Andrzej TUTAJ, Jacek AUGUSTYN
AGH UNIVERSITY OF SCIENCE AND TECHNOLOGY,
al. A. Mickiewicza 30, 30-059 Krakow

Low cost environment for rapid prototyping of control algorithms for mechatronic plants based on Microsoft Office Excel application

Abstract

The paper presents a low-cost rapid prototyping environment for control algorithm development based on Microsoft Excel program and a measurement and control board built on an ARM Cortex-M core microcontroller. The solution is intended mainly for mechatronic devices. It provides tools for computer simulations and soft real-time closed loop control experiments. An application to a fast and unstable mechatronic plant of an inverted pendulum on a cart is presented to show system capabilities.

Keywords: rapid prototyping, mechatronic system, Microsoft Office Excel, VBA, soft real time.

1. Introduction

The final implementation of a control algorithm for a mechatronic plant is usually preceded by a development stage where a rapid prototyping environment (RPE) is employed. The environment consists of software and hardware components interfacing both with a designer and with a real-world plant. It provides means for numerical modelling of the controller and the plant as well as tools for controller tuning. The environment allows to carry numerical simulations of the closed control loop as well as to conduct real-time experiments involving the real-world plant. Time series of control system signals can be logged for further processing and visualisation. Two popular examples of RPEs are system based on MATLAB and LabVIEW software, respectively. The hardware part of RPE interfacing with a plant is usually an input-output (IO) PCI extension card installed in a PC computer or an external measurement and control board connected to the computer via USB bus or Ethernet network. Controller and plant numerical models can be often coded using graphical languages implementing dataflow programming paradigm. During the experimental stage the hard real-time regime is usually employed. Professional commercial RPE solutions are flexible and universal but their relatively high prices prohibit their usage in projects with limited budgets.

The paper proposes an alternative low-cost RPE solution based on Microsoft Excel program running under MS Windows operating system on a PC computer and communicating via the USB bus with a measurement and control board built on a microcontroller with an ARM Cortex-M core. The environment provides the most crucial RPE features including tools for computer simulations and real-time experiments. On the Excel site, a workbook combining worksheet and Visual Basic for Application (VBA) programming is used. It implements RK4 solver for computer simulations and communicates with IO board for experiments. On the microcontroller (MCU) site, a dedicated application written by the authors measures sensor signals, feeds control signals and handles communication with Excel. For experiments the solution provides the soft real-time regime, sufficient for a wide range of applications. The authors applied the system to a relatively fast and unstable mechatronic plant of an inverted pendulum on a cart. The results of real-time control experiments are provided in the paper. They are compared to the results obtained with a professional MATLAB based environment to prove the effectiveness of the proposed solution.

Contrary to a popular opinion, Excel program is quite well-suited to scientific and engineering applications including those in metrology and automation fields. Several such uses are reported in the literature. An Excel based data acquisition application for rotary kilns is described in [1]. A method for interfacing Excel with a plant using a measurement and control extension card is given in [2]. A traffic monitoring application for a logistic centre is presented in [3]. A method of accessing PLC controller data using DDE and OLE mechanisms is described in [4]. A GPIB communication technique with measurement instruments is presented in [5]. Excel can also be used as a computer aided design tool in the control theory field, as shown in [6] and [7].

The article is organised as follows. An overall structure of the low-cost alternative RPE proposed in the paper is outlined in Section 2. Section 3 discusses the integration of system elements. The measurement and control board and the Excel application are described in Sections 4 and 5 respectively. Section 6 provides an experimental verification of the proposed RPE solution. Two last sections present conclusions and references respectively.

2. Alternative low cost RPE solution

The structure of the RPE proposed in the article is shown in Fig. 1. The environment consists of an Excel application running on a PC computer under Windows XP operating system, communicating via the USB bus with a measurement and control board based on an ARM Cortex-M core microcontroller and interfacing with the real-world plant.

