

On Chip Rapid Control Prototyping for Education

Romain Delpoux, Lubin Kerhuel, Vincent Léchappé, Arnaud Duvernois

▶ **To cite this version:**

Romain Delpoux, Lubin Kerhuel, Vincent Léchappé, Arnaud Duvernois. On Chip Rapid Control Prototyping for Education. 21th IFAC World Congress, Jul 2020, Berlin, Germany. hal-02928430

HAL Id: hal-02928430
https://hal.science/hal-02928430

Submitted on 2 Sep 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

On Chip Rapid Control Prototyping for Education

Romain Delpoux, Lubin Kerhuel, Vincent Léchappé and Arnaud Duvernois

***Abstract*— In this note various test benches for Rapid Control Prototyping (RCP) targeting microcontrollers are presented. The methodology relies on a Matlab/Simulink interface which makes the target configuration and coding easier. Developing low level embedded code is bypassed by a high-level implementation which is straightforward for control system engineers. These benches are intended for students, engineers or researchers looking to validate the effectiveness of their control algorithms on low cost industrial targets.**

I. INTRODUCTION

Digital controllers are increasingly present in industrial applications (automotive, aeronautics, space, industry...). However, manufacturers seek solutions to reduce global cost while maintaining good performances. Developing embedded code for microcontrollers requires skills that are out of scope of the curriculum of control system engineers which is typically limited to high level tools like Matlab®/Simulink® and RCP tools like dSPACE® or Speedgoat. This task is often left to embedded system engineers who are specialists of hardware architecture but have little knowledge in control theory. According to some recent discussions with automotive and aerospace specialists, there is a lack of engineers able to deal with the full process from control design to hardware implementation. This observation is also confirmed by the increasing desire of students in control system courses to learn how to implement algorithms directly on dedicated hardware. From this assessment, it has been decided to develop test benches using simple and fast methodology for RCP based on the interface of a microcontroller directly with Matlab/Simulink by taking advantage of the graphical interface provided by the Microchip Technology MPLAB® device blocks for Simulink. It is important to note that unlike some existing setup [1], the objective is not to 'hide' and make totally transparent the code generation but on the contrary to make it accessible in order to tackle the problems that arise from this implementation. In addition to the teaching benefit, the second objective is to fill the gap between convenient but expensive RCP platforms and complex but cheaper microcontroller tools. The developed methodology will speed up the implementation of advanced control algorithms on real hardware by removing the phase of coding on the target. A dsPIC® Digital Signal Controllers (DSCs) and its dedicated MPLAB toolbox for Simulink has been used for this article but other manufacturer DSCs or Microcontroller Unit (MCU) may benefit from this methodology whenever a dedicated toolbox exists. The goal of demonstrators presented in this note is to propose low cost hardware to illustrate real-time control for linear and nonlinear Single Input Single Output (SISO) and Multi Input Multi Output (MIMO) systems regardless of the targeted architecture. The proposed control algorithms are available online[1].

Many works in the literature present RCP solutions, however, they are usually aimed at control teaching on a dedicated board or platform, and do not propose a flexible solution that allows a deep analysis of the implementation problems: [2] (with Lego Mindstorms NXT), [3], [4] (with Arduino), [1] (with a Digital Signal Processing (DSP) system), [5] (16 bit microcontroller). Some other works on RCP are dedicated to research purposes but they are usually focused on illustrating the efficiency of the control algorithm than providing a handy interface to analyse in detail the technical issues related to hardware coding : [6] (dsPIC DSCs), [7] (RTAI-Lab).

The contributions of the proposed method is twofold i) it speeds up the control design process using the well known Simulink interface to go back and forth from simulation to implementation on hardware. The hardware configuration is as easy as a dSPACE or Speedgoat RCP setup, but with the major difference that it targets low cost microcontrollers which can embedded in comercial or custom designed board. ii) it smooths the learning curve for students in control system course by providing a handy interface that offers a way to gradually introduce some implementation issues (data acquisition, discretization, fixed-point conversion, Pulse Width Modulation (PWM) generation,...).

