Rapid Control Prototyping Applications using TI C2000 DSP

PETRU DOBRA¹, RADU DUMA¹, DANIEL MOGA¹, RADU ADRIAN MUNTEANU², MIHAI MUNTEANU³

¹) Department of Automatic Control
²) Department of Electrical Measurement
³) Department of Electrotechnics
Technical University of Cluj
St. C. Daicoviciu 15, 400020 Cluj-Napoca
ROMANIA

Abstract: - The paper presents two rapid control prototyping applications: velocity control of a DC motor and adaptive system identification and control of an automotive alternator. The control algorithms for the applications run on a TMS320F2812 DSP. Once the desired functionality has been captured and simulated, using MATLAB/Simulink/Embedded Target for Texas Instruments (TI) C2000 DSP environment can be generated code for the DSP. All task assignments to processor are automatically made by the software.

Key-Words: - Embedded Target, DSP, DC motor, Rapid control prototyping, Digital Signal Controller, Matlab/Simulink.

1 Introduction
Developing controllers for applications (electrical drive systems) means large expenditure, when performed with usual development methods. The workload comprises development of a mathematical model as well as algorithm design and implementation, off-line simulation, and optimization. The whole process has to be restarted on occurring errors or divergences, which makes the development process time consuming and costly [1].

Rapid Control Prototyping (RCP) is a way out of this situation, especially if the control algorithm is complex and a lot of iteration steps are necessary. RCP requires two components: a Computer Aided Control System Design (CACSD) software and a dedicated hardware with hard real-time operating system (Fig.1). The graphical programming approach removes the need to write software by hand and allows the engineer to focus instead on improving functionality and performance. Complete system design is carried out within the simulation environment.

Fig.1. General architecture of an RCP system.

The CACSD tool used in the implementation of the presented applications is Matlab/Simulink/Real-Time Workshop. The target processor is a TMS320F2812 DSP.

With the great diversity of applications, a development environment must be flexible and provide exactly the functionality necessary for efficient problem solving. Simulink software from Mathworks is such a graphical modelling tool. A recent toolbox of Simulink is the 'Embedded Target' for Texas Instruments' C2000 DSP platform.

2 C2000 Embedded Target
Simulink, Real-Time Workshop, the Embedded Target for TI C2000 DSP, and Link for Code Composer Studio (CCS) provide an integrated platform for design, simulation, implementation, and verification of embedded control systems on standard and custom C2000 DSP targets [2] (Fig.2).

Fig.2. The steps from model to the implementation.
Simulink models are constructed from standard libraries. Embedded Target provides blocks specific to the C2000 DSP family: I/O, CAN, PWM, QEP, Read From Memory, and Write To Memory. A Target Preference block has to be added to the model. It does not connect to any other blocks, but stands alone to set the target preferences for the model (build options for the compiler, assembler and linker which will be invoked to generate the executable image file for download to the DSP).

Once the desired functionality has been captured and simulated, can be generated code for the DSP. Simulink/Real-Time Workshop generates a C language real time implementation of the model, creates and populates a CCS project with the code. CCS is opened, the project compiled and linked, and the image file downloaded to the target DSP.

The code may be instrumented with Real Time Data eXchange modules to stream data to and from the target. These are additional I/O blocks from the Embedded Target library.

A key feature of Embedded Target is its ability to generate efficient DSP code [3].

3 Target Processor
The control algorithm runs on eZdsp F2812 target equipped with a TI TMS320F2812 DSP, for fast fixed-point calculation at 150 MHz (6.67 ns cycle time).

The TMS320F2812 belongs to a group of devices that are called “Digital Signal Controller (DSC)”. DSC is a new type of microcontroller, where the processing power is delivered by a DSP - a single chip device combining both the computing power of a Digital Signal Processor and the embedded peripherals of a single chip computing system.

The board can be adapted to a wide range of closed-loop applications due to its motor control peripherals (two event managers (EVA, EVB)), 16 12-Bit ADC channels with fast conversion rate: 80 ns/12.5 MSPS, up to 56 general purpose I/O (GPIO) pins, high-performance 32-bit CPU, three 32-Bit CPU-timers.

The multiple bus architecture, commonly termed “Harvard Bus”, enables the F2812 to fetch an instruction, read a data value and write a data value in a single cycle. All peripherals and memories attached to the memory bus will prioritize memory accesses.

4 Velocity Control of DC Motor
A digital control application for a DC motor is implemented, in which a tuning method is proposed for DC motors, using DSP from TI. Experimental application for the DC motor illustrates the effectiveness and the simplicity of the proposed method for controller design.

