Sborník vědeckých prací Vysoké školy báňské - Technické univerzity Ostrava
číslo 2, rok 2006, ročník LII, řada strojní
článek č. 1528

Petr DOLEŽEL[*], Vladimír VAŠEK[**]

CONTROL ALGORITHMS FOR MICROCONTROLLERS

ŘÍDICÍ ALGORITMY PRO MIKROPOČÍTAČE

Abstract

This project was focused on Motorola microcontrollers. The aim of this project was to create library, which can make programming process easier. This library is composed of modules that can be implemented to the final program. The programming process is then based on joining different modules together. Modules are compatible among all Motorola microcontroller; however, one module is not. This special module operates with microcontroller's peripheries. This module has to be modified according to microcontroller's specification. It was impossible to create absolutely universal module because each microcontroller has different functions and the direct access to the memory is provided via different registers. Modules for continuous identification, supportive modules and adaptive control were created and tested on real systems.

Abstrakt

Tento projekt je zaměřen na mikropočítače Motorola. Cílem bylo vytvořit knihovnu programových modulů, které mají ulehčit programovací proces. Moduly lze snadno implementovat do výsledného programu. Všechny moduly kromě jednoho jsou kompatibilní s různými řadami mikropočítačů Motorola. Modul, který kompatibilní není, obsahuje funkce, které přímo ovládají periferie mikropočítače a je tedy nutné tento modul upravit dle specifikací jednotlivých mikropočítačů. V tomto případě nebylo možné vytvořit univerzální modul, protože různé mikropočítače mají různé funkce a přístup do registrů je proto u každého mikropočítače odlišný. V tomto projektu byly vytvořeny moduly pro adaptivní řízení, průběžnou identifikaci nebo moduly s podpůrnými funkcemi.

1 INTRODUCTION

The modern methods of automation theory are used widely in technological processes nowadays. In this case the microcontrollers are commonly used for this purpose. This brings many advantages such is cheap costs, small size, all needed peripheries integrated directly on chip etc. However, there are some disadvantages [1]. The programming process is quite sophisticated. There are two typical ways to create program. The first one is the low level programming language such is assembler. The second one is via special programming software commonly based on C language.

2 METHODS

This work is connected with the previous project that was focused on Motorola microcontroller M68HC11. Previous modules were created in assembler [2]. The disadvantage is that these modules might be used only in this microcontroller family. They are not compatible with the other families especially with HC08.

[*] Ing., Department of Automation and Control Technology, Faculty of Applied Informatics, Tomas Bata University, Nad Stráněmi 4511, 760 05 Zlín, tel. (+420) 576 035 133, e-mail dolezel@fai.utb.cz
[**] Prof. Ing. CSc, Department of Automation and Control Technology, Faculty of Applied Informatics, Tomas Bata University, Nad Stráněmi 4511, 760 05 Zlín, tel. (+420) 576 035 220, e-mail vasek@fai.utb.cz

One of the aims was to provide compatibility among different types of microcontrollers. CodeWarrior was used for creating new modules. This program was chosen because of its advantages [1]. This programming tool is based on C language that makes these modules compatible among different microcontroller's families. CodeWarrior is able to work with floating point numbers. It offers *float* and *double*. These floating point numbers are in standard IEEE format in 4 (*float*) or 8 (*double*) byte size. There are only *float* type numbers in modules because the functions for basic operations are much faster and need smaller memory space then functions for *double* type.

All new modules are created only by this tool. However, there is one module that is not absolutely compatible. This module operates with microcontrollers peripheries such is AD converter, timer, interrupts etc. These properties are different in each microcontroller, so that the access has to be done by different registers. The library contains one example of this module for one microcontroller MC9S12E128. This module can be simply modified according to microcontroller's specification.

3 IMPLEMENTATION

3.1 Special modules

The special module has many functions for control microcontroller's peripheries. It depends on user and controlled system which functions will be chosen. The analog-to-digital converters were used for measuring the process value, real time or timers were used for sample timing and serial communication was used for communication with superior system (PC).

There are also functions for interrupts from timer or real time in this module. These functions provide sample time or slower PWM. The optional parameter is the sample time or in case of PWM the band wide. The minimal value is 0.5 second.

Controlled system can be very fast and in this case the best choice is to use internal PWM for control this system. Internal PWM can be found in several microcontrollers; however, this function is not standard. This internal PWM is very fast. It depends on clock source (bus clock) and other properties defined by user in registers of PWM. It usually operates in MHz. This module also offers software-generated PWM suitable for slower systems or for microcontrollers with no internal PWM.

Many of the control algorithms work with matrices or theirs parameters are calculated from the system of linear equations. For this case the module for standard operations with matrices and module for calculating with systems of linear equations was created.