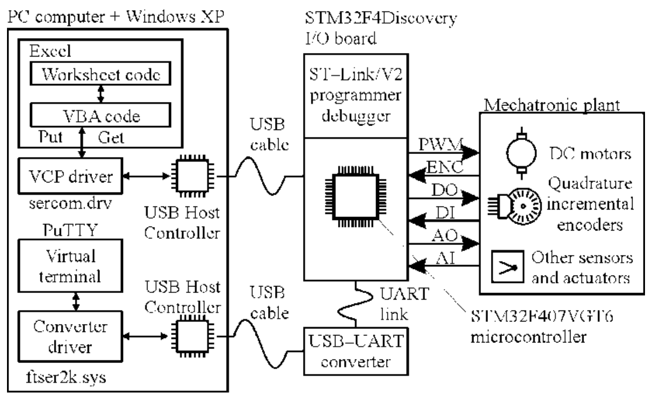

Fig. 1. Structure of the low-cost rapid prototyping environment

The workbook running under Excel combines worksheet formulae and VBA code in order to provide the user with tools for conducting computer simulations as well as soft real-time closed loop control experiments. In addition to the main time-critical computer to board USB connection for the measurement and control signals transmission, there is also an auxiliary communication channel utilising a USB to UART converter and providing a developer with a text mode service console for trace purposes.

Figure 2 presents a timing diagram for the RPE running a real-time experiment. The board measures sensor signals. This action is driven by a hardware clock and the board is also responsible for control system timing. Measurement results are transferred to the computer where the Excel application calculates control values and sends them back to the board. These actions are event driven. The Excel application is also responsible for data logging for further off-line analysis and processing.

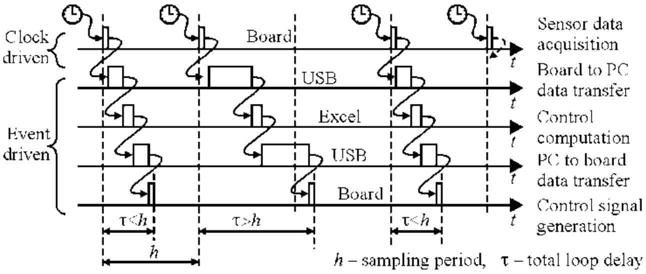

Fig. 2. Timing diagram for the real-time closed loop control

The following sections provide detailed description of the key elements constituting the proposed RPE.

3. Computer and board integration

Three popular communication links widely used in automation systems are: RS232 connection, USB bus and Ethernet network. The authors have chosen a direct USB link for real-time data exchange because it can provide a throughput much higher than RS232 and is supported by almost all PC computers hardware as well as Windows operating systems. Microcontrollers with USB peripherals are easily available, and unlike Ethernet, USB requires neither bulky additional electronic components nor an extensive software stack on the device site. USB 2.0 version, Full Speed mode, CDC class, and bulk transfer type are utilised in the proposed solution. The environment employs an EHCI host controller and a standard virtual com port (VCP) system driver on the PC computer site. VCP driver simplifies handling of communication by the Excel application and bulk transfer type allows to achieve higher exchange frequency and larger data stream than with interrupt or isochronous modes.

For the auxiliary computer-to-board trace connection, the USB to UART converter is employed as this link does not require high data throughput and that approach simplifies a microcontroller software structure.

4. Measurement and control board

The STM32F407 microcontroller with ARM Cortex-M4 core has been selected as the key component for a measurement and control board as it incorporates several peripherals well suited for automation purposes. A structure of an implemented system running on the MCU is shown in Fig. 3.

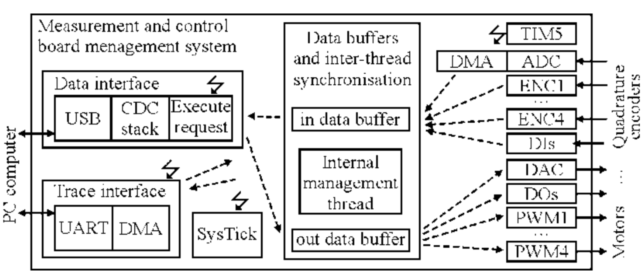

Fig. 3. Simplified measurement and control board software structure

A USB peripheral handles the main channel of the communication with the PC computer for measurement and control signals exchange. The auxiliary channel for trace and configuration purposes involves a USART module. The TIM5 timer triggers the measurement action of sensor signals and also acts as the main clock synchronising the control loop. Control signals are fed to actuators as soon as they are received from the PC computer. General purpose timers operating in special modes act as incremental encoder counters and PWM signal generators. ADC and DAC converters provide a mean to interface with analogue sensors and actuators while general purpose inputs and outputs accept or feed binary signals. A DMA controller facilitates high bandwidth internal data transfers while interrupt controller helps to synchronise software modules. A built-in debugging unit facilitates the system development stage.

5. Excel application for computer simulations and control experiments

Contrary to a popular but somewhat erroneous opinion, Microsoft Excel can make a useful tool for scientific and engineering purposes and its applications are, by no means, limited to econometrics or statistics fields. Excel supports array formulae that can be used to perform vector and matrix computations. Several built-in functions implement matrix algebra operations including matrix inverse, multiplication, or determinant calculation. Heterogeneous systems of linear equations may be solved with reglinp function implementing a multivariable linear regression algorithm. Excel libraries provide a vast set of mathematical and engineering functions including trigonometric and hyperbolic ones, Bessel special functions, and support for complex numbers algebra. For visual data presentation, Excel provides several types of 2D and 3D charts and plots supplemented with curve fitting tools. Iterative calculations feature available in Excel helps to implement a wide range of recursive algorithms including gradient optimisation methods and boundary problem solvers for partial differential equations. The Solver add-in implements a flexible optimisation algorithm that can be used for problems with a multivariable objective function and several constraints of different kinds. The Data Analysis add-in provides tools for fast Fourier transform calculation, histogram computation, series of random number generation, to name a few. To some extent Excel possesses database capabilities with worksheets in a workbook being an equivalent to database tables and with a few functions implementing SELECT like database operations.

Excel supports Visual Basic for Applications (VBA) programming language, providing a built in integrated development environment (IDE). An automatic code generation implemented as macro recording feature facilitates the program creation. One can write any VBA user defined function (UDF) that can be used in worksheet formulae (see Fig. 4). Conversely, VBA procedures can access and change contents of worksheet cells. Thus, VBA codes and worksheet formulae can be combined allowing a user to distribute computations between these two environments.

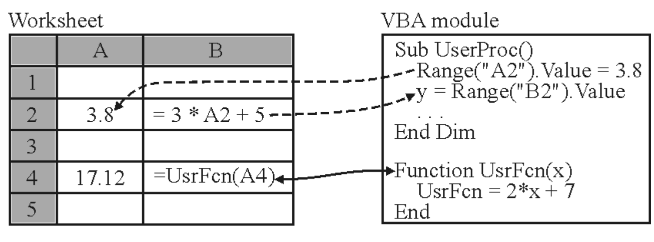

Fig. 4. Methods of combining worksheet and VBA code calculations

It is possible to call from a VBA code a routine defined in an external DLL library, including those implementing the Win32 API interface. A graphical user interface (GUI) can be handled by a VBA code with controls placed either directly in a worksheet or on a standalone dialog window. A workbook containing VBA module may be saved as an Excel add-in and can be then included in other workbooks to extend their functionalities. All these features combined with a relatively low price make Excel an attractive choice for an RPE implementation.

As in the proposed RPE the timer synchronising the real-time control loop is implemented on the IO board, the availability of a clock on the Excel side is not critical. However, such a feature can help to detect a control cycle deadline violation for diagnostic purposes during the development stage or in the regular use. One can measure the time using Timer VBA function, however, it provides relatively poor resolution equal to 15.625 ms on the ASUS F3E notebook PC computer tested by the authors. Considerably higher resolution of 279.37 ns can be achieved with QueryPerformanceCounter and QueryPerformanceFrequency functions provided by the Win32 API. Arguably, the finest resolution is provided by RDTSC (read time stamp counter) assembler instruction supported by newer Intel microprocessors. It equals 602 ps on the notebook tested by authors.

The VCP driver involved in computer to board communication creates in Windows operating system a special file that can be manipulated using standard file operation methods. The VBA programmer can use either intrinsic VBA statements (FreeFile, Open, Get, Put, Close) or Win32 API functions available in kernel32.dll system library (CreateFile, WriteFile, ReadFile, CloseHandle). The authors have conducted several communication experiments for both variants with round trip times (RTT) measured and logged. The test configuration involves a notebook PC computer, a four-port USB hub and STM32F4Discovery board. A Sub VBA procedure running in Excel comprises a For…Next loop repeating 10,000 times with the highest achievable speed the sequence of 16 byte long request packet sending succeeded by 48 byte long replay packet receiving. The time is measured with QueryPerformanceCounter Win32 API function. The board sends a reply immediately after the reception of a request. Thus, the observed cycle time is affected mainly by the communication and the method of VCP driver accessing from VBA code. The PC computer used in tests is ASUS F3E with Intel® Core™2 Duo T5450 @ 1.66 GHz CPU, 4 GB of RAM memory running Microsoft Windows XP Professional 2002 SP3 operating system and with Microsoft Office Excel 2007 SP3 application installed. The results of the tests for both communication variants are presented in Fig. 5 and Tab. 1. As performances are comparable, the native VBA solution is suggested as permitting a simpler and compact implementation.

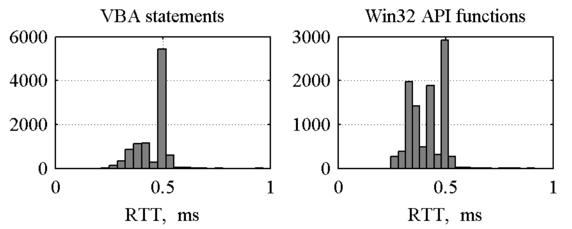

Fig. 5. Round trip time histograms for VBA statements and Win32 API functions

Tab. 1. Communication times statistics for a round trip transmission

VCP driver access method	min	mean	median	max
VBA statements	214 µs	456 µs	497 µs	967 µs
Win32 API functions	247 µs	414 µs	419 µs	913 µs

For off-line computer simulations of a control loop a framework presented in Fig. 6 can be employed. The models of the controller and the plant respectively are placed in two separate worksheets in the form of state space equations coded with Excel formulae. A 4^{th} order Runge-Kutta ODE solver algorithm is implemented as a VBA procedure activated from a worksheet acting as a user interface. The procedure iteratively interacts with controller and plant models, solves simultaneously their state equations and logs resulting time series in a fourth worksheet.

A similar approach of VBA and worksheet codes combination can be employed for real-time experiments (compare Fig. 1 and 8a). The control algorithm is coded in a worksheet in the form of state space equations or a static relationship. The RK4 solver for control signal computations is implemented in a VBA procedure. The procedure also handles the communication with the board as well as measurement and control data coding and decoding. Data is also logged into a separate worksheet. Precise time synchronisation is delegated to the measurement and control board as it is equipped with high resolution hardware timers.

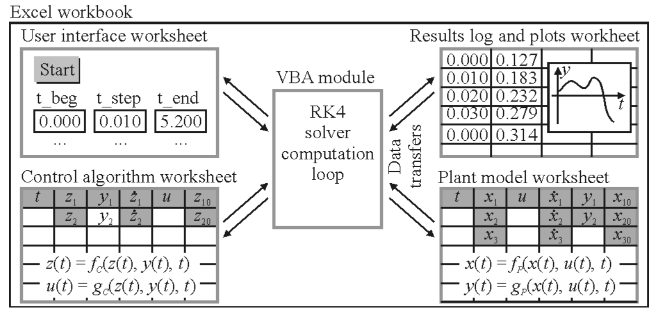

Fig. 6. The framework of an Excel application for computer simulations

6. Experimental verification

For the verification purposes, the proposed RPE has been applied to a pendulum on a cart mechatronic plant widely used in benchmark tests. A controller stabilising the pendulum in the upright liable position has been synthesised and computer simulations as well as real-time experiments have been conducted. The plant (Fig. 7) consists of a cart C moving along a rail R on a ball slide S and driven via a cogbelt B and a cogwheel W1 by a DC motor M fed with a PWM signal. A pendulum P can rotate freely relatively to the cart thanks to two ball bearings. The positions of the cart (s) and the pendulum (ϕ) are measured with two incremental quadrature encoders.

