New virtual laboratories presenting advanced motion control concepts

Martin Goubej, Alois Krejčí, Jan Reitinger

University of West Bohemia, Univerzitní 8, 301 00, Pilsen, Czech Republic
E-mail: mgoubej@ntis.zcu.cz, krejcia@ntis.zcu.cz, reitinge@ntis.zcu.cz

Abstract.

The paper deals with development of software framework for rapid generation of remote virtual laboratories. Client-server architecture is chosen in order to employ real-time simulation core which is running on a dedicated server. Ordinary web browser is used as a final renderer to achieve hardware independent solution which can be run on different target platforms including laptops, tablets or mobile phones. The provided toolchain allows automatic generation of the virtual laboratory source code from the configuration file created in the open-source Inkscape graphic editor. Three virtual laboratories presenting advanced motion control algorithms have been developed showing the applicability of the proposed approach.

1. Introduction

Presently, virtual laboratories are recognized as a powerful tool which is often used for education in many technical areas such as power industry [1], [2] or automation and robotics [3–5]. These labs have many advantages like high efficiency in students or operators training or low cost maintenance. They are often more demonstrative than real experiments and can be very helpful for people with motor disabilities [5]. Virtual labs can be employed for simulation of many complex scenarios which cannot be simply reproduced using physical models or real plants (e.g. external disturbance injection or system failure). Unfortunately, many laboratories cannot be directly used for controlling physical systems, because their algorithms cannot be easily converted into a real-time control code [6, 7].

There are two principal groups of virtual laboratories: monolithic applications and labs based on server - client architecture. Both ways have their pros and cons. Monolithic applications do not require internet connection for operation. There is no need for any server running additional program, which is in most cases Matlab [8], [9] or LabView [10]. On the other hand, SW platforms which are usually used to develop monolithic (self-contained) labs cannot provide sufficient tools to simulate virtual systems. For example, Easy Java/Javascript Simulations (EjsS) is often used to develop more self-contained labs which do not need remote server. However, the model cannot be built from mature function blocks which is problematic especially when creating complex control schemes [11]. Technology of self-contained labs with Matlab/Simulink compatible algorithm based on Java SE applet technology was described in [12]. Unfortunately, employment of the Java applets has become problematic due to the recent security policy changes in modern web browsers. Therefore, an innovated solution which does not rely on the Java platform was developed. It provides a unified framework for the automatic generation of both
virtual laboratories and human machine interfaces for real-time control of virtual models or real physical plants.

The virtual laboratory is based on the server - client architecture. A virtual model is developed in Matlab/Simulink environment using special block library called RexLib just as in [12]. This algorithm is executed on the server part in control system called REX [13]. The visualisation interface is assembled in open-source vector graphics editor Inkscape with a RexHMI extension, which is intended for easy creation of graphical Human-Machine Interfaces (HMI) [14]. The final vector graphic is converted into HTML5 and can be accessed on the client side via a web browser. This approach leads to platform independent solution which can be used on a wide range of target platforms.

The paper describes the individual components of the developed software framework. The proposed solution is employed for the generation of three virtual laboratories dedicated to various motion control problems including time-optimal position control, electronic camming and coordinated control of robotic arm. The laboratories present advanced features provided by motion control functional blocks implementing the industrial PLCopen Motion Control standard.

The outline of the paper is as follows: Section 2 presents the proposed toolchain for rapid HMI development. Section 3 deals with brief introduction to PLCopen Motion Control standard followed by description of three Virtual laboratories related to various control problems. Concluding remarks and ideas for future work are given in section 4.

2. Human Machine Interface

There are several ways how to design a webpage providing human machine interface for various virtual models or physical plants. First one is manual coding using HTML language. The unified framework contains a "low level" javascript API which deals with the interconnection between the simulation/control core and the HMI. However, this option is meant only for special adjustments. For rapid prototyping, several extensions for the Inkscape vector graphic editor were developed.

