Digital Control Applications using TI Digital Signal Controller

PETRU DOBRA, RADU DUMA, DANIEL MOGA, MIRELA TRUSCA
Department of Automatic Control
Technical University of Cluj
St. C. Daicoviciu 15, 400020 Cluj-Napoca
ROMANIA
petru.dobra@aut.utcluj.ro, radu.duma@yahoo.com, daniel.moga@aut.utcluj.ro,
mirela.trusca@aut.utcluj.ro

Abstract: - Two digital control applications are presented: velocity control of a DC motor and adaptive system identification and control of an automotive alternator. A tuning method is proposed for DC motors. Experimental application for the DC motor illustrates the effectiveness and the simplicity of the proposed method for controller design**.** Using adaptive system identification**,** an automotive alternator is identified. A regulator is implemented in order to control its output voltage. The control applications run on a Texas Instruments Digital Signal Controller. Rapid Control Prototyping is used for code generation. Once the desired functionality has been captured and simulated, using the MATLAB/Simulink/Real-Time Workshop environment can be generated code for the DSP. All task assignments to processor are automatically made by the software.

Key-Words: - Embedded Target, DC motor, Digital Signal Controller, Digital Control, Rapid Control Prototyping, Digital Signal Processor, Matlab/Simulink, Code Composer Studio**.**

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). CACSD tools are extensively used to generate real time code automatically. 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.

Digital signal processors (DSP) are a key technology enabling electric drives to be smoother, more efficient, more reliable and most importantly, cheaper. This is being exploited in a whole spectrum of applications such as washing machines, heating, air conditioning and electric power assisted steering in cars. The processing power of DSP controllers is allowing the accurate control of motors with fewer sensors.

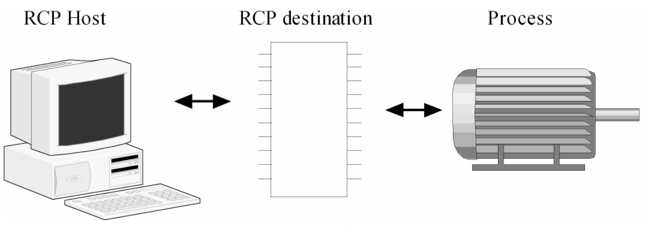

Fig.1. General architecture of an RCP system.

Apart from the basic requirement to control the speed and the direction of the motor, modern motor controllers might also be expected to regulate energy consumption and to implement other functions, ranging from power factor correction to Ethernet drivers and TCP/IP protocol stacks. Depending on application requirements, system complexity ranges from the minimal DC brush motor controller to high end vector drives [2].

DSP application development has changed markedly in the last decade. The lack of efficient C compilers for DSP obliged developers to write algorithms by hand using a low level assembly language - a laborious process. Today is easier: powerful C and C++ compilers are available, which remove the need to code in assembly, speeding the development process [3].

The CACSD tool used in the implementation of the presented applications is Matlab/Simulink/Real-Time Workshop. The target processor is a TMS320F2812 DSP. This processor is highly integrated, high-performance solution for demanding control applications.

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 modeling tool.

Simulink is a platform for multidomain simulation and model-based design of dynamic systems. It provides an interactive graphical environment and a customizable set of block libraries. The engineer can accurately design, simulate, implement, and test control, signal processing, communications, and other time-varying systems.

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 Texas Instruments (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 [4] (Fig.2).

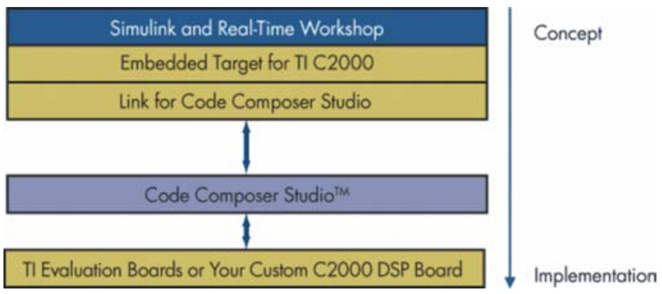
Simulink and Real-Time Workshop
Embedded Target for TI C2000
Link for Code Composer Studio
Code Composer Studio™
TI Evaluation Boards or Your Custom C2000 DSP Board
Concept
Implementation

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. Engineers can automatically generate prototype code for any of the supported boards, combining these blocks with standard blocks from Simulink, Simulink Fixed Point, and the Signal Processing Blockset. User defined blocks, S-functions, can be added to the Simulink model. Thus portions of proven code can be integrated in the model.

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 algorithms run 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 TMS320F2812 device, member of the TMS320C28x DSP generation, is highly integrated, high-performance solutions for demanding control applications.

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 a DSC from TI. Experimental application for the DC motor illustrates the effectiveness and the simplicity of the proposed method for controller design. A PID tuning method for a DC motor is implemented and tested.

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.