4.1 DC Motor
The DC Motor comprises the motor, a shaft sprocket with 32 teeth, an optical gear tooth sensor for speed measurement, and a DC/DC electronic converter. The DC/DC electronic converter is driven by pulse width modulated (PWM) signals from the F2812 target. The whole setup is shown in Fig.3.

![DC Motor Control Setup](image)

4.2 Motor Driver
The interface between the F2812 target and the motor is implemented by an electronic circuit. The motor is supplied at 12 V and its speed is controlled using a PWM signal.

The speed is measured using an optical encoder sensor and a shaft sprocket with 32 teeth. The signal from the sensor is conditioned using a 74HCT14 Trigger S/th circuit and is applied at the input pin of a capture and log transition unit of the eZdsp F2812 target. Fig.4 presents the hardware setup at work.

4.3 Controller Implementation
The Simulink model is constructed from blocks of the C2000 Embedded Target Library which are used to represent algorithms and peripherals specific to the C2800 DSP family.

The block diagram of the Simulink closed loop control structure is shown in Fig.5 and the feedback control system in Fig.6.

The system works at a sampling rate of 1 ms. For the DC Motor speed measurement an Capture Unit of the Event Manager is used. In order to reduce
signal jitter or period fluctuation a first order delay filter is used. The capture unit log transitions detected on the capture unit pin by recording the times of these transitions into a two-level-deep FIFO stack.

Fig. 4. The hardware setup at work.

Fig. 5. Simulink block diagram of the DC Motor control system.

Fig. 6. Simulink block diagram for motor speed measurement.

4.4 Controller Tuning

A key method for auto-tuning is to use the relay feedback method [4].

The parameters of the controller can be computed using Ziegler-Nichols tuning methods or tuning with specify phase and amplitude margin. Better results were obtained by specifying the phase and amplitude margin. The implementation of the latter tuning method is presented below. Consider a situation where one point on the Nyquist curve for the open loop system is known. With PID control it is possible to move the given point on the Nyquist curve to an arbitrary position in the complex plane.

A limit cycle is obtained by introducing in the control loop a relay type nonlinearity (Fig.7). The Simulink block diagram for obtaining the limit cycle is shown in Fig.8.

Fig. 7. Relay feedback diagram.

Fig. 8. Simulink block diagram of Ziegler – Nichols ultimate period method.

If the error is smaller than zero the PWM duty ratio will be 95% and if it is greater than zero the duty ratio will be of 80%, thus obtaining a limit cycle. RTDX blocks are used to fetch and retrieve data from the target. For this in the Simulink model From RTDX and To RTDX blocks have been added.

The obtained limit cycle is shown in Fig.9.

Fig. 9. Limit cycle.

From Fig.9 the critical period $T_u$ is determined. The ultimate gain $K_u$ is computed using relation:

$$K_u = \frac{4d}{\pi a}$$  \hspace{1cm} (1)

The open loop transfer function with PID control is:

$$k_p (1 - \frac{1}{j\omega T_i} + j\omega T_d)H_{DCM} \cdot$$  \hspace{1cm} (2)

Consider the phase of the PID controller at $\omega_0 = 2\pi/T_u$ to be $\gamma_m$.

$$\gamma_m =$$
\[
\arctg\left(\omega T_d - \frac{1}{\omega T_i}\right) = \gamma_m. \quad (3)
\]

Solving equation (3), for \( T_i = \alpha T_d \) with \( \alpha \in (2 \div 6) \), is obtained the derivative time constant:
\[
T_d = \frac{1}{2\alpha_0} \left(\tan \gamma_m + \sqrt{4/\alpha + \tan^2 \gamma_m}\right). \quad (4)
\]

If the magnitude of the open loop transfer function is specified to \( k_m \) simple trigonometric calculations give:
\[
k_p = k_m K_u \cos \gamma_m \quad (5)
\]

The drive can be controlled using a graphical user interface (GUI), created using GUIDE, the MATLAB Graphical User Interface development environment, which provides a set of tools for creating GUIs.

The instrumentation control panel is shown in Fig.10. The target speed, which is set by turning the needle of an Angular Gauge Control, the speed, the command and the PWM signal can be analyzed and displayed. In the upper graphic of the panel, the DC motor step response is shown.

Fig.10. Instrumentation control panel.

Using this design approach, there is no need to determine an algorithm or to compute a mathematical model of the motor. The parameters of the controller are determined using experimental methods. The overshoot obtained is under 10%, an acceptable result.