II. DEMO BENCHES

This section is devoted to the description of different test benches used for labs at INSA de Lyon concerning control theory. The proposed demonstrators allow to easily implement the notion of linear and non-linear control for SISO or MIMO systems. The idea for these lab benches is to propose low-cost hardware solution for RCP. Note that all the benches described below are using Matlab/Simulink. Futher details on RCP are given in Section IV.

A. Mobile robots

In this section we present two mobile robots the first is a line follower (Fig. 1) and the second is an inverted pendulum (Fig. 2). Both robots are equipped with the 16-bit dsPIC

R. Delpoux, V. Léchappé and A. Duvernois are with Univ Lyon, INSA Lyon, ECL, Université Claude Bernard Lyon 1, CNRS, Ampère, F-69621, Villeurbanne, France. surname.name@insa-lyon.fr

L. Kerhuel is with Microchip Technology Inc, F-64100, Bayonne, France. lubin.kerhuel@microchip.com

[1] www.ctrl-elec.fr & lubin.kerhuel.eu

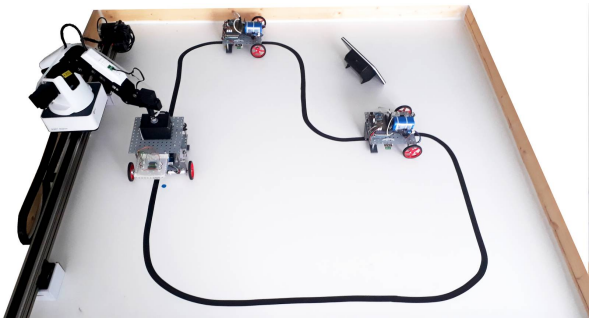


Fig. 1. Line follower robot



Fig. 2. Inverted Pendulum

DSCs 33FJ126MC802 because it embeds two Quadrature Encoder Interface (QEI) which allows to properly control the two DC motors on each wheels. The line follower detects the line to follow through a serial link from a camera then a linear controller is designed to ensure speed and orientation control. The inverted pendulum use the acceleration and the angular rate acquired from an Serial Peripheral Interface (SPI) bus to reconstruct the pendulum attitude. The synthesized control law is a Linear Quadratic Regulator (LQR) using a linearized model of the pendulum. Both systems represent an interesting Lab of Engineering with increasing complexity.

B. Motor Control

The benches described in this subsection have been developed for the Electrical department at INSA de Lyon France for students specialized in motor and converter control. They are controlled using the Microchip MCLV-2 demo board equipped with the 16-bit dsPIC DSCs 33EP256MC506. The first bench is a DC motor (Fig. 3). This bench was developed to help the student understanding the real time environment required for such a control. For speed control, PWM interface, Analog Digital Converter (ADC) interface synchronized with PWM to measure current and QEI to measure the speed are needed. The proposed controller is a linear state feedback considering the Direct Current (DC) and the velocity. From this configuration, more advanced controllers can be designed on Permanent Magnet Synchronous Motor (PMSM) or Induction Motor (Fig. 4). On the picture both motors are connected by their shaft. This bench permits to implement on these motors basic control algorithm as well as advanced control algorithm such as field oriented control. Both motors can be used as motor or generator which play the load to the other motor.