A basic principle for any HMI created by the framework is that there is only a single page which is scaled according to the resolution of the target device (desktop, tablet, mobile phone, etc.). This presumption allows to create multiplatform and multidevice HMIs. When one runs the Inkscape editor, there is only one main page. All the components inserted to the page will be automatically generated to the web page at the corresponding location. There are several extensions written in Python language supporting this behavior. The basic principle of the HMI generation is shown in Fig. 1.

The whole concept uses a set of SVG components and javascript files which control their behavior. These two parts are called HMI module. Each module contains configuration (SVG file) and control part (JavaScript file). The configuration is stored in SVG file inside the element <description> in JSON (Javascript Object Notation) format. When user inserts new component to the page, the configuration is cloned and then one can use the HMI extensions for Inkscape to alter it.

When the page is ready, it is exported to the final HTML file. There is a basic template which contains all the necessary libraries like jQuery, Three.js and also the HMI library from the framework. This template is extended with all the javascript controllers for the HMI modules inside the webpage. When the HTML file is loaded to the browser, the HMI library traverses all the components and starts their appropriate controllers. There are several HMI extensions which are described in the following section.

**Element editor** is the first extension (shown in Fig.2) which must be used. It serves for editing of individual HMI modules properties. The Element editor changes the JSON configuration
inside the <description> element inside the SVG group. One can edit Title, Module, Options and Connections using this tool.

Title is shown in the web browser after hovering the cursor above the element and is also used in Connections as shortening during the initialization.

Module uniquely identifies the type of the component and assigns the proper JavaScript functionality.

Connections tab contains names of signals that trigger the element behavior. There is a possibility to use an existing name after clicking on the Browse button.

Options tab edits some additional options of the component.

HMIConfig editor is similar to the Element editor (see Fig. 3). The options for the whole visualization project are edited here. HMIConfig editor is shown after calling the Element editor with no element selected. It contains three main tabs Connections, Options and HTML export. Connections are the signals used in the visualization which are assigned to the signals from the simulation/control core. If there is a proper scheme already running in the core, the signals might be browsed online.

Options tab contains an IP address of the target device and other options for the whole project.

HTML export tab contain configuration of the web page exporter.

Library Browser extension just opens the windows explorer with the selection of HMI modules.

HTML Builder extension compiles the web site code into the target folder, provided that a valid location has been set in the HMI Configuration menu. This extension unpacks the webpage template and inserts the content of the current SVG project. It also loads all the javascript controllers for the used modules and link them to the page. Such a exported project can be run from webserver or locally from generated file.

Group animation is the last extension (shown in Fig.4) which is different from the others. It creates a HMI module from any group by defining its behavior (translation, rotation, changes of color, size and opacity etc.). Such a module can be consequently edited by the Element editor.
3. Virtual Laboratories
This section describes three virtual laboratories which were created by means of the developed HMI framework. These labs are dedicated to specific motion control problems which emerge in many industrial applications. The goal is to present advanced functional blocks implementing the PLCopen standard which is briefly explained in the following section.

PLCopen is an independent organisation providing efficiency in industrial automation based on the needs of users. PLCopen members have concentrated on technical specifications around the IEC 61131-3 standard and its implementations in order to reduce cost in industrial engineering. The outcome includes standardized libraries for:

- Motion Control
- Communication
- Safety
- XML, etc.

PLCopen Motion Control: The motion control industry incorporates an extensive variety of incompatible solutions and systems in terms of various programming languages,
communication and application interfaces and hardware platforms. This incompatibility induces considerable additional costs for the end users due to prolonged commissioning time, complicated staff training and difficult integration of vendor-specific functions. There was a strong demand for standardization which initiated the establishment of the Motion Control Task Force which was formed within the PLCopen organization. This task force released a set of standardized programming interfaces for motion control systems in the form of a functional block library which covers a wide range of industrial application domains [15].