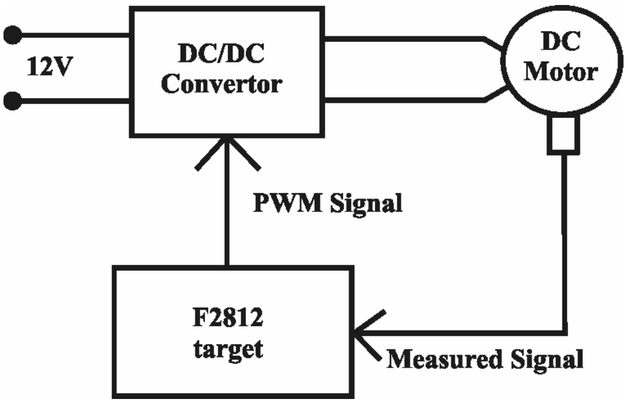

Fig.3. DC motor control setup with F2812.

4.2 Motor Driver

The interface between the F2812 target and the motor is implemented by an electric circuit (Fig.4). 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 Smidth circuit and is applied at the input pin of a capture and log transition unit of the eZdsp F2812 target. Fig.5 presents the hardware setup at work.

4.3 Controller Implementation

Using MATLAB and Simulink for modeling, analysis, design and offline simulation has become a de facto standard for control system development.

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 closed loop control structure is shown in Fig.6 and the Simulink model of the control algorithm in Fig.7. The speed feedback subsystem is presented in Fig.8 and the target speed subsystem in Fig.9.

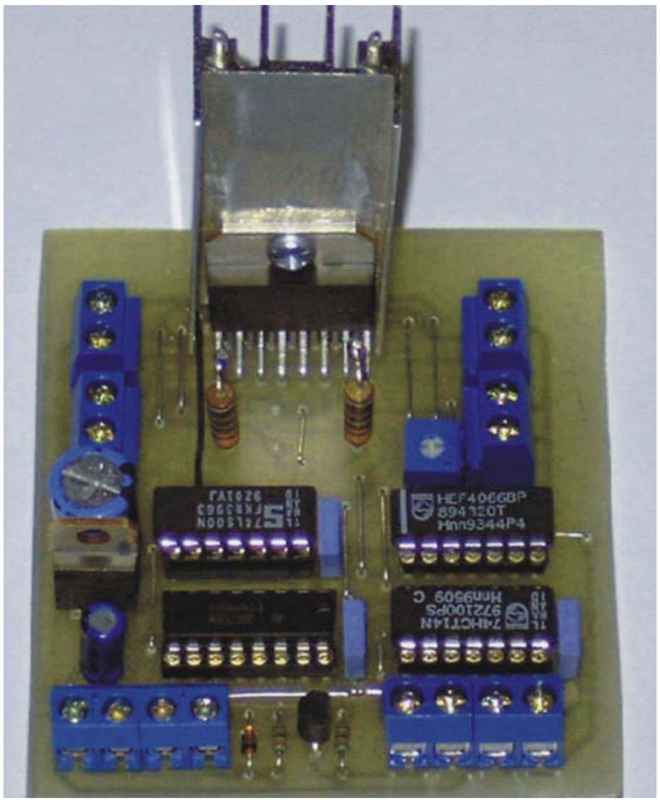

Fig. 4. The motor drive.

Fig.5. The hardware setup at work.

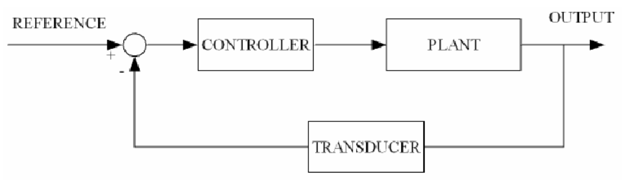

Fig.6. Block diagram of the closed loop control structure

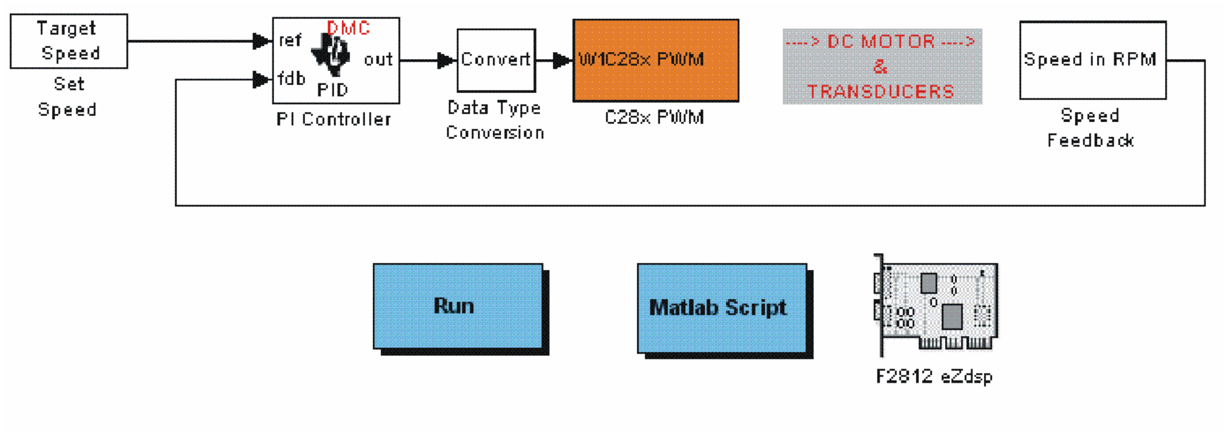

Fig.7. Simulink block diagram of the DC Motor control system.