3.2 System identification

The continuous identification was used for system identification [3]. This module is able to identify the system according the following equation (1).

$$G\left(z^{-1}\right)=\frac{B\left(z^{-1}\right)}{A\left(z^{-1}\right)}=\frac{b_1 z^{-1}+b_2 z^{-2}+\ldots+b_{nb} z^{-nb}}{1+a_1 z^{-1}+a_2 z^{-2}+\ldots+a_{na} z^{-na}} \tag{1}$$

The data vector and vector of estimates can be seen in equation (2) and (3):

$$\phi^T{}_{k-1}=\left[-y_{k-1},-y_{k-2},\ldots,-y_{k-na},u_{k-1},u_{k-2},\ldots,u_{k-nb}\right] \tag{2}$$

$$\hat{\Theta}^T(k)=\left\lfloor \hat{a}_1,\hat{a}_2,\ldots,\hat{a}_{na},\hat{b}_1,\hat{b}_2,\ldots,\hat{b}_{nb}\right\rfloor \tag{3}$$

This module can be used also for delta representation but in this case the order of controlled system cannot be higher than two. Systems examined in this project were identified as systems of the first order (4) or second order (5):

$$G\left(z^{-1}\right)=\frac{b_1 z^{-1}}{1+a_1 z^{-1}} \tag{4}$$

$$G(z^{-1}) = \frac{b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \tag{5}$$

3.3 Ziegler-Nichols method

Ziegler-Nichols method is one of the basic control algorithms. The module can control system of the second order. For calculation of the control value the standard Takahashi algorithm was used according to [4]. This module contains only one function with no optional parameters.

3.4 Pole-placement method

Next module contains four functions for pole-placement method. Control algorithms for PID-A (**Fig. 1**) and PID-B (**Fig. 2**) can be set according to required process control. Each function has two optional parameters. These parameters influence the control process.

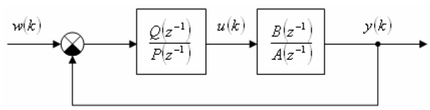

Fig. 1 Block scheme of PID-A algorithm

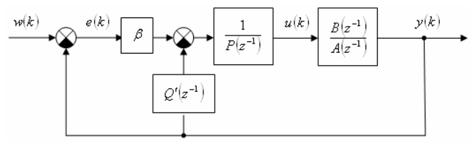

Fig. 2 Block scheme of PID-B algorithm

There are two functions for each block scheme of the system. These functions get regulator parameters from different equations that represent the pole placement. The placement is set via polynomial D. The first type (PID-A1 and PID-B1) can be calculated from polynomial (6):

$$D(z^{-1}) = 1 + d_1 z^{-1} + d_2 z^{-2} \tag{6}$$

The second type (PID-A2 and PID-B2) of regulator has parameters calculated according to the following polynomial:

$$D(z^{-1}) = (z - \alpha)^2 [z - (\alpha + j\omega)] [z - (\alpha - j\omega)] \tag{7}$$

3.5 Adaptive algorithms

Adaptive regulators are implemented into two modules. Both of them contain two different approaches to regulating system. There are two different block schemes of the system – feedback regulator is shown in **Fig. 3** (FB)

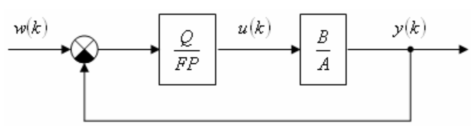

Fig. 3 Feedback regulator

and feedback regulator with feed forward loop as can be seen in **Fig. 4** (FBFW):

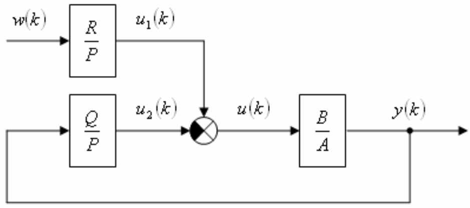

Fig. 4 Feedback regulator with feed forward loop

The synthesis of FB regulator is based on solution of diophantic equation:

$$AFP + BQ = D \tag{8}$$

There are two diophantic equations for FBFW regulator:

$$\begin{aligned} AP + BQ &= D \\ FT + BR &= D \end{aligned} \tag{9}$$

The polynomial D can be set for three different control performances:

1. Dead-beat (DB)
2. Quadratic optimal control (LQ)
3. Pole-placement method (PP)

The first module contains functions for first order systems and the second one for the systems of second order. All functions have optional parameters which set required control performance.

4 RESULTS

All created modules were tested on real system. The simple heat system was used for this purpose. Measurements were taken on the same system with the same setpoints. The initial value was set to 60°C. After 300 seconds was switched to 80°C. The last value was again 60°C. Systems were identified by continuous identification module.

The Ziegler-Nichols method with sample time 10 seconds can be seen in **Fig. 5**. Module for this regulator has no setting parameters.

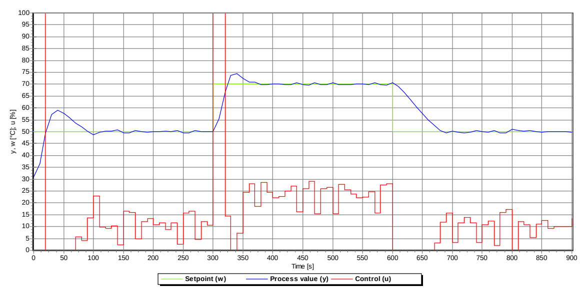

Fig. 5 Ziegler Nichols method

Next figure (Fig. 6) shows progression of parameters of the system that was identified in the previous examination.