Three pillars of the newly developed platform have been defined as: performance, functionality and standardization (see Fig. 5). These objectives are achieved by means of the specific functional properties:

- **Simplicity** - ease of use, towards the application program builder and installation & maintenance
- **Efficiency** - in the number of Function Blocks, directed to efficiency in design
- **Consistency** - conforming to IEC 61131-3 standard
- **Universality** - hardware independent
- **Flexibility** - future extensions / range of applications

Only the user interface and overall behavior of the individual functional blocks is defined by the PLCopen standard. Particular implementation issues are left for a motion control
system supplier. The specification deals only with the trajectory generation layer of the motion control system providing setpoint values for the individual plant actuators. Hardware specific regulatory layer which tracks the commanded motion can be connected by means of a suitable communication interface (Fig. 6). Functional block library conforming the PLCopen standard has been developed at the University of West Bohemia. It is available as a set of C++ functions for various SW platforms including Matlab/Simulink and real-time control system REX. It consists of three subgroups of functional blocks (SingleLib, MultiLib and CoordLib) dedicated to different application requirements including single axis control, synchronized motion with electronic gear or cam and coordinated multi-axis control. Virtual laboratory presenting a fundamental functionality of the individual parts of the library were developed in the presented HMI framework.

3.1. SingleLib  Time-optimal position control of single degree of freedom mechanism
This laboratory represents the task of point to point position control which is very frequent in various industrial applications. The goal is to perform a smooth transition of a working mechanism from an initial displacement to a final destination in the shortest time possible while respecting the physical constraints of the system given by a maximum velocity, acceleration
and jerk (derivative of acceleration). This leads to a problem of state constrained time-optimal control of triple integrator plant. Most of the commercially available systems do not consider the jerk limit which is essential for a smooth shape of the resulting trajectory of motion. Either only acceleration limit is assumed which significantly reduces the complexity of the problem or a suboptimal heuristic solution is used. The presented functional blocks use a fully analytical time-optimal algorithm which was presented in [16].

**FUNDAMENTAL BLOCKS:**
- RM_Axis - Motion control axis
- MC_Power - Axis activation (power on/off)
- MC_Stop - Stopping a movement
- MC_Reset - Reset axis errors
- MC_MoveAbsolute - Move to position (absolute coordinates) - Block moves an axis to specified position as fast as possible. If no further action is pending, final velocity is zero, axis moves to position and stops.
- MC_MoveRelative - Move to position (relative to the execution point).

**Case Study:** This task represents point to point time-optimal control. One can activate the axis via the `MC_power` button. There are two ways how to set the end position: The first is by using the `MoveAbsolute` block. The user can change the parameters of this block (e.g. Position, Velocity, Acceleration). The motion can be started by clicking on the `Execute` button. The second way uses `MoveRelative` block. It has the same parameters as `MoveAbsolute` except that the final position is a relative distance to the actual execution (parameter Distance). User can switch to the trend or regulation scheme panel with a ComboBox. The trend shows the actual position, velocity and acceleration. Visualization of the motion is located on the bottom panel of the laboratory. The lab appearance is shown in Fig. 7.

![Figure 6: Single DoF Motion Control system - Interconnection of trajectory generator layer and cascade control scheme of a electrical drive](image)

3.2. **MultiLib - electronic cam and electronic gear**
The second laboratory deals with electronic camming. Mechanical cams transform a rotary motion into a linear one and allow keeping mutual synchronization of multiple moving mechanisms. They are used in numerous applications including piston pumps, packaging machines, textile producing machines, machine tools or transport systems. The mechanical cam
can be replaced by its electronic counterpart. An electronic control system keeps synchronization between an independently controlled (master) axis and a dependent (slave) axis. Their relative motion is derived from a prescribed displacement diagram - cam profile, which determines the relation between the master and slave position. The master axis can be a real physical drive or a virtual motion generator inside the control system software. Multiple slave axes can be assigned to a single master axis to achieve precise synchronization of several actuators with a variable overall speed of operation. The main advantages of the electronic cam include fast and easy change of the cam profile, reliable high-speed and high accuracy operation and no mechanical wear of the cam-follower mechanism which would lead to performance deterioration. The electronic gear is a special case of the electronic cam with a linear displacement diagram which results in constant velocity and position ratio between the master and slave axis (Fig. 8).