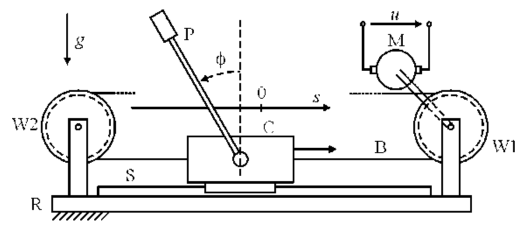

Fig. 7. Inverted pendulum on a cart mechatronic plant

A mathematical model of the plant can be obtained with the Lagrangian method. Then, a fourth order continuous time nonlinear state space equation system can be derived. For a given equilibrium point a linearised LTI model can be obtained and used for LQ controller design. An algebraic Riccati equation solving problem can be formulated as an optimisation problem, coded as worksheet formulae, and solved with Solver add-in. For a practical implementation, a standard multidimensional proportional controller thus obtained needs to be modified in several ways. A virtual limit switches have to be added and control signal limits have to be imposed. Velocities can be estimated with difference quotients and static friction in the ball slide can be compensated for with a control signal component depending on the cart velocity sign. The complete controller algorithm can be easily implemented in Excel as a few worksheet formulae. The sampling period of 10 ms has been selected based on dynamical properties of the plant. The control application for Excel has been built according to the framework described in the previous section. Fig. 8a shows a block diagram of all elements involved in a closed control loop realised in the system.

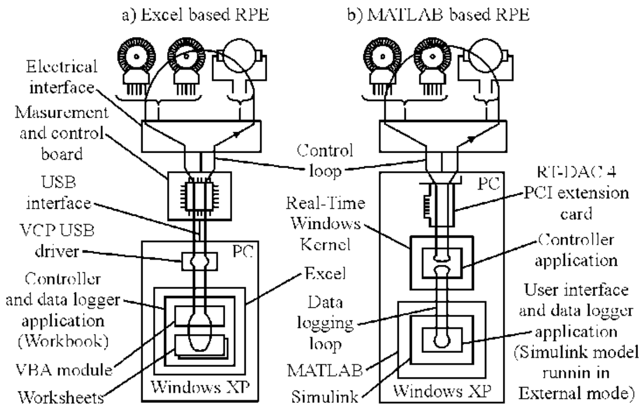

Fig. 8. Experimental configurations: a) Excel based, b) MATLAB based

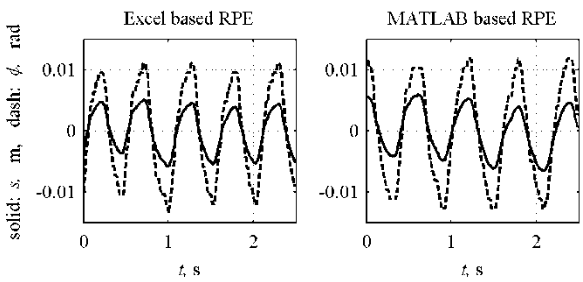

Fig. 9. Cart and pendulum positions series for Excel and MATLAB environments

For the sake of comparison, another implementation of a control system shown in Fig. 8b has also been tested. It utilises commercial RPE solution consisting of RT DAC 4 PCI card and MATLAB/Simulink software. The controller algorithm has been coded as a Simulink model and then a real-time application has been built. Several experiments have been conducted with both configurations shown in Fig. 8. Pendulum and cart position time series obtained in Excel and MATLAB based RPEs are presented in Fig. 9, respectively. They correspond to steady state conditions after the initial transients disappearance. The time series for both solutions are comparable. Performance index values calculated over a finite time horizon for both environments differ only by 7%. Close results confirm the effectiveness of the proposed Excel based RPE solution.