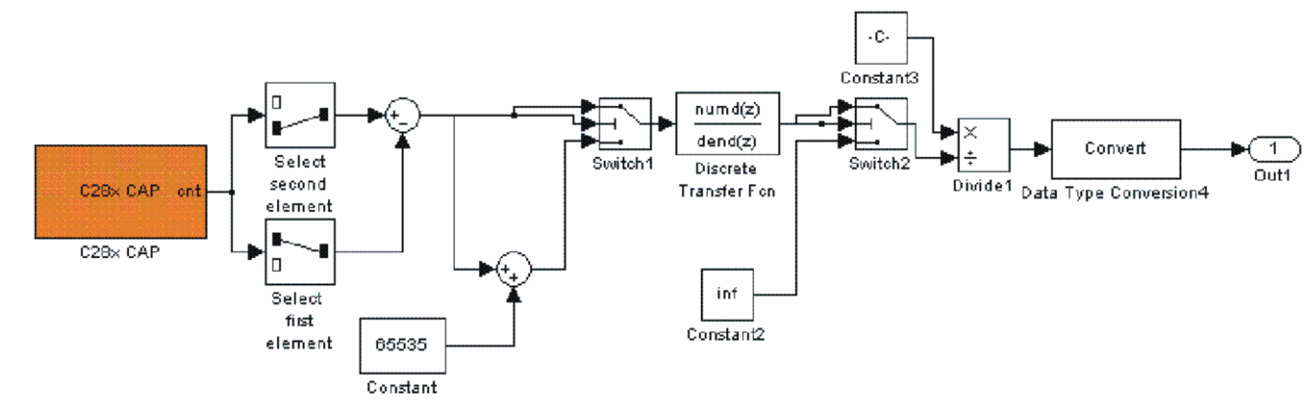

Fig.8. Simulink block diagram for motor speed measurement.

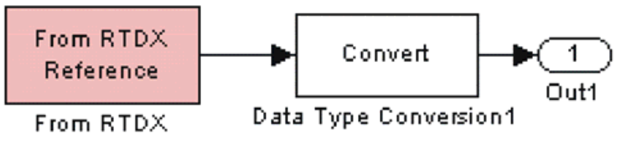

Fig.9. Simulink block diagram for setting the reference speed

A Target Preference block has to be added to the model, in this case the F2812 eZdsp block. It does not connect to any other blocks, but stands alone to set the target preferences for the model. This allows the user to control build options for the compiler, assembler and linker who will be invoked to generate the executable image file for download to the DSP

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. The capture unit logs transitions detected on the capture unit pin by recording the times of these transitions into a two-level-deep FIFO stack.

At multiples of the sample time the values from the two-level-deep FIFO stack are subtracted. For a good resolution the 75 MHz input clock, associated with the capture unit, is divided by a factor of 1/128. By this factor the general purpose timer is prescaled to produce the counting rate of 1.70752.

The counter associated with the capture unit is on 16 bits. If it reaches the upper limit of 2^{16}-1 it will reset to 0. For a motor speed between 0-4000 RPM the measured signal is in the range 80 – 360 Hz.

In order to reduce signal jitter or period fluctuation a first order delay filter is used. The time constant of this filter is 0.3 seconds. The output of the filter is applied at the feedback input port of the PID Controller block, from the Digital Motor Control Library (DMC).

At the reference input port of the controller is applied the target speed using a RTDX block. The drive can be controlled using a graphical user interface (GUI), created using GUIDE, the MATLAB Graphical User Interface development environment.

The controller computes the duty ratio of the PWM control signal generated using a Simulink/Embedded Target block. The frequency of the PWM signal is fixed at 5 KHz, but its duty ratio changes based on a PID control law.

4.4 Controller Tuning

A key method for auto-tuning is to use the relay feedback method [5]. Processes with the dynamics typically encountered in process control will exhibit limit cycle oscillations. The relay feedback causes the process to oscillate with controlled amplitude. The frequency of the limit cycle is approximately the ultimate frequency where the process has a phase lag of 180°. The ratio of the amplitude of limit cycle and the relay amplitude is approximately the process gain at that frequency. Thus a point on the Nyquist curve of the open loop dynamics close to the ultimate point is determined.

The parameters of the controller will be computed by specifying the phase and amplitude margin [6]. 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.10). The Simulink block diagram for obtaining the limit cycle is shown in Fig.11.

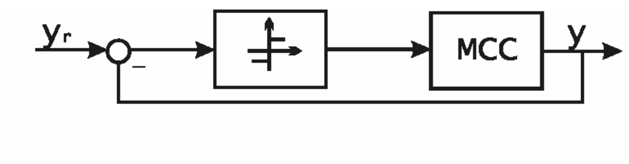

Fig.10. Relay feedback diagram.