**FUNDAMENTAL BLOCKS:**
- MC_CamTableSelect - Cam definition
- MC_CamIn - Engage the cam
- MC_GearIn - Engage the master/slave velocity ratio
- MC_MoveVelocity (SingleLib) - Move with constant velocity

**Case Study:** The laboratory demonstrates the motion control in multiple axes connected through a cam and gear relation. User can switch on the axes (master and slaves) with the $MC\_power$
button. The cam profile is fixed and cannot be changed. Gear ratio and parameters of the master axis velocity (Velocity, Acceleration, Deceleration, Jerk and Direction) are specified in GearIn and MoveVelocity blocks. The motion can be started/stopped with MP_start/MP_stop buttons. The run of the simulation is displayed in the trend on the bottom side (see Fig. 9).

Figure 8: Cam and gear

Figure 9: Virtual laboratory of electronic cam and electronic gear

3.3. CoordLib  Coordinated control of robotic manipulator AGEBOT

Robotic manipulators are widely used in many industrial applications (e.g. welding, soldering, painting, material handling or machining). Commercial robotic arms manufacturers have developed several different programming languages and application interfaces. The PLCopen standard provides a unified programming standard in the form of functional blocks specifically designed for the coordinated control of multi-axes machines. The geometric shape of desired trajectory of motion is usually parameterized in the machine or tool/workspace coordinate
system with the use of a proper interpolation algorithm (e.g. linear, circular or spline trajectory). Time path of the executed motion is generated based on the desired federate specified by the user. The resulting trajectory is consequently transformed into the axes coordinate system using a machine specific kinematic transform block which computes a contribution of the individual axes to the coordinated motion.

**FUNDAMNETAL BLOCKS:**

- **RM_AxesGroup** - Motion control axes group
- **MC_MoveLinearAbsolute** - linear movement on an axes group
- **MCP_SetKinTransform_Agebot** - sets a kinematic transforms between the machine and actuator coordinate systems

AGEBOT was designed as a special robot architecture which consist of two main parts [17], [18]. Serial manipulator (SM) and parallel manipulator (PM). SM ensures basic positioning of the end effector including the translations in x, y, z axes and orientation of the longitudinal axis of PM. This motion is used for handling of parts which are to be exposed of cleaning process in cleaning chambers. PM plays an important role in precise positioning of handled parts inside cleaning chambers because PM allows full orientating about x, y, z axes. Mechanical design of PM is allows separation of vulnerable components (motors, sensors, etc.) from an aggressive environment inside the cleaning chamber. Overall view of the AGEBOT is shown in Fig. 10.

**Case Study:** The laboratory presents basic principles of coordinate motion control on the serial part of the AGEBOT robotic manipulator. The robot axes can be activated by **MC_power** button. Target position of the effector in a 2D plane can be set with x, y and angle coordinates (tilt angle of the last arm). This motion is executed by **MoveLinear** block by clicking on the **Execute** button. There is possibility to manually operate the individual robot axes. User can switch between the visualisation and regulation scheme panel with a ComboBox. Trend chart is displayed on the bottom of the lab. The HMI is shown in Fig. 11.

