JOURNAL OF THEORETICAL
AND APPLIED MECHANICS
3, 38, 2000

RAPID PROTOTYPING OF MECHATRONIC SYSTEMS

Tadeusz Uhl*
Tomasz Bojko*
Zbigniew Mrozek**
Wojciech Szwabowski*

** Department of Robotics and Machine Dynamics, University of Mining and Metallurgy, Cracow*
*** Department of Computer Research, Cracow University of Technology*
e-mail: tuhl@rob.wibro.agh.edu.pl

The paper describes a mechatronic approach towards designing and implementation complex motion control systems. Particular attention was paid to virtual and physical prototyping of the control system. Possible implementations were divided into two groups: single applications or mass production. Authors attempt to present a unified approach towards the development of the process of designing control systems of both types. The prototyping environment was build around MATLAB, Simulink and RTW software from The MathWorks Inc.
Verification of presented methodology was done using physical model of a flexible robot arm: a with flexible beam, DC motor drive and harmonic gear. Object is very difficult to control due to non-linear and non-stationary behaviour of harmonic gear.

Key words: mechatronics, rapid prototyping, flexible arm, control

1. Introduction

Current tendencies in the development of machines and equipment design lead to the mechatronics – methodology of interdisciplinary projects, where all the elements forming a structure are treated as equally important irrespective of their physical nature. Computer-aided design (CAD) tools are used in the development process. They allow for multicriterional optimisation of a structure, but also reduce the time necessary to create a new product.

The main advantage of rapid prototyping is unification of design methodology and reusability of equipment and software used in development. Other profits are: better quality of final product and shortage of time period from initial design to product on the market.

2. Mechatronic approach

The paper describes a mechatronic approach towards the design of control systems. Auslander and Kempf (1996) defined mechatronics as the system design that "has the goals of producing systems with superior performance while minimising time to market and manufacturing costs, and maximising reliability".

The term *mechatronics* is now widely accepted world-wide as a design methodology which integrates (from the earliest stages of design) mechanical, electrical and electronic engineering together with information technology, within a wide range of products and processes. In other words mechatronics is a synergetic combination of mechanical engineering, electronic, control and informatics in design production and processes.

It means, that parts of a different nature will cooperate in one machine: mechanical, electrical, electronics and software. To make cooperetion of this parts possible and effective – one should take integration and multicriterional optimisation into account at each stage of the development process. Otherwise the parts of different nature will not cooperate properly and effect of synergy can be lost.

An example of mechatronic system with several subsystems is presented in Fig.1. There are the following subsystems:

- sensors, data acquisition and pre-processing one
- actuators (motors, drives, gears, etc.)
- computer hardware (microprocessor, DSP, embedded processor, ASIC, etc.) and software
- mechanical one.

Prototyping of mechatronic system is a very complex process. Application of computer system to management and design is strongly recommended. A very fast progress in computer hardware and software and much more computer power available at the same costs make it possible to support the development at each stage of prototyping of mechatronic systems.

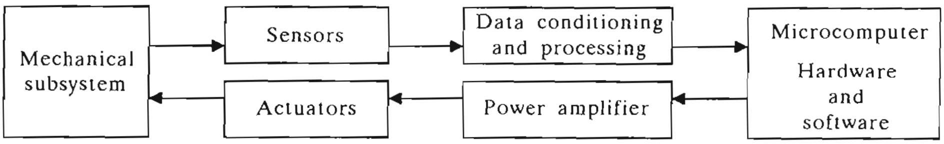

Fig. 1. An example of mechatronic system structure

2.1. Computer-Aided Design (CAD) of mechanical system

There are many software packages supporting the design process of mechatronic systems. The most advanced and popular are:

- CATIA (Dassoult)
- IDEAS (SDRC Inc.)
- INTERGRAPH (Microstation)
- ProEngineer
- Mechanical Desktop (Autodesk).

The first (most advanced) and the last (most popular) ones are used in our laboratory. CATIA is very expensive and requires powerful computer systems and well trained staff. It is used mainly in aviation and automotive industry due to high costs of its purchasing and exploitation. AutoCAD (with Mechanical Desktop extension) is not a professional CAD tool, but is very useful in preparing less complex design. AutoCAD is very popular in small and medium enterprises in Poland.

Simulation is the most important methodology used in supporting of rapid prototyping. One can simulate movement of mechanical parts represented by its mathematical model, external and internal forces, energy, etc, taking into account all constraints.

Movement of complex mechanical systems can be simulated using the packages: ADAMS, DYMAC, DADS, TREETOPS, COMPAN, NEWEUL, MEDYNA. Those packages are very powerful, computational results are visualised on a computer screen and animated. The main drawback is the lack of needed interfaces: internal model of mechanical system cannot be exported from one package to the other one, used at subsequent stages of the design process.

There are several packages (MAPLE V, MATHEMATICA, MAKSYMA, DERIVE, REDUCE, AXIOM) used for symbolic transformation and processing of mathematical formulas. Even a very large and complex set of equations can be processed. And, what is very important, one can modify any part of mathematical expression, process all new equations with a symbolic package and obtain almost immediately the new set of equations in a standard form of mathematical model. The above packages are very useful as they may generate model equations in the form of computer program in C language. This C language program can be used for simulation on the subsequent stages of design process. Special attention should be given to DYMOLA package. It can be used for modelling of systems of a different nature; e.g., mechanical, electrical, thermodynamical, chemical, etc. It is modern and allows for object oriented programming.

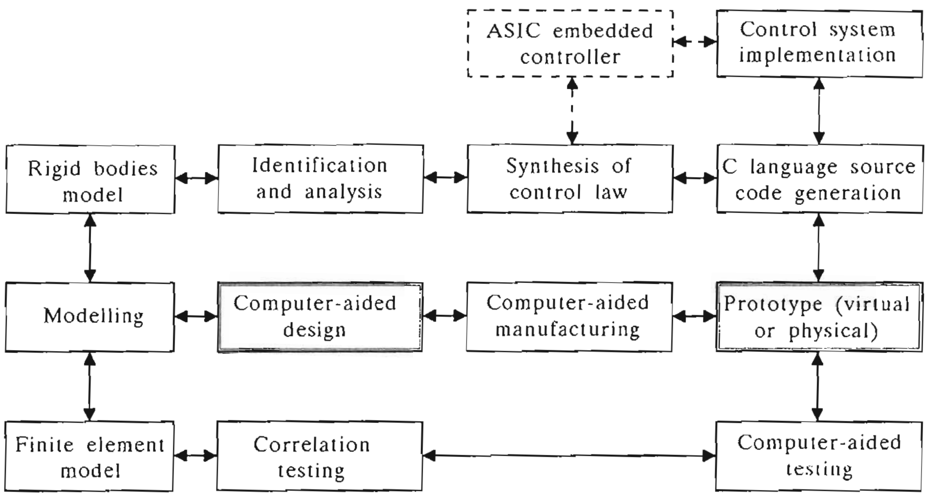

Fig. 2. Customised structure of computer-aided design methodology used in the Mechatronic Laboratory of University of Mining and Metallurgy

A mathematical model of the system, formulated using any of the above packages or prepared manually (in the form of Simulink or $MATRIX_X$ block diagram), can be used for simulation in one of the following packages: ACSL, SIMNON, DESIRE, MATLAB/Simulink, $MATRIX_X$. As the model is accepted as a set of equations or computer program, the real system can be of any physical nature, even non existing in reality (very important class of virtual systems). The mathematical model is very useful in rapid prototyping, as it may integrate knowledge of all disciplines involved in the design process. Integration of knowledge and experience from many disciplines is the base of mechatronic design methodology.

Analysis of mechanical proprieties should be carried out at the stage of design of mechanical parts. In our laboratory we use the PATRAN/NASTRAN package for 3D geometry modelling of mechanical parts and stress analysis using the finite element method. The design process is iterative and each iteration loop can be replated to achieve assumed system performance or limit of iteration number. One can go back to change the model of mechanical parts if simulation shows the possibility to improve the shape of this part. Such a process of generating and testing virtual (existing only in a computer memory and on a computer screen) mechanical construction is very common in automotive industry (car bodies, motors, turbines), in aviation (planes, space shuttles, satellites), boats and different equipment. According to American sources, using CAD may result in $30 \div 50\%$ shortage of design time and 30% increase in the efficiency of human work. Authors claim that the mechatronic approach may enlarge this numbers.

When mechanical parts are optimised according to the chosen criteria, their virtual models are used (at the next stage of design) for automatic generation of programs for machine tools. This is performed using the Computer Aided Manufacturing (CAM) software, which can program numerical machine tools and assembly lines or shops for automatic production and assembly. The CAM software may be integrated into the CAD package. This is the case of CATIA, one of the most advanced CAD package.

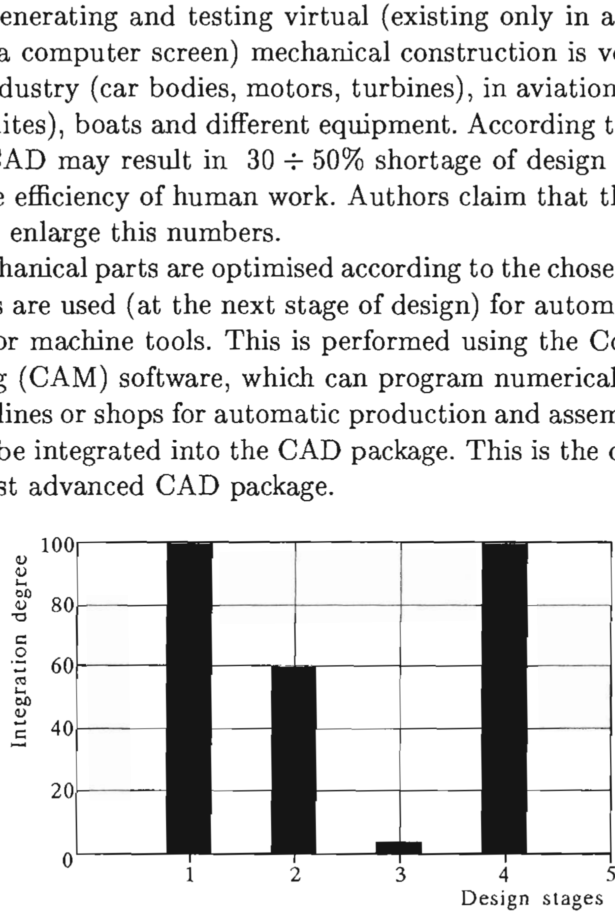

Fig. 3. Mechatronics: integration degree at particular design stages: (1) concept, (2) developing, (3) designing of details, (4) examining and tuning of the prototype

The characteristic feature of mechatronic approach consists in the fact that it is interdisciplinary and that it requires high integration of expertise in many technologies. The degree of integration differs at different stages of its implementation (see Fig.3); i.e.,

– it is very high at the stage of developing and analysing the concept (1) and at the stage of developing (2) and examining (4) of the prototype,

– it is less integrated at the stage (3), when details are designed.

To design the control system (it seems to be most important part of modern equipment) one should define the goal to be achieved, identify properties of the object, choose the control system structure, adjust its parameters and valuate the final product. Computer-aided design can be used at every stage of the designing process.

The way a control system is implemented depends on the system type (single application or mass product). For this reason various tools are used to implement a control (design) system. It is necessary that the software is compatible with various types of designs (mechanical, electrical and electronic) and must have standard interfaces, making communication between particular programs possible. Reader is referred to Uhl et al. (1999).

3. Rapid prototyping

The Mechatronics Laboratory founded in the Department of Robotics and Machine Dynamics is equipped with software and hardware allow realising all stages of computer-aided mechatronic design including; prototyping, testing and implementing of real-time control systems (Bojko and Uhl, 1997; Uhl et al., 1998; Czech and Uhl, 1999). It is based on workstations and PC's connected with the Ethernet network. The proposed system of computer-aided design procedure is presented in Fig.4.

The architecture of the software presented in the diagram can be divided into four basic loops which realise particular step of design process:

- I – Modelling and identification
- II – Synthesis of control systems
- III – Implementation of the control system for single application
- IV – Implementation of the control system for mass scale production.

3.1. Modelling and identification environment

The first loop is used for modelling and identification of parameters of a model. This stage is realised on the basis of available information about an object, presented in the form of documentation and the results of experiments with the prototype. The standard procedure consists in building of a mathematical model or block diagram of an object (a controlled object with

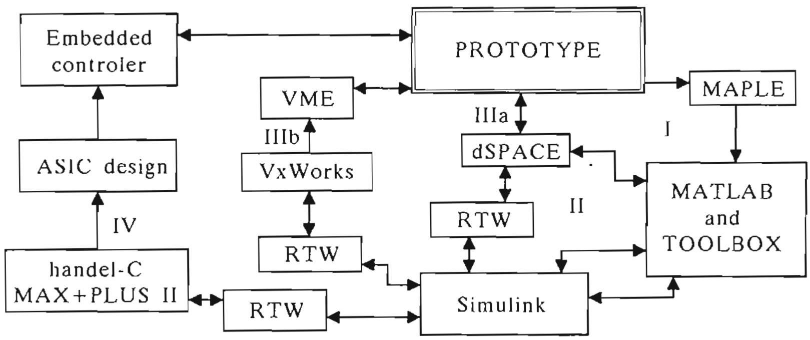

Fig. 4. Proposed system of computer-aided design of mechatronic products

mechanical and electrical parts) in the form of a set of ordinary differential equations. For these purposes MATLAB, Simulink and RTW from The MathWorks Inc. (Mrozek and Mrozek, 1998) (one can use MATRIX$_X$, SystemBuild and AutoCode from ISI instead), EASY V and MAPLE software are used in our laboratory. There are other specialised systems as DADS, ADAMS, DYMOLA. Those systems allow for analysis of the properties of objects and their models - from the point of view of the control system design.

3.2. Synthesis of control system

The second loop is applied to development and optimisation of new concepts of control systems. The computer software enables one to design controller in the form of block diagrams displayed on a computer screen. Block diagrams can be prepared having a poor programming knowledge. Then a real time C code is automatically generated from diagrams, automatically compiled, linked and loaded into flexible prototyping hardware. Any equipment used in prototyping stage should have needed input/output interfaces. This will enable communication with sensors and actuators mounted on the physical prototype of the system.

Synthesis of a control system is completed by its virtual prototyping by means of simulation of the developed model. When the code is loaded, the prototype hardware may perform all functions of the designed controller. This method is known also as *hardware in the loop simulation*, see Uhl (1994).

The controller is not assembled of mechanical and electronic parts but exists virtually in memory of prototyping hardware. During simulation alternative designs of a control system are examined and its parameters are adjusted. This is a fast, reliable and cost effective procedure. One can change controller parameters on the fly, without need to stop simulation or to recompile the computer code. Recompilation and loading of new code is needed when controller block diagram is changed. But it is still much easier, faster and cheaper than to reconstruct or assembly a physical controller.

Automatic generation of a source C code is an important feature of the prototyping environment. New code can be obtained very fast and is believed to be error free. Manual coding needs too much time and may cause problems due to possible human errors.

3.2.1. Supported hardware

It is important to choose appropriate hardware and software to build prototyping environment. High performance processors (even DSP or RISC) are needed to perform all calculations and a real time data transfer (Lim et al., 1996). One should choose prototyping hardware from the following options:

- Single board hardware, stand alone or add-on card to be mounted in a PC computer. dSPACE card dS1102 is an example of single board hardware
- Modular hardware. It is scalable and flexible. A real time operating system is needed. An industrial computer with VME bus and Motorola 68030 processor is an example of modular hardware
- Second, remote PC compatible system may be used as a target computer where the generated code runs in real time. xPC Target software from The MathWorks Inc. supports this configuration. This option is suitable for small and medium size problems only
- Host PC computer may also be used as a target computer. Special software like Real Time Windows Target from The MathWorks Inc, should be used. This option can be sufficient for small size problems only.

3.3. Implementation of a control system

An important question concerning control system design is the best choice of a controller type applied. The choice of a controller should be determined basing on the criterion of its multiple implementation.

The third path (III AB) provides possible implementations of the controller for a single application. Using of DSP processor board or VME based industrial computer is most common choice. Using of micro controller or embedded computer may cut down final cost of small project – but only if the equipment and software needed at the development stage can be reused for other projects.

The alternative method of creating a target control system for mass application is its implementation by means of an ASIC (FPGA) programmable chip or by an embedded controller (autonomous controller or micro controller). This procedure is presented, in Fig.4, within the fourth path.

4. Rapid prototyping of a control system for a flexible arm

The methodology and environment presented have been used for designing the control system of flexible arm in robotics application. Identification of the parameters of this model was an object of the research shown by Auslander and Kempf (1996).

4.1. Experimental identification of flexible arm model

All tests were done using the dSPACE card mounted in a host computer and were repeated using remote industrial VME computer. The results are similar for both the configurations. Single step, square train or white noise signal were used as an input. The signals presented in Fig.5a,b response to a square train signal: encoder signal giving angular position of flexible arm and strain gauge signal were used for identification.

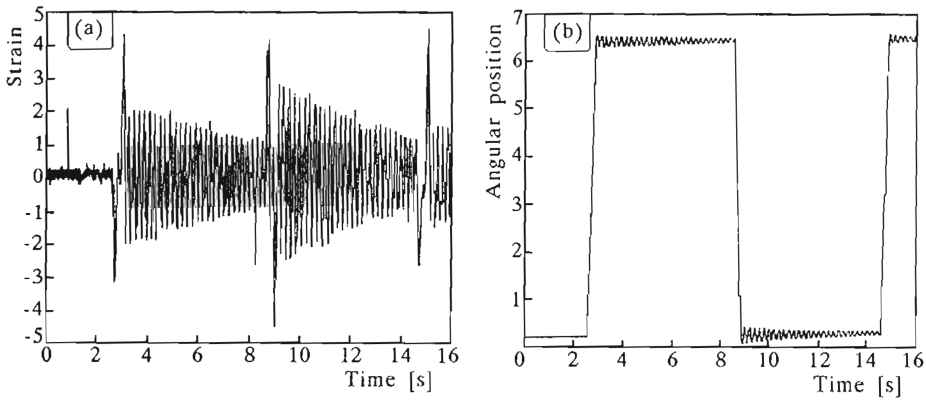

Fig. 5. (a) Strain gauge signal; (b) encoder signal of flexible arm position

The influence of system vibration can be observed and should be included as feedback signal (loop).

4.2. Mathematical model of test stand

This stage is realised on the basis of available information about an object, presented in the form of documentation and the results of experiments. It is assumed that a modal model including three basic modes of vibrations will be used for a flexible arm.

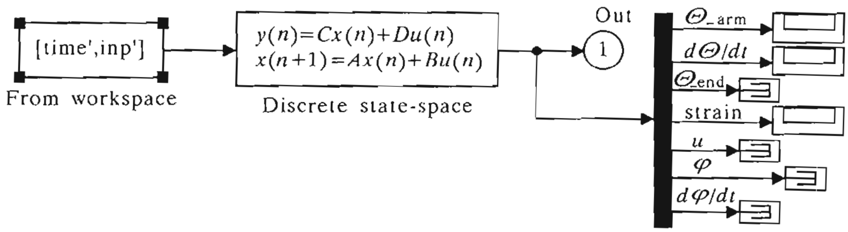

Fig. 6. Simulink state space block diagram

4.2.1. *State space and Simulink model*

A standard discrete state space model (Fig.6) was derived from the set of ordinary differential equations describing dynamics of: DC drive (last 2 equations), gear and oscillating elastic beam (4 equations) – total model order was 11, as described by Lisowski (1996)

$$
\begin{aligned}
&(\cdot)' = \frac{d}{dt}(\cdot) \\
&Y_i\Theta'' + q_i'' = 2\xi_i\omega_i q_i' - \omega_i^2 q_i \qquad i = 1,2,3 \\
&J\Theta'' + Y_1 q_1'' + Y_2 q_2'' + Y_3 q_3'' = K\Big(\frac{\varphi}{n} - \Theta\Big) \\
&L\frac{di}{dt} = Ri - k_u\varphi' + U - k_T\varphi' \\
&J_0\varphi'' = k_i i - \frac{K}{n^2}\varphi + \frac{K}{n}\Theta
\end{aligned}
\tag{4.1}
$$

where

q_i	–	ith modal variable, $i = 1,2,3$
ω_r, ξ_i, Y_r	–	rth resonance frequency, damping and shape ratios, respectively
Θ, φ, n	–	beam and motor angular position, ratio of the gear

L, R, k_i, k_u	–	DC motor constants
k_T	–	tachometer constant
K	–	elasticity coefficient.

The Simulink state space block diagram model is presented in Fig.6. Main advantage of state space approach consists in the possibility of using a variety of standard tools for synthesis of control systems.

The alternate model (Fig.7) was designed in the Simulink environment with use of MSL mechatronic library (Pjetursson and Ravn, 1995). Functional blocks representing the mechanical and electrical parts of the test stand were prepared and extended as the library. The drawback of this physical structure of mathematical model is lacking of standard methodology for designing of a control system. On the other hand, this model is more intuitive and gives a better opportunity to introduce non-linear phenomena appearin in the real object.

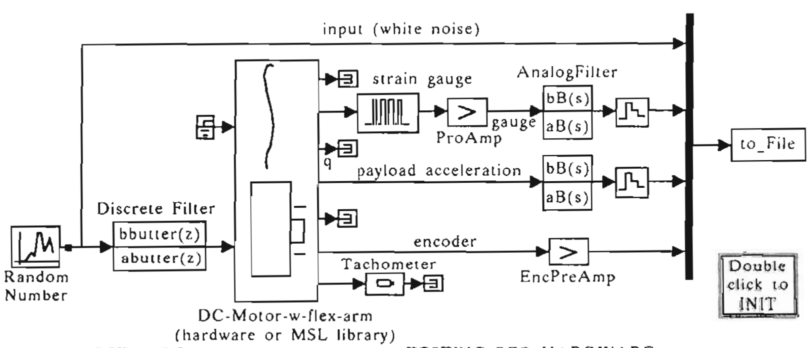

Fig. 7. Simulink block diagram (MSL library used)

4.2.2. *Discrete transfer function model*

A mathematical model of a controlled object (including its mechanical and electrical parts) forms a set of ordinary differential or difference equations. System Identification Toolbox from The MathWorks Inc. was used for parametric identification of the measured signals. A square wave input signal was used. The discrete ARX model with 0.004 s time step was obtained. The following results have been obtained:

— Transfer function: encoder signal vs. motor input

$$G(z) = \frac{0.01621z + 0.05323}{z^2 - 1.173z + 0.1733} \tag{4.2}$$

— Transfer function: tensometer signal vs. motor input

$$G(z) = \frac{-0.2551z^4 + 0.1042z^3 + 0.2233z^2 - 0.02679z - 0.07713}{z^7 - 1.056z^6 - 0.8335z^5 + 0.9167z^4} \tag{4.3}$$

The above models were tuned, using the NCD toolbox, to give a correct response to a step function input (similar to the response of real object). Based on the identified model, the control system design process has been performed.

5. Prototyping environment with dSPACE board

The advantages of applying DSP processors to motion control are numerous. The most important are:

- Possibility of achieving a high speed of computation
- Several dedicated buses in the processor (separate buses for data and instructions)
- Built-in support for real-time applications (DMA channels, timers and communication ports)
- Built-in high-speed internal memory.

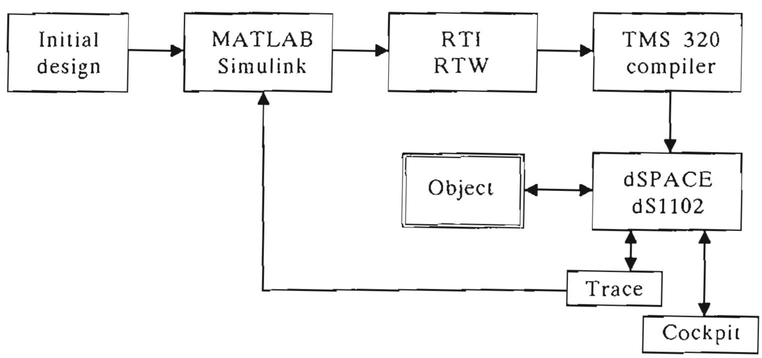

Fig. 8. An example of rapid controller prototyping in mechatronic laboratory

A single board dS1102 add on card from dSPACE GmbH, with the Texas Instruments TMS32OC31 digital signal processor, was used for identification and prototyping of a flexible arm control system. The dSPACE boards operated with Cockpit and Trace software. New versions of this software are offered named MLIB and MTRACE. The Control Desk is a new name for a complete experiment environment. It was not used in our experiments.

The main idea of prototyping board application (e.g. dSPACE board) is to formulate virtual controller and using DSP processor simulating of the controller in real time. To realise interface with the environment, drivers of I/O boards (chanells) should be linked with software model of controller. Those drivers are included to RTI library.

This board and other specialised equipment (single boards and modular hardware) from the dSPACE are designed as a temporary platform for prototyping and research. It is not suitable as a final controller for industry. So another board with different processor and industry standard VME bus was designed in our laboratory (Uhl et al., 1999; Szwabowski et al., 1999) to be used as a target controller for industrial applications.

The actual block diagram for the flexible arm test stand is presented on Fig.9. Drivers for input/output ports are imported from the RTI (Real Time Interface) library.

6. Prototyping with the use of modular VME hardware

The idea presented in this section differs in using the same hardware during the prototyping period (second loop on Fig.4) and as the final solution for industry (loop III B on Fig.4). So it does not generate extra costs for prototyping only hardware. Also no extra time is lost to move the solution from laboratory to industrial standard hardware.

In production of a single unit it is more profitable to apply a programmable solution based on an industrial computer. In order to develop control systems of this type, various industrial computers can be used provided that they are equipped with the real time operating systems. For the purpose of this project a computer with the VME bus and the real time operating system VxWorks were used. The proposed solution has been used for both the prototyping process (second loop) and implementation of a control system (third loop) of a robot flexible arm.

The RTW program automatically generates the C language code from the Simulink block diagram. The code is then compiled and loaded through the network or serial link to the industrial computer memory and then started as a separate task. The industrial computer works under the control of VxWorks version 5.3, the real-time operating system.

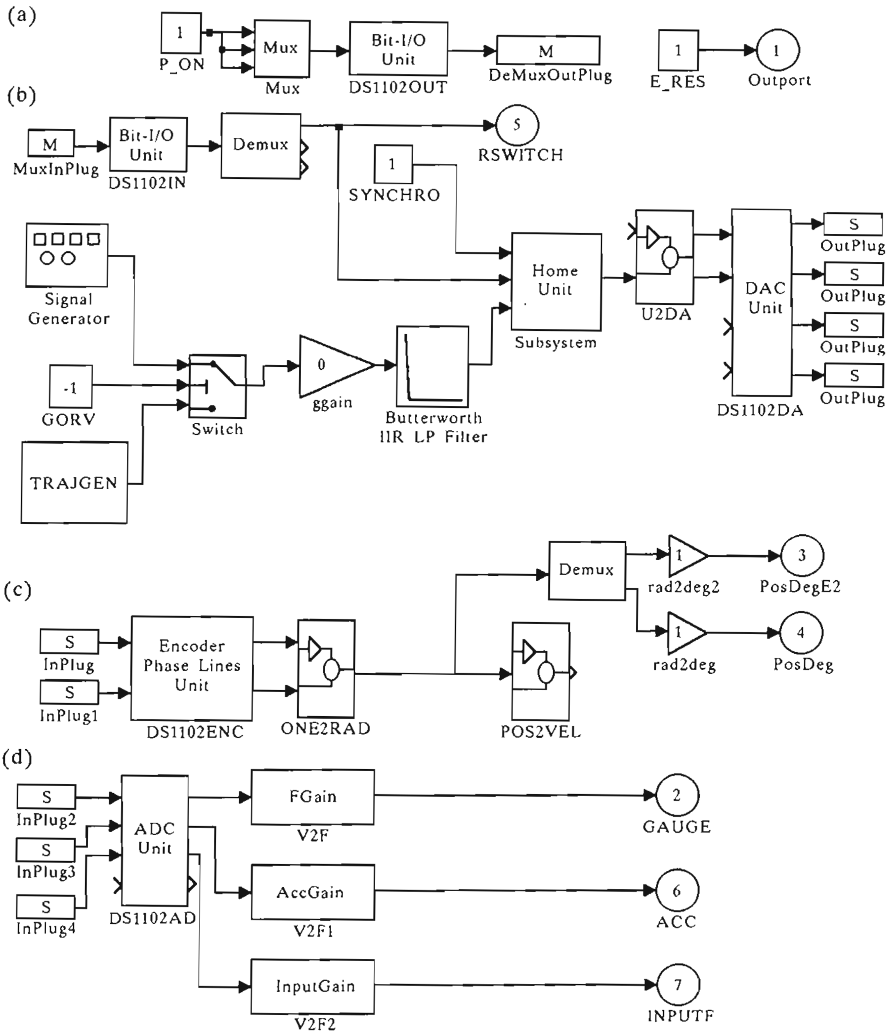

Fig. 9. Simulink block diagram of the flexible arm test stand with dSPACE interfaces: (a) – remote On/Off motor switch, (b) – required position generator, (c) – encoder interface, (d) – data acquisition interface (strain, acceleration, input voltage)

For the system to function it is necessary to prepare custom Simulink blocks, which represent the input/output boards. Creating S functions (alternate description of the Simulink block) completed this task. This block (driver) directly controls the input/output interfaces of the control system.

6.1. VxWorks – real time multitasking operating system

The VxWorks real time operating system from Wind River Systems and software tools (editor, GNU compiler, GNU debugger) create an integrated program environment Tornado.

It also includes a set of tools and utilities (for both the host and target), communication options (Ethernet, serial, etc) and ROM emulator. The Tornado environment is roughly prepared to work with the RTW package. When installed, important services of VxWorks systems are ready to support the development process. Free GNU C-compiler and debugger were also used.

The first mile stone in building rapid prototyping environment was to install successfully the VxWorks, real time multitasking operating system on the industrial computer.

The next step was to configure Simulink and RTW to generate a source code, to compile it, link with library and load a binary code into target hardware: the VME computer. Now, the system is ready and one can:

- Generate a source C code for the chosen Simulink block diagram with mouse click. Then the C code is automatically compiled, linked with needed libraries (including drivers for particular I/O channels) and loaded it into a target VME computer – using the Ethernet link.

- Tune on-line the simulated controller parameters. This can be done by loading the target with a desired process goal and parameters directly from the Simulink running in external mode on the host computer.

One or more jobs are generated in the VxWorks operating system for managing loaded program. If different time steps are used for different parts of the model – supplementary jobs (tRate1, tRate2, tRaten) are introduced. Job tBaseRate has the highest priority and its name is assigned for job with smallest time steps. It is the only job if the model uses one time step only. The other job is responsible for network communication with the Simulink program, which runs on the host computer.

The Simulink, if runs in an external mode, can communicate with target processor (on add-on card or remote computer) using the TCP/IP network or serial connection. This link is used for tuning virtual model parameters (gain, time constant, etc) on the fly, without the need for recompiling the model or stop simulation. On the other hand, if the structure (not parameters only) of the virtual controller is changed – new code should be generated and this needs recompilation. Then a new binary code should be loaded to the remote processor memory.

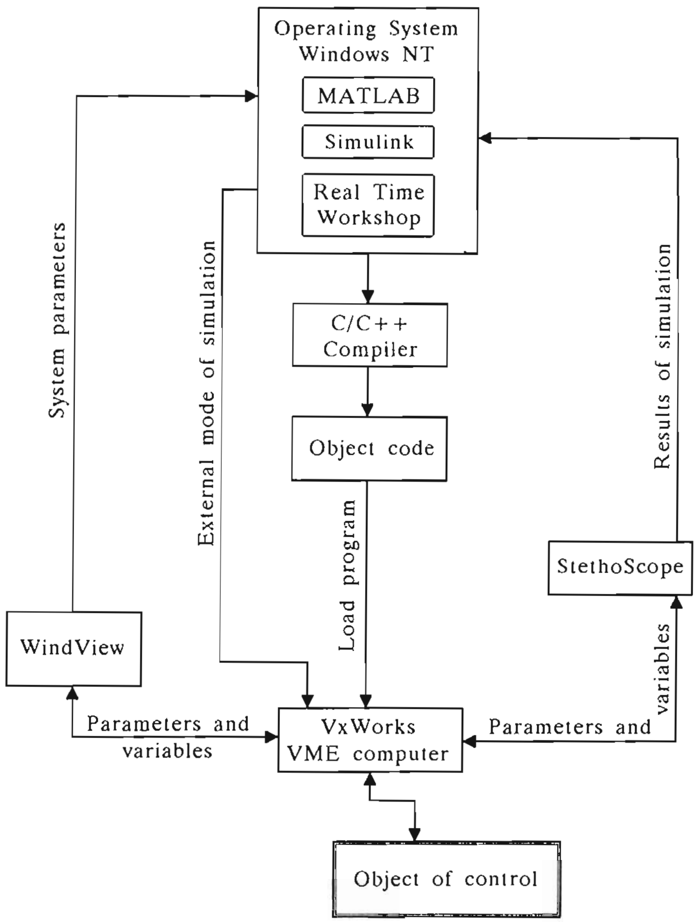

Fig. 10. Fast prototyping using an industrial computer with the VME bus

This environment provides the conditions for program creating, testing and loading the application into the target hardware. Simulink and RTW (Real Time Workshop) software is installed on a host computer. The Simulink works in the external simulation mode and is connected to a target computer via Ethernet. A target computer works under the control of VxWorks operating system. It has input/output interfaces, which connect the controller to the controlled object. Linked computers are presented in Fig.11.

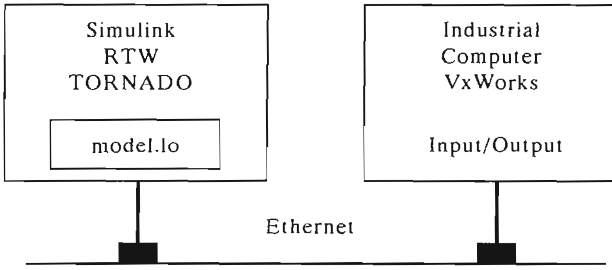

Fig. 11. Scheme of hardware connection

To obtain an extremely fast virtual controller, the controller code should be optimised. In many typical applications, the RTW code linked with drivers is effective enough to achieve the assumed system performance.

6.2. StethoScope and WindView

During the prototyping stage – information about state variables, inputs and outputs is strongly required. The particular task time scheduling is very helpful to find the possibility of better optimised structure. So two other programs are included into the research environment:

- StethoScope [13] data monitor from Real Time Innovations
- WindView state monitor for jobs running under VxWorks operating system.

The StethoScope installs and manages extra jobs in operating system to collect (with desired time step) the values of program variables and chosen parameters. The results can be presented on line on the host computer (in graphic mode) and saved in a file for later examination.

The WindView program visualises the states of all jobs in the system. This is very useful for debugging and tuning jobs (e.g. choosing its time step) WindView helps to tune the overall system performance and to understand the system behaviour in a hazardous situation.

6.3. Hardware description

For controller implementation an industrial computer with the VME bus is employed (Marzec, 1994; Jasiński et al., 1995). The VM42 processor board

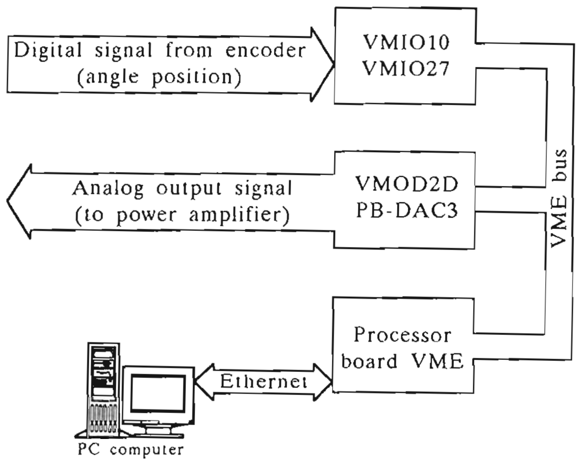

Fig. 12. General structure of the control system for positioning of a flexible arm of robot

(part of the VME based modular industrial computer) is equipped with the Motorola 68040 processor. In order to communicate with sensors and actuators, the following input /output boards were used (see Fig.12):

- PB-ADC3 analog input overlay board including eight 12-bits analog inputs with approximately 30 μs per channel conversion time
- PB-DAC3 analog output board, including four 12-bits, analog outputs with approx. 30 μs per channel conversion time. Both boards were hosted on one VMOD-2D carrier board
- VMIO 27 encoder counter board, which is used to measure the angular position with 24-bits resolution. The board works in the VMIO10 carrier module.

6.3.1. *Custom drivers library*

The second milestone was to prepare driver software that handles a communication between a real time program and an I/O device. Creating our own custom Simulink block in C-language completed this task. They are known as S-functions. They directly control the input/output interfaces of the control system. The supported tasks of each driver are:

- Obtaining dialog box parameter values (from Simulink)
- Initialising the board, i.e., enabling and clearing ports, registers, setting modes of operation, gains and initial condition on signal outputs

- Reading from or writing to hardware (depending on whether the block is for input or output)
- Cleaning up and disabling board when job is finished.

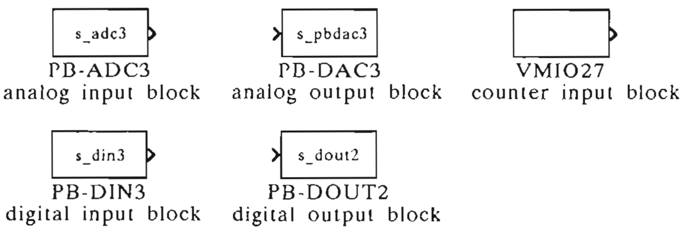

Fig. 13. Device drivers blocks used in fast prototyping

Drivers can be imported from our own custom library (Fig.13) to any Simulink block diagram using simple drag and drop procedures. All drivers were carefully tested in a real time environment and are very reliable now.

6.4. Timing

The target processor needs about 260 microseconds to compute the response of discrete model of control object of 10-th degrees of freedom. Another 85 microseconds extra are needed to generate a control signal using the PID rule – all times were measured using WindView tools. This proves that our controller can use the 1 μs time step without any hazard, implementing even more than one feedback loop and use a more complex control law.

In our research, the time step 4 ms was chosen. It was found that the target computer could effectively control the object and do other tasks as debugging, data collecting and transmitting.

6.5. Testing VME bus computer

To check how Simulink, RTW, Tornado, and VxWorks interact, the results of internal simulation on the Pentium MMX 166 MHz desk top computer and the results of external simulation on the VM42 industrial computer were compared. The differences between both simulations never exceeded the value of LSB error.

7. Verification of the results using the test stand

In order to verify the worked out procedures in an experimental way a laboratory stand was build, which is presented in Fig.14. The flexible arm in this stand is a steel beam with the dimensions: $900 \times 20 \times 6$ mm. The DC motor drive and harmonic gear of 1:82 ratio are used as driving system.

The drive system is controlled by a servo-amplifier, which works in a velocity control mode. A signal proportional to the motor rotational speed is generated by a tachometer coupled to the motor. The angular position of DC motor is measured by encoder with resolution of 250 pulses per turn. Within the laboratory stand it is possible to connect another encoder (18000 pulses per turn) to flexible arm holder. Then a direct measurement of the angular position of the flexible arm holder is possible. A HOTTINGER ME10 amplifier and four strain gauges are glued to an elastic beam. They are used to measure the bending strains. Accelerometer is mounted on the other end of the beam. A static experiment was carried out to determine a scale coefficient between the strain gauge output voltage and the angular displacement of the beam tip.

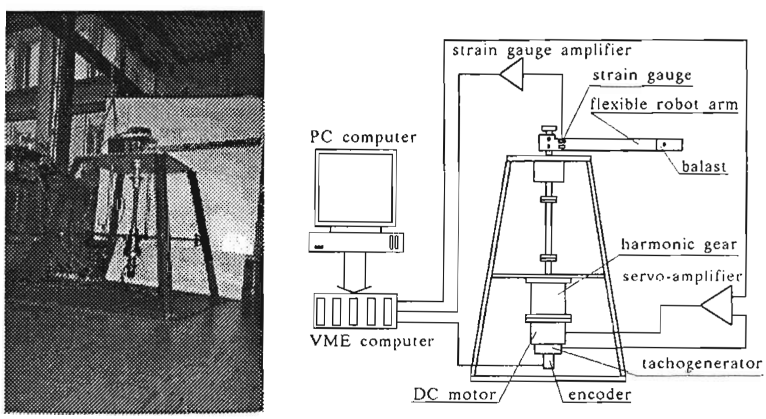

Fig. 14. Test stand with flexible arm, actuator (DC motor drive) and sensors (encoders, strain gauges and accelerometer)

The Simulink block diagram used for prototyping control system for flexible arm test stand is presented in Fig.15. Using the VME modular computer one can use similar block diagram, but the driver should be imported from our custom library (see Fig.13) instead of RTI library.

Several control algorithms were tested, including simple PI and PD controllers and more advanced predictive controllers. Some others (e.g. neural and fuzzy) are being investigate now. Figure 14 presents block diagram used

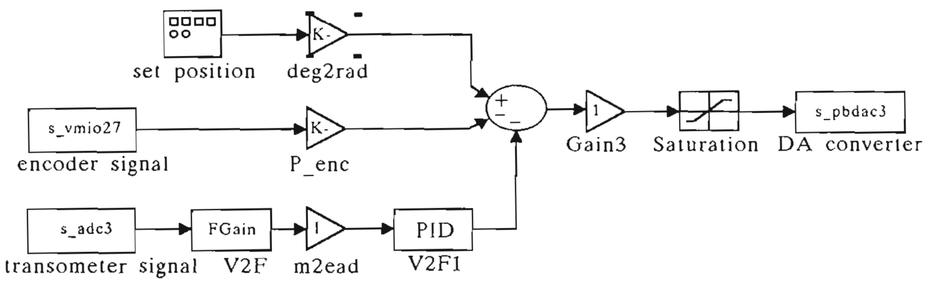

Fig. 15. Simulink block diagram for external simulation of the system with feedback loop of angle position and strain

to tune coefficients for mixed type controller: PID for tensometer signal and proportional for encoder signal. Next pictures show results: using both PID and P controllers give better results than simple P position controller. More results can be found in Uhl at al. (1999).

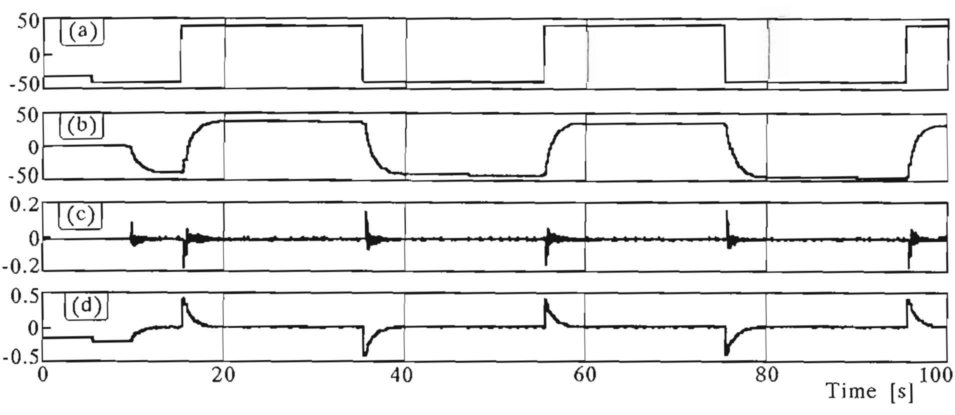

Fig. 16. Results of experiment (signals vs. time): (a) required angle position, (b) actual angle position, (c) signal from tensometer (strain gauge), (d) control signal (to power amplifier)

Comparing the current beam position plots shown in Fig.17 and Fig.18 one can see the difference in oscillation damping. Including only the encoder signal in the control loop (Fig.18) is not effective in oscillation damping. Much better result was achieved when both the encoder and tensometer gauge signals were included in the control loop (see block diagram Fig.15 and plot on Fig.17).