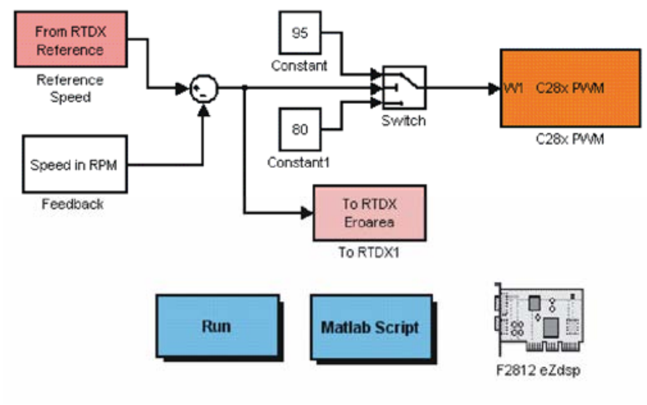

Fig.11. Simulink block diagram of Ziegler – Nichols ultimate period method.

The obtained limit cycle is shown in Fig.12.

From Fig.12 the critical period T_u is determined. The ultimate gain K_u is computed using relation:

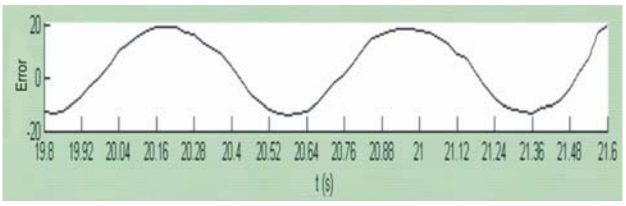

Fig.12. Limit cycle.

$$K_u = \frac{4d}{\pi a} \tag{1}$$

The open loop transfer function with PID control is:

$$k_p(1 - \frac{1}{j\omega T_i} + j\omega T_d)H_{DCM}. \tag{2}$$

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

$$arctg(\omega T_d - \frac{1}{\omega T_i}) = \gamma_m. \tag{3}$$

Solving equation (3), for $T_i = \alpha T_d$ with $\alpha \in (2 \div 6)$, is obtained the derivative time constant:

$$T_d = \frac{1}{2\omega_0}(\tan\gamma_m + \sqrt{4/\alpha + \tan^2\gamma_m}). \tag{4}$$

If the magnitude of the open loop transfer function is specified to be k_m simple trigonometric calculations give:

$$k_p = k_m K_u \cos\gamma_m \tag{5}$$

The step response, using the computed PID controller, is shown in Fig. 13.

In order to obtain the step reference input signal, the Simulink diagram shown in Fig. 14 is used.

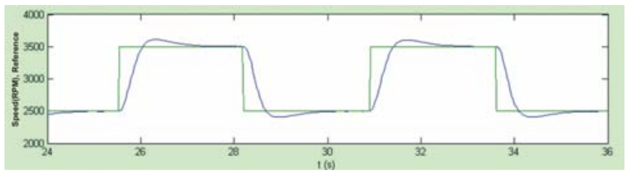

Fig 13. DC motor step response.

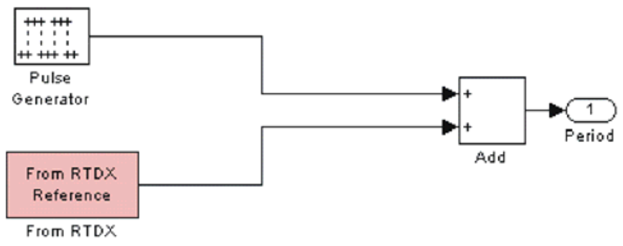

Fig 14. : Simulink block diagram for obtaining the step input signal

4.4 Results

The drive can be controlled using a GUI, created using GUIDE. RTDX blocks are used to fetch and retrieve data from the target.

The instrumentation control panel is shown in Fig.15. 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, and in the lower one the duty ratio of the PWM signal in percentage.

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

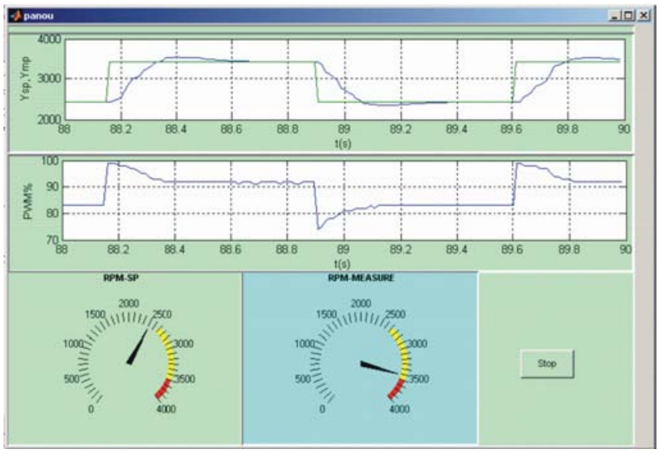

Fig.15. Instrumentation control panel.