Figure 10: AGEBOT from 3D CAD visualization
Figure 11: Virtual laboratory of robotic manipulator AGEBOT in 2D
4. Conclusion
The presented software tools allow simple creation of virtual laboratories with attractive visual appearance. Rapid development and automatic code generation is supported by a set of components integrated into the Inkscape graphic editor. The provided framework allows simple connection to REX control system which offers the functionality of real-time simulation using an extensive set of Matlab compatible functional block library. However, the visualization tools can be adjusted to another simulation core using a proper communication interface. Case study dealing with advanced motion control algorithms is provided as an example of use. THE VIRTUAL LABORATORIES ARE FREELY ACCESSIBLE AT WEBSITE: http://lab.fav.zcu.cz/motioncontrol. The developed tools can also serve for creation of human machine interfaces for model or hardware-in-the-loop simulators as well as for real-time control of a physical plant. Future work will be dedicated to extension of the advanced functions of the visualization framework, especially in the field of 3D graphics.

Acknowledgement
This work was supported by the Technology Agency of the Czech Republic – project No. TE02000103 and by Czech Ministry of Education, Youth and Sports – project No. LO1506.

References
[1] Redel-Macas M, Pinzi S, Martinez-Jimenez M, Dorado G and Dorado M 2015 Virtual laboratory on biomass for energy generation Journal of Cleaner Production
[2] Iovan M, Surianu F D and Molnar-Matei F 2015 Virtual laboratory for power quality study Procedia - Social and Behavioral Sciences
[3] Prieto-Blazquez J, Arnedo-Moreno J and Herrera-Joancomarti J 2008 An integrated structure for a virtual networking laboratory IEEE Transactions on Industrial Electronics ISBN 0278-0046
[4] Villar-Zafran A, Zarza-Sanchez S, Lazo-Villa J A and Fernandez-Cant R M 2012 Multiplatform virtual laboratory for engineering education 9th International Conference on Remote Engineering and Virtual Instrumentation (REV)
[5] Duarte M, Butz B P, Miller S M and Mahalingam A 2008 An intelligent universal virtual laboratory (uvl) IEEE Transactions on Education ISBN 0018-9359
[6] Koretsky M, Amatore D, Barnes C and Kimura S 2008 Enhancement of student learning in experimental design using a virtual laboratory IEEE Transactions on Education
[7] Urán S and Jezernik K 2008 Virtual laboratory for creative control design experiments IEEE Transactions on Education
[8] Elmagroudi B and Ziyyat A 2012 Virtual laboratory for teaching electromagnetism using matlab Proceedings of 2012 International Conference on Interactive Mobile and Computer Aided Learning (IMCL)
[9] Sengupta D, Jain N and Kumar C S 2013 An educational website on kinematics of robots IEEE Fifth International Conference on Technology for Education
[10] Yang-Mei L and Bo C 2014 Electronic circuit virtual laboratory based on labview and multisim 2014 7th International Conference on Intelligent Computation Technology and Automation
[11] Christian W and Esqueambre F 2009 Modeling Science: From Free Fall to Chaos
[12] Reitinger J, Čech M, Schlegel M and Bakla P 2014 New tools for teaching vibration damping concepts: Constlab.eu IFAC Proceedings Volumes (IFAC-PapersOnline)
[13] Severa O and Čech M 2012 Rex - rapid development tool for automation and robotics Proceedings of 2012 IEEE/ASME 8th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications
[14] REX Controls 2015 Software downloads URL http://www.rexcontrols.com/download
[15] PLCopen 2015 Ploopen motion control Available online: http://www.plcopen.org/pages/tc2_motion_control/ accessed on: 2015-9-16 URL http://www.plcopen.org/pages/tc2_motion_control/
[16] Blaha L, Schlegel M and Možina 2009 Optimal control of chain of integrators with constraints 17th International Conference on Process Control 2009
[17] Svejdja M and Goubej M 2012 Innovative design and control of robotic manipulator for chemically aggressive environments 13th International Carpathian Control Conference (ICCC)
[18] Svejdja M and Goubej M 2013 Dynamic analysis and control of robotic manipulator for chemically aggressive environments IEEE International Conference on Mechatronics (ICM)