7. Conclusions

The article presents a low-cost rapid prototyping environment employing Microsoft Excel application and a measurement and control board based on a microcontroller. The environment provides a user with means for the control algorithm and plant model formulation, controller tuning, off-line computer simulations and soft real-time control experiments involving real-world plants. The solution proved to be adequate for relatively fast and unstable mechatronic systems. An underlying Excel application combines worksheet formulae and VBA procedures. The control engineer or researcher uses worksheet formulae. Thus, they work in an environment well familiar to Excel users and allowing a straightforward calculation coding. More sophisticated tasks that are unlikely to change frequently (communication routines, ODE solvers) are implemented using the VBA language and thus are hidden from the control algorithm designer.

The environment presents most features characteristic to an RPE solution. The Excel based system is certainly not as universal and flexible as professional environments but a definitely lower price makes it a reasonable alternative in case of projects with a limited budget or for educational purposes. As the solution uses a popular spreadsheet application, almost no time-consuming training is necessary for most users.

In further work, the authors intend to improve performances of the proposed RPE both in terms of the closed loop control cycle time and the sensor data throughput in order to apply it as a high-speed data acquisition solution for electrical power systems.

This work was partially funded by National Centre for Research and Development, grant No. PBS/1/A9/1/2012.

8. References

[1] Yao J., Du Q. L.: A simple data acquisition software for serial devices based on Excel VBA and its application in rotary kilns. Advanced Materials Research, vol. 591 593, pp. 1638 1644, 2012.

[2] Aliane N.: Data acquisition and real time control using spreadsheets: Interfacing Excel with external hardware. ISA Transactions, vol. 49, iss. 3, pp. 264-269, Jul 2010.

[3] Bottero M., Dalla Chiara B., Deflorio F. P.: Wireless sensor networks for traffic monitoring in a logistic centre. Transportation Research Part C: Emerging Technologies, vol. 26, pp. 99 124, Jan 2013.

[4] Pineda Sanchez M., Perez Cruz J.: Real time connection of programmable logic controllers to Excel spreadsheets. Proceedings of the 2002 IEEE International Symposium on Industrial Electronics, vol. 1, pp. 233 - 238, 2002.

[5] Hall K.: NSA using EXCEL '97 with GPIB allows making measurement and analysis less tedious. IEEE International Symposium on Electromagnetic Compatibility, vol. 1, pp. 515 518, 19 23 Aug 2002.

[6] Aliane N.: Spreadsheet based control system analysis and design: A command oriented toolbox for education. IEEE Control Systems Magazine, vol. 28, iss. 5, pp. 108 113, 2008.

[7] Mirapalheta G.C., Penedo S.R.M., Justo, J.F.: A feedback system dynamic response analysis by root locus method using Excel spreadsheet and XNumbers add in package. International Journal of Electrical Engineering Education, vol. 50, iss. 1, pp. 69 79, Jan 2013.

Received: 16.09.2014 *Paper reviewed* *Accepted: 05.01.2015*

Ph.D. Andrzej TUTAJ

He received his MSc degree in automatics and robotics from AGH University of Science and Technology, Krakow, Poland, in 2000 and the PhD degree in automatics and robotics from AGH in 2009. He is currently an associate professor at the Department of Automatics and Biomedical Engineering at AGH. His areas of interest include networked control systems, state and parameter estimation, model based control, and industrial automation systems.

e-mail: tutaj@agh.edu.pl

Ph.D. Jacek AUGUSTYN

He received the MSc degree in electronics from AGH University of Science and Technology, Krakow, Poland, in 1996 and the PhD degree in automatics from AGH in 2004. He is an associate professor in the Department of Automatics and Biomedical Engineering at AGH. He has authored two books about DSP programming and embedded systems. His main areas of interest are real time operating systems for embedded platforms and industrial control.

e-mail: jag@agh.edu.pl