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 [7]. 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 [8], [9].

The coefficients of adaptive filters continuously and automatically adapt to given signal in order to achieve the desired response. Adaptive filtering can be used in system identification, inverse system modelling, noise cancellation, signal prediction, and many others.

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

The coefficients of a digital filter determine the characteristics of the filter. In many practical application filter specification are unknown at the design time and/or change with time. In addition, it may be necessary to attenuate noise that has spectral overlap with the desired signals. For this applications have to be used adaptive filters with time-varying coefficients updated by adaptive algorithms to track the unknown and/or changing environments.

As shown in Fig.16, an adaptive filter consists of two main functional units: a digital filter with time-varying coefficients to perform desired filtering and an adaptive algorithm to adapt the coefficients of the filter in order to improve performance. The filter structure is determined and fixed at the design stage, but the adaptive algorithm continuously adjusts its coefficients.

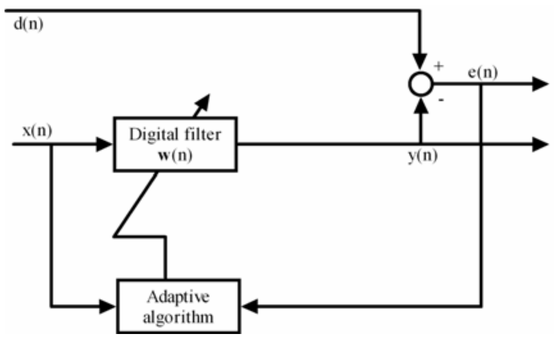

Fig.16: Block diagram of an adaptive filter.

The generic adaptive filtering given in Fig.16 shows that the input signal, $x(n)$, is filtered by the digital filter to produce the output signal, $y(n)$. The adaptive algorithm adjusts the coefficients of the filter contained in the vector $w(n)$ to minimize the error signal, $e(n)$, which is the difference between the desired signal, $d(n)$, and the filter output, $y(n)$. Therefore the adaptive-filter design is based on the characteristics of the input signal $x(n)$ and the signal $d(n)$. In this way, the adaptive filter adapts to the environment described by these signals. When the environment changes, the filter adapts to the new characteristics by generating a new set of coefficients.

The digital filter shown in Fig.16 can be an FIR, symmetric FIR, IIR, or other structure. The coefficients of an adaptive filter are time functions because the adaptive algorithm continuously updates them either sample by sample or block by block.

To design an adaptive filter the correct filter structure has to be selected. For the selected filter structure many adaptive algorithms can be used to update the filter coefficients. The complexity of an adaptive algorithm is usually measured in terms of its multiplication and memory requirements [10].

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.

Non parametric system identification methods determine the time and the frequency response of a liner time invariant system without prior knowledge of the model structure of a system. They are often used as a preliminary tool to get initial estimate of the model structure of a system with unknown dynamics.

Adaptive system identification is illustrated in Fig.17. The adaptive filter is connected in parallel with the unknown system (or plant) to be modelled. The modelling signal $x(n)$ excites both the unknown system and the adaptive filter. The objectives of the adaptive filter is to adapt to the unknown plant; thus, the adaptive filter output, $y(n)$, closely matches the unknown system output, $d(n)$. This is achieved by minimizing the error signal, $e(n)$, which is the difference between the physical response $d(n)$ and the modelled response $y(n)$. If the excitation signal, $x(n)$, is rich in frequency contents such as white noise and the internal plant is small, the adaptive filter converges to the unknown system from I/O viewpoint.

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 estimates 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 [11].

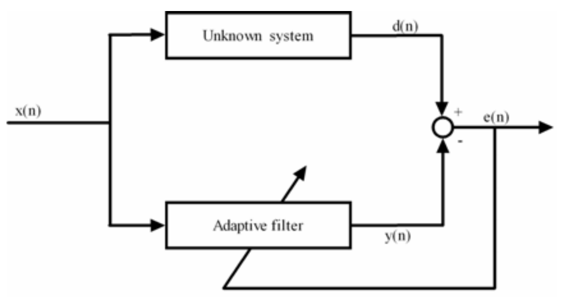

Fig.17. System identification using adaptive filters.

The block diagram of the test setup is shown in Fig.18. 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.

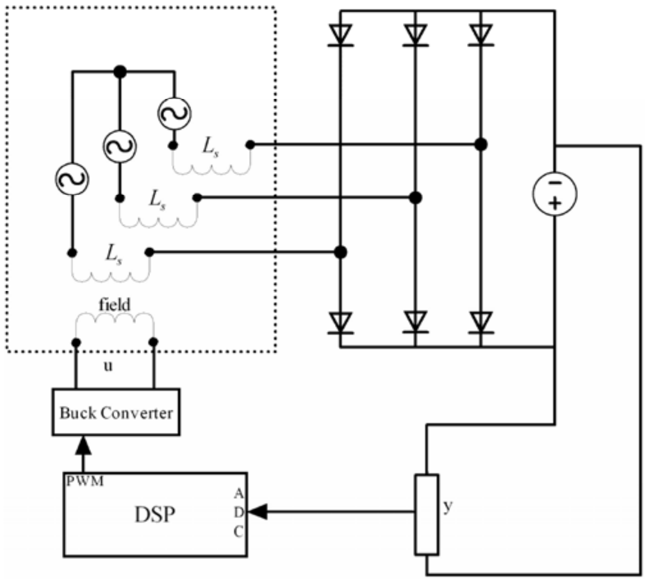

Fig.18. Adaptive system identification block diagram setup using F2812 eZdsp target.

The automotive alternator is driven by an asynchronous motor. The motor is controlled using an inverter. The hardware setup at work is shown in Fig.19.

The operational characteristics of the automotive alternator system are shown in Fig.20. The output voltage versus input voltage curves of Fig.20 are calculated for constant speed of the alternator and parameterized by the field current.

Two ADC channels of the eZdsp target are used to acquire the field and the output voltage. The results are shown in Fig.21.

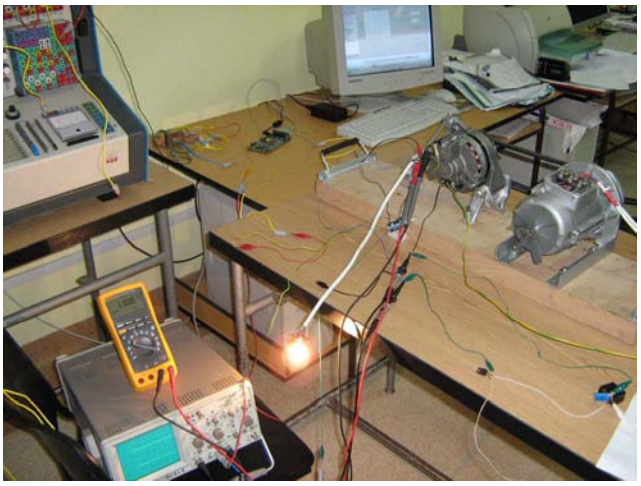
Fig.19. Hardware setup at work.

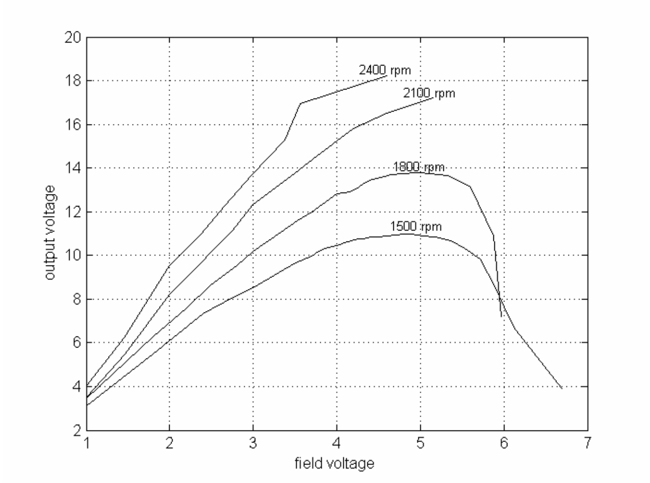

Fig.20. Operational characteristics of the alternator.

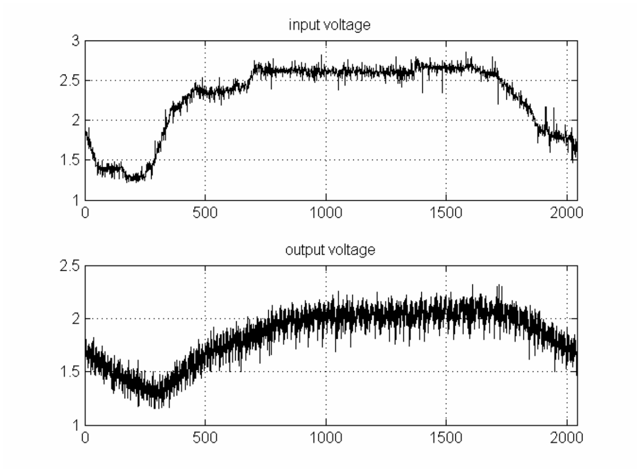

Fig.21. The input and output voltage of the alternator system.

The Simulink block diagram used for the adaptive system identification of the automotive alternator is shown in Fig.22. The field voltage and the output voltage of the alternator are applied at the input ports of Simulink LMS block. The block computes the coefficients of the digital filter, which are fetched from the target board using a RTDX block.

The pseudocode algorithm for computing the automotive alternator transfer function, using the coefficients of the adaptive filter is presented below:

- *estimate Markov parametres using adaptive system identification;*
- *construct the Hankel Matrix using the estimated Markov parameters;*
- *decompose into singular values the Hankel matrix;*
- *using the singular values decomposition matrices, compute the state space matrices of the process to be modeled;*
- *convert the state space model to continuous transfer function;*

The continuous form of the discrete transfer function 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 (Fig.23) implements the interface between the eZdsp F2812 target and the automotive alternator.

The converter determines the excitation voltage of the alternator. Its input voltage, V_i, is 5V and its output voltage, V_o, is directly proportional to the duty ratio (μ) of the PWM signal ($V_i = \mu Vo$). 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 [12]. The determined controller, using Kessler method, is presented below:

$$H_c(s) = 17.5 * (1 + \frac{1}{0.125s}) * \frac{1}{s/52+1}.$$

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.24.

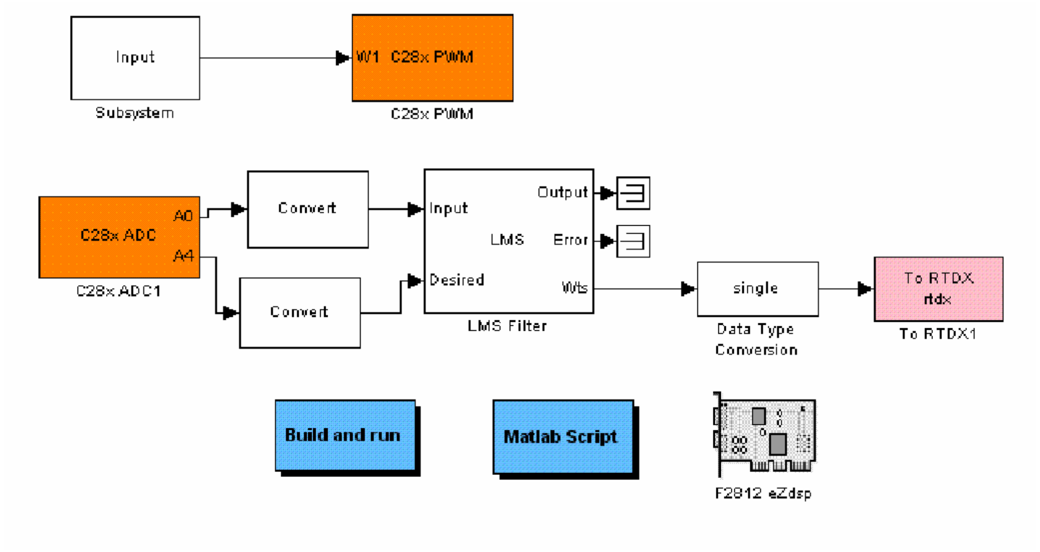

Fig.22. Simulink block diagram for adaptive identification.

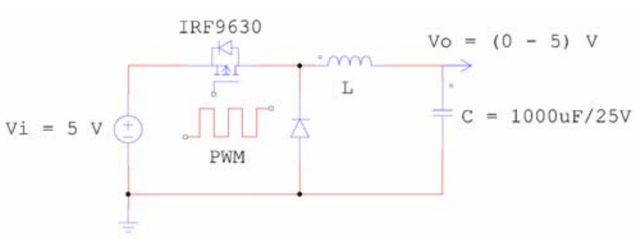

Fig.23 : The driver for the automotive alternator

5.3 Results

The output voltage of the alternator can be controlled using the instrumentation control panel shown is Fig.25.

A rectangular reference input signal is applied at the reference input port of the PID controller. 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, and in the lower one, the duty ratio of the PWM signal in percentage.

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 [13].

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

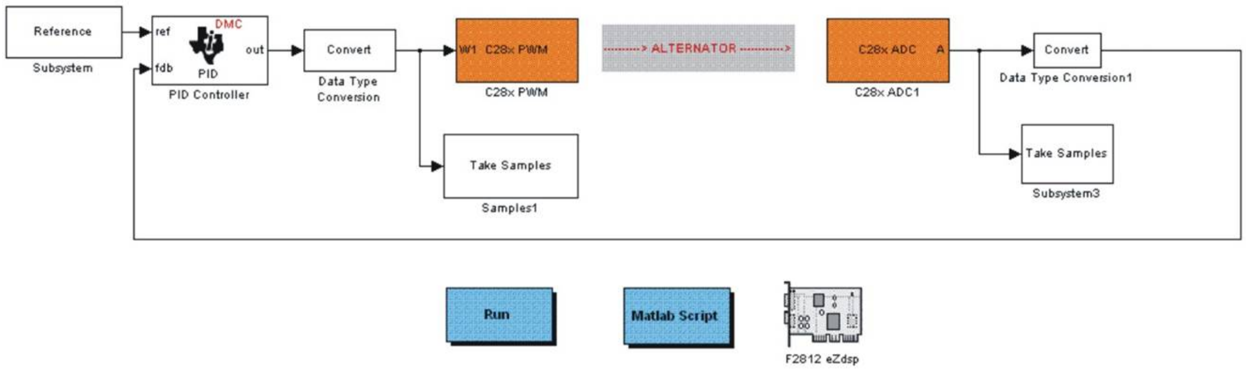

Fig.24. Simulink block diagram for controlling the output voltage control.