5 Adaptive System Identification and Control of an Automotive Alternator

The electrical power requirements in automobiles have been rising rapidly for many years and are expected to continue to rise [5]. This trend is driven by the replacement of engine-driven loads with electrically-powered versions, and by the introduction of a wide range of new functionality in vehicles. The continuous increase in power requirements is pushing the limits of conventional automotive power generation and control technology [6], [7].

Because of the increasing speed and flexibility of DSP processor, real-time adaptive filtering is becoming an enabling technology for communication, net-work, audio, and control systems.

An automotive alternator is identified using adaptive system identification. Once the coefficients of the digital filter are calculated the mathematical model of the unknown system is determined. A PI control law is implemented in order to control the output voltage of the alternator.

5.1 Adaptive system identification

System identification is an important step to verify the theoretical model with experimental data, since the best theoretical models are only approximations of the real system.

In adaptive system identification, the adaptive filter is connected in parallel with the unknown system (or plant) to be modelled. The modelling signal excites both the unknown system and the adaptive filter. The objective of the adaptive filter is to adapt to the unknown plant. This is achieved by minimizing the error signal which is the difference between the physical response and the modelled response.

The coefficients of the adaptive filter are computed using the LMS algorithm. The Simulink LMS block from the Signal Processing Toolbox is used. The LMS Filter block can implement an adaptive FIR filter using five different algorithms. The block estimates the filter weights, or coefficients, needed to minimize the error between the output signal, and the desired signal.

The FIR filter will estimate the Markov parameters of the process to be identified. The Markov parameters are used to compute the transfer function of the process using the algorithm based on the singular value decomposition of the Hankel matrix. The Hankel matrix is constructed using the Markov parameters [8].

The test setup is shown in Fig.11. The excitation field of the alternator is controlled using a PWM signal generated by the F2812 eZdsp target. The output voltage of the alternator is applied at the desired input port of the LMS Filter Block. The hardware setup at work is shown in Fig.12.
The operational characteristics of the automotive alternator system are shown in Fig.13. The output voltage versus input voltage curves of Fig.13 are calculated for constant speed of the alternator and parameterized by the field current.

The Simulink block diagram used for the adaptive system identification of the automotive alternator is shown in Fig.14.

The computed transfer function, using the coefficients of the adaptive filter, is presented below:

\[
H(s) = \frac{34.97s + 1844}{s^2 + 261.2s + 2046}
\]

5.2 Controller Implementation
Once determined the mathematical model, a PI control law is implemented. A buck converter implements the interface between the eZdsp F2812 target and the automotive alternator.

The converter determines the excitation voltage of the alternator. Its input voltage is 5V and its output voltage, will be directly proportional to the duty ratio of the PWM signal. The PWM signal is applied at the input gate of a MOSFET transistor. The output voltage of the alternator is measured using an ADC channel, and is used to compute the duty ratio of the PWM signal based on a PI control law.

The parameters of the controller where computed using the method introduced by Kessler [9]. The determined controller, using Kessler method, is presented below:

\[
H_c(s) = 17.5 \cdot \left(1 + \frac{1}{0.125s}\right) \cdot \frac{1}{s/52 + 1}
\]

The obtained PI controller has a first order filter, for the cancellation of the zero of the identified process.

The Simulink block diagram used for controlling the output voltage of the automotive alternator is...
presented in Fig.15.

![Simulink block diagram for controlling the output voltage control.](image)

Fig.15. Simulink block diagram for controlling the output voltage control.

The output voltage of the alternator can be controlled using the instrumentation control panel shown in Fig.16.

![Instrumentation control panel](image)

Fig.16. Instrumentation control panel

The target voltage, which is set by turning the needle of an Angular Gauge Control, the output voltage, the command and the PWM signal can be analyzed and displayed. In the upper graphic of the panel, the step response of the alternator is shown.

### 6 Instrumentation, Data acquisition and Results

Until recently, developers were forced to stop their application with a breakpoint to exchange data "snapshots" with the host computer in a technique that is called "stop-mode debugging". This intrusive approach can be misleading, because the isolated snapshot of a halted high-speed application cannot show the real-world operation of the system [10].

To solve this problem, TI developed RTDX, or Real Time Data Exchange, which gives designers continuous, real-time visibility into their applications.

RTDX enables real-time, asynchronous exchange of data between the target and the host, without stopping the target.

### 7 Conclusion

The real-time code for the complete system is automatically generated using Embedded Target and Real-Time Workshop. No hand-coding is required. Time for implementation and testing is minimized.

The graphical programming approach removes the need to write software by hand and allows the engineer to focus instead on improving functionality and performance.

### References:


