IEEE Computer Society, 2009.

[5] Web sites, High performance computing March 29 2010. `http://www.itsolutions.cz/TKSL`.

[6] J. Kunovský, V. Šátek, and M. Kraus. New trends in Taylor series based computations. In *Numerical Analysis and Applied Mathematics*, pages 282–285. American Institute of Physics, 2009.

[7] J. Kunovský, V. Šátek, M. Kraus, and J. Kopřiva. Automatic method order settings. In *Proceedings of Eleventh International Conference on Computer Modelling and Simulation*, pages 117–122. IEEE Computer Society, 2009.

[8] R. Barrio A, F. Blesa B, M. Lara C, and Grupo De Mecnica Espacial. Monografas de la Real Academia de Ciencias de Zaragoza. 22: 6774, (2003). High-precision numerical solution of ODE with high-order Taylor methods in parallel.

[9] A. Jorba and M. Zou. A software package for the numerical integration of ODEs by means of high-order taylor methods. *Experimental Mathematics*, 14:99–117, 2005.

[10] R. Barrio. Performance of the Taylor series method for ODEs/DAEs. *Applied Mathematics and Computation*, 163:525–545, 2005.

[11] Web sites, Quanser control systems, March 29 2010. `http://www.quanser.com/choice.asp`.

[12] E. Kreyszig. *Advanced Engineering Mathematics*. John Wiley and Sons, Inc., 9th edition, 2006.

[13] R. Dorf and R. Bishop. *Modern Control Systems*. Prentice Hall, 11th edition, 2007.

[14] S. Jakubek. *Lecture notes: Zustandsregelung von Mehrgroessensystemen*. Vienna University of Technology, 2010.

[15] A. Quarteroni. *Numerical Mathematics*. Springer-Verlag, New York, 1st edition, 2000.