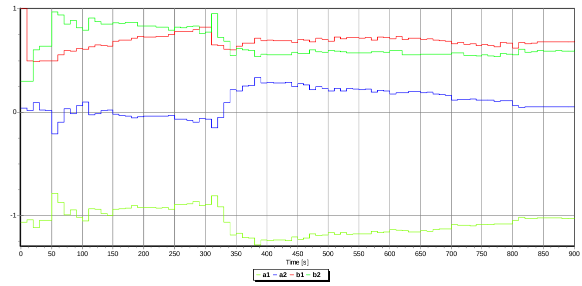

Fig. 6 Parameters of identified system

Pole-placement method was used next. An example of this method can be seen in Fig. 7. Sample time is 10 seconds. Figure 8 shows an adaptive feed-back regulator with LQ method.

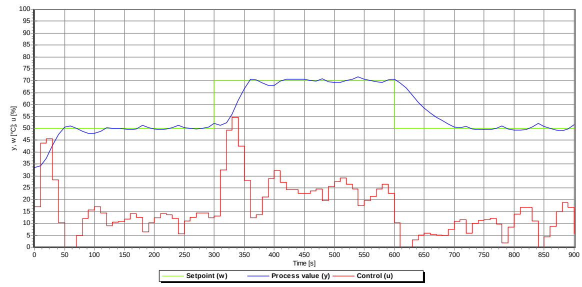

Fig. 7 Poleplacement method PID–B1

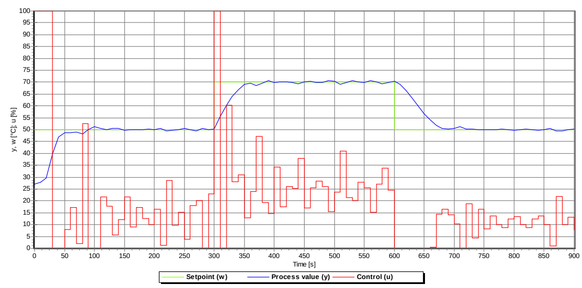

Fig. 8 LQ adaptive FB regulator

Next figure (**Fig. 9**) shows adaptive regulator also. But in this example the feed-back feed-forward system was used. Poleplacement method was used for setting the parameters of regulator. Function parameters were set to 0.5 and 1.

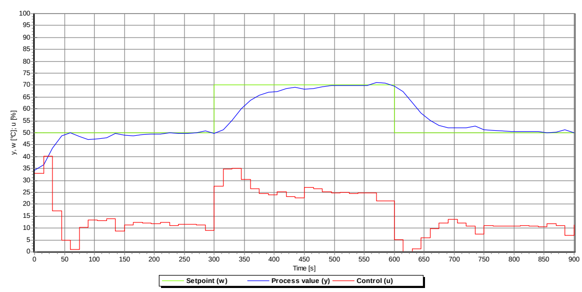

Fig. 9 PP adaptive FBFW regulator

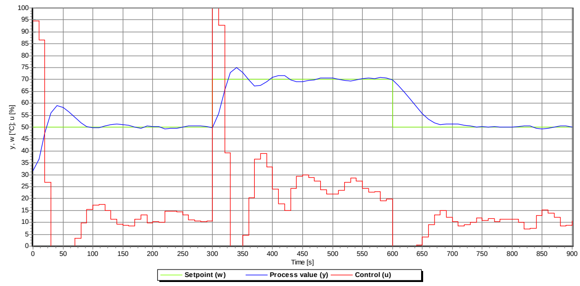

Fig. 10 DB adaptive regulator

The controlled system identified as a system of the first order can be seen in **Fig. 10**.

5 CONCLUSIONS

Application of adaptive regulators in microcontrollers is simple with such library. CodeWarrior is very good tool for creating compatible modules. The implementation of all control modules can be done by including them simply to the project via files with prototypes of functions.

The whole project is far from complete because many different algorithms are continuously developed. The library can grow by adding new modules that can implement many other algorithms such is module for delta representations or for controlling multidimensional systems.

ACKNOWLEDGEMENT

This work was supported by research project 708 352 102 MSM. This support is gratefully acknowledged.

REFERENCES

[1] MORTON, D. TODD. *Embedded Microcon-trollers.* Prentice Hall, Upper Saddle River 2001. ISBN 0-13-907577-1

[2] DOLINAY, J. 2002. *Diploma work – Programové moduly řídicích a identifikačních algoritmů pro mikropočítač 68HC11.* Zlín 2002.

[3] BOBÁL, V., CHALUPA, P. 2003. *Self-Tuning Controllers Simulink Library*. Zlín 2003. Available in: http://www.utb.cz/stctool/

[4] BOBÁL, V., BÖHM, J., PROKOP, R., FESSL, J. 1999. *Praktické aspekty samočinně se nastavujících regulátorů: algoritmy a implementace.* Brno 1999. ISBN 80-214-1299-2

Reviewer: prof. Dr. RNDr. Lubomír Smutný, VŠB-Technical University of Ostrava