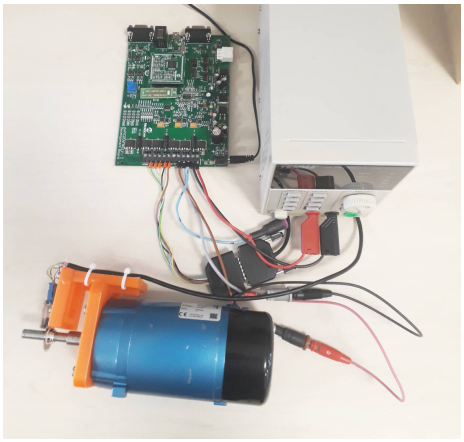


Fig. 3. DC motor

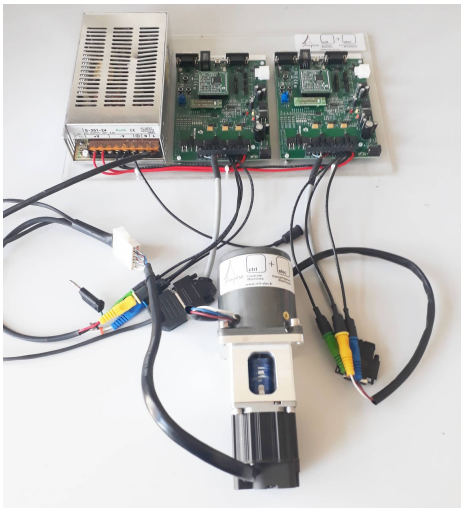


Fig. 4. AC motors

III. Model validation

In addition to the real time implementation of advanced control algorithms, this architecture gives the possibility to compare the theoretical model with real data from the experiment to ensure that the model is accurate. This phase makes possible to refine the simulation model and consider the maximum of uncertainties in the validation model. It is important to differentiate the validation and control models. The **validation model** must take into account most of the physical phenomena in order to be the closest to the experimental results. The **control model** designed is based on the dominant dynamics in order to propose an

efficient control law. The application of the control law on the validation model shows the robustness of the control and highlights some potential problems that could arise on the real system. Possible issues may arise from saturations due to high frequency noise derivative or dynamics not taken into account by the control model.

Real-time experiments data can be sent using the UART to MATLAB/Simulink. The received logs can be used in simulation to compare an input sent both to the model and the real system (see Fig. 5).

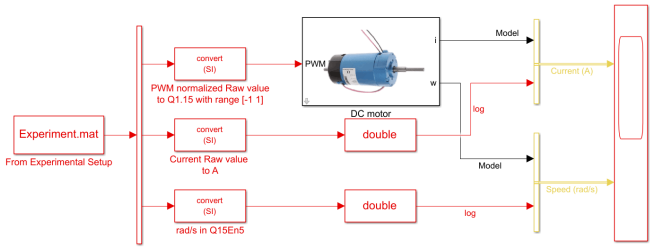


Fig. 5. Use of log as simulation input

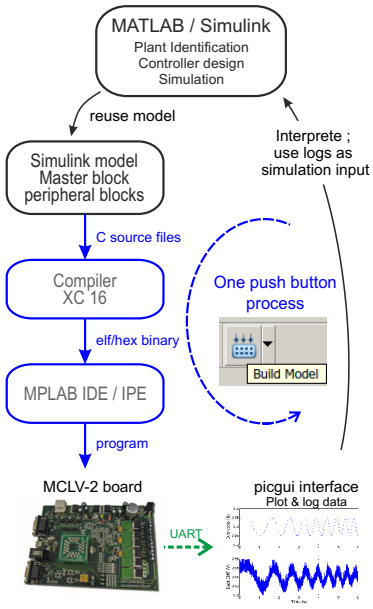


Fig. 6. Rapid Prototyping Scheme

IV. Rapid Control Prototyping on dsPIC DSCs

This section details the RCP solution. Matlab/Simulink is used with the Embedded Coder and the fixed-point toolbox from MathWorks. The add-on MPLAB device blocks for Simulink adds to Simulink the capability to target up to 360 Microchip microcontrollers (dsPIC, PIC32, SAMx7). Theses blocks rely on XC16 & XC32 compiler and the MPLAB X IDE [2] to compile the generated code and program the board from the Simulink interface with a single 'Build' push button on top right of the simulink interface (see Figure 6). The compiled model has a discrete time solver and few dsPIC peripheral blocks.

A. Rapid Control Prototyping (RCP)

An example of the Simulink interface using the MPLAB device blocks is represented on Fig. 7. Here a DC motor control scheme is proposed. On this example the control voltage is applied as PWM signal driving an H-bridge converter. The control algorithm inputs are the current and velocity measurements. The velocity is obtained from an incremental encoder through the use of the QEI peripheral. The current is measured using a shunt resistor, conditioned with the dsPIC DSCs internal op-amps and converted with the ADC peripheral. The ADC conversion is synchronized with the PWM period so that it ensures that the ADC sampling time is taken when the low side of the H-bridge is on. Speed and current control loops are sampled at different rate. An option in Simulink highlights with colors the different sampling rates in a model as presented on Fig. 7. The red color represents the fastest sampling rate. Here the sampling frequency for the control of the fast current dynamics is $F_{\text{elec}} = 20$ kHz. The green color represents the sampling rate for the mechanical dynamics, here $F_{\text{meca}} = 1$ kHz.

[2]blockset is distributed as suppot package Add-on www.mathworks.com ; Compiler and IDE are tools available from www.microchip.com.

Standard Simulink blocks generate code, which compiles on a dsPIC DSCs target. Any model can thus be embedded with respect of real-time constraints. Simulation are typically performed in the continuous time domain. A differential equation solver is used to solve simulation outputs. The real-time embedded software does not implement such solver so a discrete-time equivalent controller must be used.

In a **Single tasking program**, all tasks started at a given base model time step must be completed within the end of that time slot to respect real-time constraints.

In a **Multi-tasking scheduler**, a monotonic-rate scheduler is implemented where higher rate tasks have higher priority and interrupt lower task rate which have a lower priority. This simple priority rule is well suited for control algorithms. Its limited implementation penalty in execution time worth the gain in flexibility.

Figure 8 presents a timing analysis of the cascaded PI algorithms. One pin driven by the 'CPU load' block and two pins driven by the 'task state' block show respectively the CPU state (black curve where 1 is busy and 0 is idle state), the fast task D1 (red) and slow task D2 (green) respective start and stop on rising and falling edges. The lower graph is a magnification of the higher priority task D1 preemption of the lower priority task D2. Note that, the slow D2 task pin state is not cleared when being preempted but exclusively when task is completed. The shaded region in the tasks D1 and D2 shows the CPU execution on each task.

The code performances are surprisingly good. Peripheral are handled in the background through interrupts. As a result, peripheral blocks for input peripherals are not blocking as they just have to read the last result obtained from that peripheral; keeping the CPU time dedicated to the control algorithm. For example, the base time step is typically triggered by the ADC block end of conversion interrupt

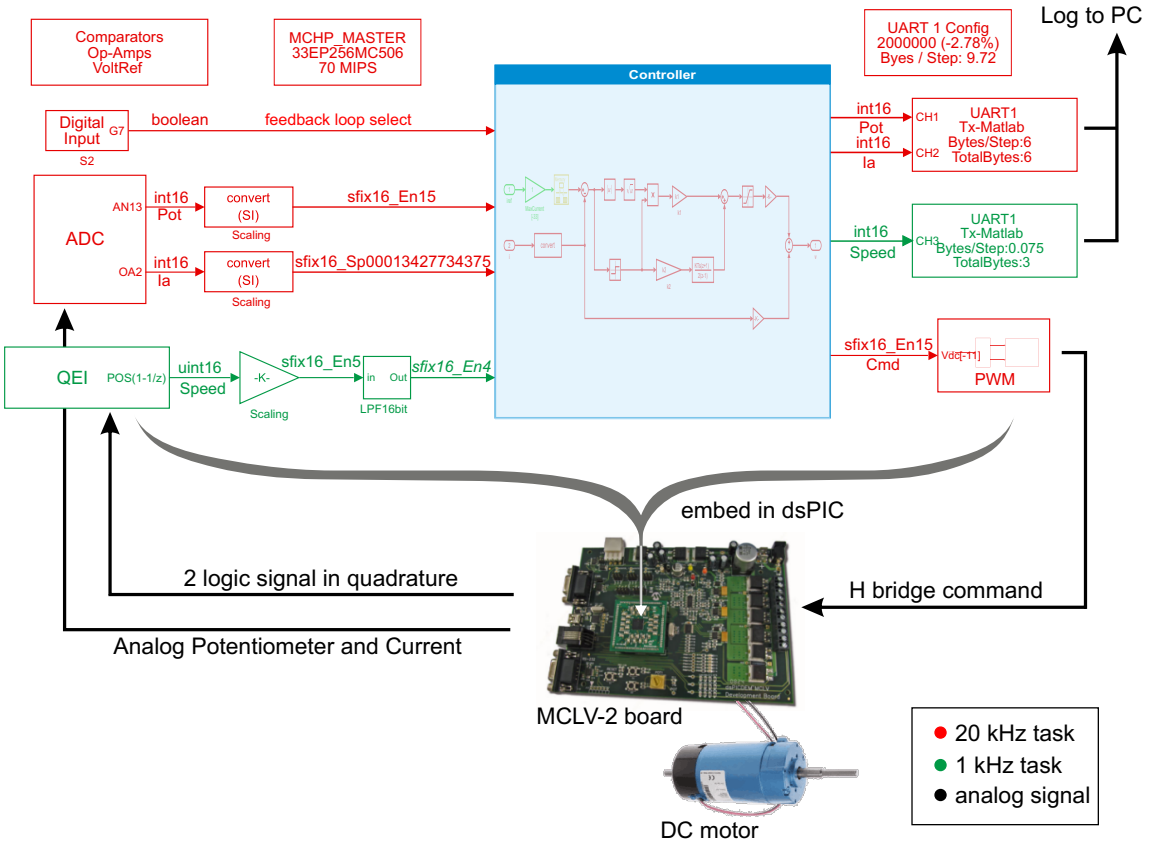


Fig. 7. Control global scheme example. The model is compiled and run as a standalone program on the targeted board

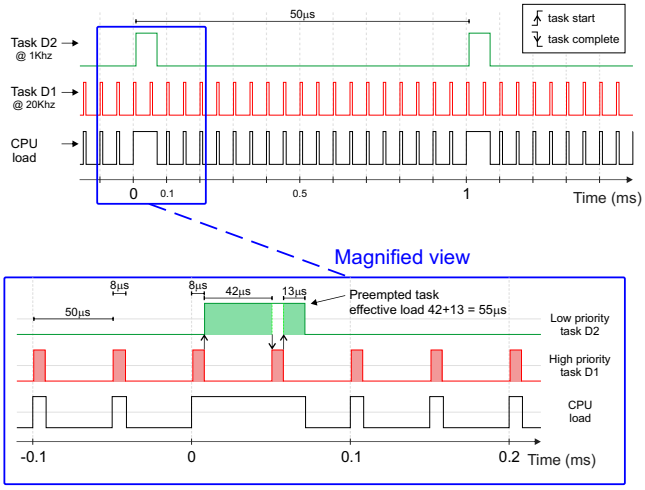


Fig. 8. Scope measurement of task execution for a Multi-rate model with a multi-tasking rate monotonic scheduler

thus when the ADC block is evaluated within the algorithm, the conversion results are already available. The ADC block does not have to wait for a sample and hold sequence nor by the following conversion sequence. The same remark holds for others peripheral blocks like the UART transmission/reception, the SPI or I2C blocks.

V. Conclusion

This note presents different benches for real time embedded control developed for education. Using RCP that targets microcontrollers, the proposed benches are very low cost but enable high level control theory implementation. The note also highlight various points required for a real-time embedded implementation like fixed point and multi-rate scheduling.

References

[1] D. Hercog, M. urkovi, and K. Jezernik, "DSP Based Rapid Control Prototyping Systems for Engineering Education and Research," in *Proceedings of the 2006 IEEE Conference on Computer Aided Control Systems Design*, pp. 2292–2297.

[2] C. Rodrguez, J. L. Guzman, M. Berenguel, and S. Dormido, "Teaching real-time programming using mobile robots," in *11th IFAC Symposium on Advances in Control Education ACE 2016*, vol. 49, pp. 10–15.

[3] H. M. Omar, "Enhancing automatic control learning through Arduino-based projects," vol. 43, no. 5, pp. 652–663.

[4] A. Soriano, L. Marn, M. Valls, A. Valera, and P. Albertos, "Low Cost Platform for Automatic Control Education Based on Open Hardware." in *Proceedings of the 19th IFAC World Congress*, vol. 47, pp. 9044–9050.

[5] R. Chakirov and Y. Vagapov, "Rapid Control Prototyping Platform for the Design of Control Systems for Automotive Electromechanical Actuators," in *5th IFAC Symposium on Mechatronic Systems*, ser. 5th IFAC Symposium on Mechatronic Systems, vol. 43, pp. 646–651.

[6] M. Lizarraga, G. H. Elkaim, and R. Curry, "SLUGS UAV: A flexible and versatile hardware/software platform for guidance navigation and control research," in *Proceedings of the 2013 American Control Conference*, pp. 674–679.

[7] R. Bucher and S. Balemi, "Rapid controller prototyping with Matlab/Simulink and Linux," vol. 14, no. 2, pp. 185–192.