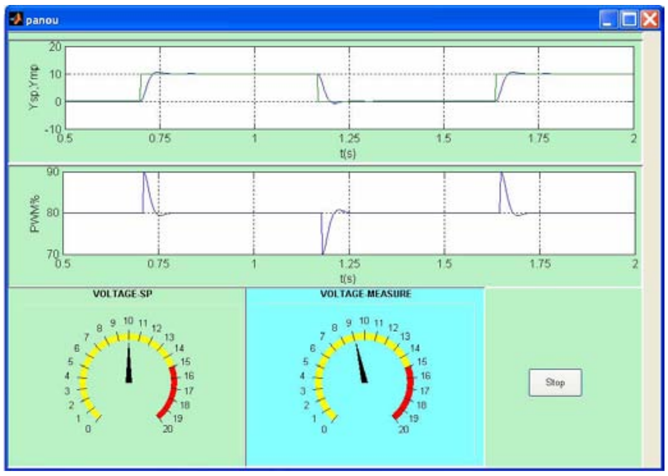

Fig.25. Instrumentation control panel

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

In essence, the RTDX data link provides a "data pipe" between DSP application and the host. The bidirectional capability allows developers to access data from the application for real-time visibility, or to simulate data input to the DSP, perhaps before real-time sensor hardware is available. This shortens development time by giving developers a realistic view of the way their systems operate.

8 Conclusion

The paper presented two rapid control prototyping applications: velocity control of a DC motor and adaptive system identification and control of an automotive alternator

The coefficients of the DC motor were computed using an experimental method. The obtained overshot is under 10%, an acceptable result.

As an extension of the automotive digital control application we intend to implement a DSC based battery charging system.

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.

This design approach removes the need to write software by hand and allows the engineer to focus instead on improving functionality and performance.

References:

[1] H. Hanselmann, Control: the total development environment, *International Conference on Signal Processing Applications*, Boston, MA, 1995.

[2] K. Godbole, O. Monnier, and J. Skinner Gray , “Modern motor controllers must do more than just spin the motor”, http://www.neon.co. uk/ campus/articles/texas/ ti_articles_spring04.cfm

[3] R. Poley, “Blocks bring benefits”, [Online] Available:http://www.neon.co.uk/campus/article s/texas/ti_articles_autumn05.cfm

[4] *****, www.mathwork.com.

[5] J. Ziegler and N. Nichols, Optimum settings for automatic controllers, *Trans. ASME*, pp. 759–768, 1942.

[6] W. K. Ho, C. C. Hang and L. S. Cao, Tuning of PID controllers based on gain and phase margin specifications, *Automatica*, VOL. 31, NO. 3, ,pp. 497-502, 1995.

[7] D. J. Perreault and V. Caliskan , Automotive Power Generation and Control, *IEEE TRANSACTIONS ON POWER ELECTRONICS*, VOL. 19, NO. 3, 2004.

[8] J. M. Miller, Multiple voltage electrical power distribution system for automotive applications, *Proc. 31st Intersociety Energy Conversion Engineering Conference (IECEC)*, 1996.

[9] J. M. Miller, D. Goel, D. Kaminski, H.-P. Schoner and T. M. Jahns, Making the case for a next generation electrical system, *Proc. IEEE-SAE International Conference on Transportation Electronics (Convergence),* 1998

[10] S. M. Kuo, and W.-S. Can. *Digital Signal Processors Architectures, Implementations and Applications*. Pearson Prentice Hall, New Jersey, 2005.

[11] L. Ljung, *System Identification- Theory For the User,*. PTR Prentice Hall, Upper Saddle River, N.J., 1999.

[12]C. Kessler, Das symmetriche optimum, *Regelungstechnik*, 6, 395-400 and 432-436, 1958.

[13] *****, www.ti.com.

[14]S. Sp. Pappas, Robust High Performance Servo Controller Design Technique using Matlab/Simulink, *WSEAS TRANSACTIONS on SYSTEMS and CONTROL*, Issue 2, Volume 1, December 2006.

[15] W. K. Ho, K. W. Lim, and W Xu, Optimal gain and phase margin tuning for PID controllers, *Automatica*, VOL. 34, No. 8, pp. 1009-1014, 1998.

[16] W. K. Ho, Y. Hong, A. Hanson, H. Hjalmarson and J. W. Deng, Relay auto-tuning of PID controllers using interative feedback method tuning, *Automatica*, VOL. 39, pp. 149-157, 2003.

[17]T. Sekozawa, Model-based Control and Learning Control Method for Automobile Idle Speed Control using Electric Throttle, *WSEAS TRANSACTIONS on SYSTEMS and CONTROL*, Issue 2, Volume 3, February 2008.

[18] A. Leva and F. Schivo, Robust autotunning of industrial regulators based on complex process models: the DIMC approach, *Proc. IEEE Conference on Conference of Computer Aided Control System Design*, Munich, 2006.

[19] A. A. Voda and L. D. Landau, A method for auto-calibration of PID controllers, *Automatica*, VOL. 31, pp. 41-53, 1995.

[20] X. Yuan, Y. Wang, L. Wu, Adaptive Inverse Control of Excitation System with Actuator Uncertainty, *WSEAS TRANSACTIONS on SYSTEMS and CONTROL*, Issue 8, Volume 2, August 2007.