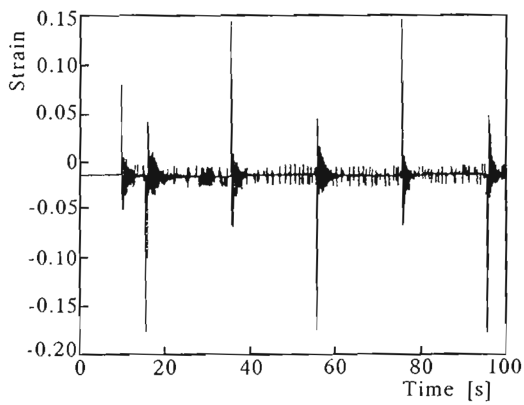

Fig. 17. Vibrations of a flexible arm. Encoder and tensometer signals in the feedback loop

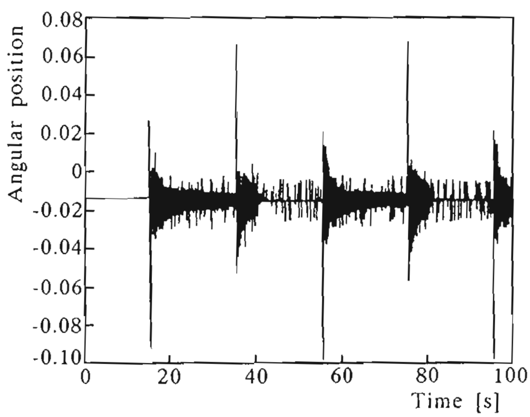

Fig. 18. Vibrations of a flexible arm while only controlling the angular position

8. Conclusions

The prototyping and control systems implementation procedure developed and tested within the framework of this project, shortens the time necessary for implementation of complex motion control systems and carrying out polioptimisation of parameters of designed systems.

As a result of iterative adjustment during prototyping we obtain a control algorithm, which is optimised for the assumed criteria.

Within the framework of this project an implementation methodology for the prototyped controller was developed on a computer with the VME bus and a real time operating system. At the time of the research the high control quality was assessed (for both: the DSP board and an industrial computer).

Acknowledgements

The work presented was part of the research project financing by the State Committee for Scientific Research under grant No. 7TO7B014 14.

References

1. AUSLANDER D.M., KEMPF C.J., 1996, *MECHATRONICS...*, Prentice Hall, London
2. ARAI F., FUKUDA T., 1998, *Micromanipulation – Three-Dimensional High Speed Micromanipulation*, Tutorial notes, Tutorial on Micro Mechatronics and Micro Robotics, IEEE ICRA98, 372-378
3. BOJKO T., UHL T., 1997, Application of Fast Prototyping Methodology in Robot Design, (in Polish), *Mat. Konf. I Krajowej Konferencji Metody i Systemy Komputerowe w Badaniach Naukowych i Projektowaniu Inżynierskim*, AGH Kraków, 343-348
4. CZECH M., UHL T., 1999, Using RT-Linux in Data Acquisition (in Polish), *2-nd National Conference: Metody i Systemy Komputerowe*, Kraków, 19-21
5. JASIŃSKI S., MACIEJEWSKI M., SZYDŁOWSKI J., 1995, VME is Young Again (in Polish), *Elektronika*, 6, 12-16
6. LIM S.Y., DAWSON D.M., VEDAGARBHA P., 1996., Advanced Motion Control of Mechatronic Systems Via a High-Speed DSP, *Mechatronics*, **65**, 1, 101-122
7. LISOWSKI W., 1996, Identification of Flexible Arm of Robot, PhD thesis, (in Polish), AGH Kraków
8. MARZEC B., 1994, *Introduction to VMEbus Standard*, (in Polish), Warszawa
9. MROZEK B., MROZEK M., 1998, *Matlab 5x Simulink 2x*, (in Polish), PLJ Warszawa

10. PJETURSSON A., RAVN O., 1995, *Mechatronic Simulink Library, Reference Manual*, Institute of Automation, Technical University of Denmark, 2. edition
11. *Real-Time Workshop.*, User's Guide. Version 3. The MathWorks Inc., 1999
12. SÖDERSTRÖM T., STOICA P. 1994, *System Identification*, Prentice Hall (Polish edition: Wydawnictwo Naukowe PWN, Warszawa 1997)
13. STETHOSCOPE FOR WINDOWS, User's Manual. Version 1.1, Real-Time Innovations, 1996
14. SZWABOWSKI W., BOGACZ M., UHL T., 1999, Fast Prototyping of Controller in Real Time Systems, (in Polish), *2-nd National Conference: Metody i Systemy Komputerowe*, Kraków, 25-27
15. SZYMKAT M., UHL T., 1994, Matlab Based Environments for Mechatronic Robot Modelling, In *European Robotics and Intelligent System Conference – EURISCON'94*, II, 817-829, Malaga, Spain
16. *Toranado* User Guide, User's Manual, Wind River Systems Inc., 1997
17. UHL T., BOJKO T., MROZEK Z., SZWABOWSKI W., 1998, Control of Flexible Robot Arm, Fast Prototyping and Implementation (in Polish), *I Krajowe Warsztaty Technologii Szybkiego Prototypowania z Zastosowaniem Procesorów Sygnałowych*, Kraków, 17-25
18. UHL T., 1994, Hardware in the Loop Simulation – New Method in Prototyping for Control and Monitoring (in Polish), *Pomiary Automatyka Kontrola*, 5, 118-124 (reprinted in Komputerowe wspomaganie prac projektowych, CCATIE, Kraków 1995)
19. UHL T., BOJKO T., MROZEK Z., 1999, Mechatronic Approach Towards Designing and Implementing of Control Systems, *RoMoCo*, Poznań, 126-132
20. *VxWorks*, Programming Guide 5.1, Wind River Systems, 1996

Szybkie prototypowanie w projektowaniu mechatronicznym

Streszczenie

W pracy przedstawiono metodykę projektowania mechatronicznego ze szczególnym uwzględnieniem procesu prototypowania. Przedstawione procedury umożliwiają skrócenie czasu powstawania produktu oraz integrację dziedzin wchodzących w skład mechatroniki. Szczególną uwagę zwrócono na interdyscyplinarny charakter projektowania mechatronicznego oraz rolę budowy i badania prototypu w realizacji współczesnych produktów mechatronicznych. Przedstawiono system komputerowego wspomagania projektowania mechatronicznego. Jako przykład zastosowania opracowanych procedur wykonano układ sterowania elastycznym ramieniem robota.

Manuscript received December 21, 1999; accepted for print April 14, 2000