Published in Journal of Theoretical and Applied Mechanics, 3,38,2000 pp645-668

RAPID PROTOTYPING OF MECHATRONIC SYSTEMS

Tadeusz Uhl*, Tomasz Bojko*, Zbigniew Mrozek**, Wojciech Szwabowski*
*University of Mining and Metallurgy, Cracow, Department of Robotics and Machine Dynamics,
Al. Mickiewicza 30, 30-059 Cracow, Poland,
** Cracow University of Technology (PK), Department of Computer Research, E-5
Warszawska 24, PL 31-155 Cracow, Poland,
tuhl@rob.wibro.agh.edu.pl,

Abstract. The paper describes a mechatronic approach towards designing and implementing complex motion control systems. Particular attention was paid to virtual and physical stage of the control system prototyping. 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 flexible robot arm: a test rig 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 [1996] defined mechatronics as 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 internationally as 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 synergetic combination of mechanical engineering, electronic, control and informatics in design of products and processes.

It means, parts of 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 on each stage of 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 on figure 1. There are:
- sensors, data acquisition and pre-processing subsystem
- actuators (motors, drives, gears, etc.) subsystem
- computer hardware (microprocessor, DSP, embedded processor, ASIC, etc.) and software
- mechanical subsystem

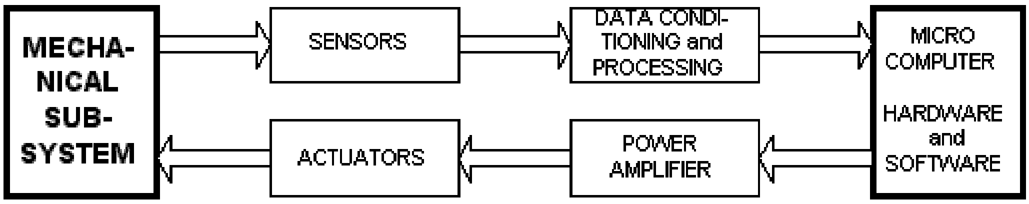


Fig.2.1. An example of mechatronic system structure

Prototyping of mechatronic system is very complex process. Using computer system in management and design is strongly recommended . Very fast progress in computer hardware and software and much more computer power for same amount of money makes it possible to support the development on each stage of prototyping of mechatronic systems.

2.1 C□□ □□□□□□ A□□□□ D□□□□□ □CAD□□□□□ □□□□□□□□□□□□□□□□□

There are many software packages supporting design 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) are used in our laboratory. CCATIA is very expensive and demands 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 to support rapid prototyping. One can simulate movement of mechanical parts described by its mathematical model, external and internal forces, energy, etc, taking into account all restrictions

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

There are several packages (MAPLE V, MATHEMATICA, MAKSYMA, DERIVE, REDUCE, AXIOM) used for symbolic manipulation and processing mathematical formulas. Even 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 symbolic package and obtain almost immediately new set of equations in standard form of mathematical model. The above packages are very useful as they may generate model equations in 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 different nature: mechanical, electrical, thermodynamical, chemical and other. It is modern and gives opportunity of object oriented programming.

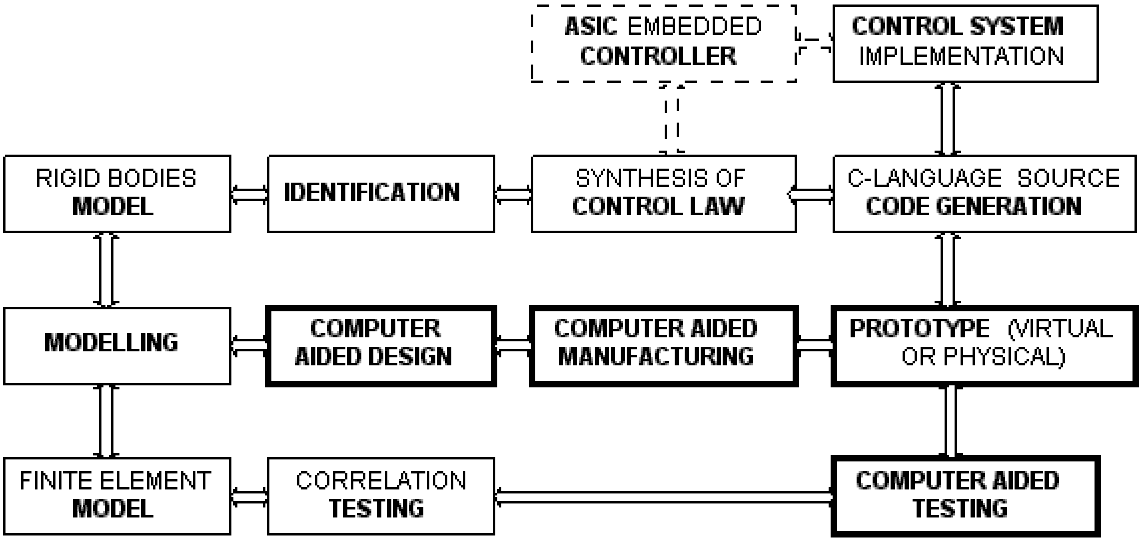


Fig 2.2 Customised structure of computer aided design methodology used in Mechatronic Laboratory of KRiDM AGH

Mathematical model of the system, prepared with any of the above packages or manually (in form of SIMULINK or MATRIX-X block diagram), can be used for simulation in one of following packages: ACSL, SIMON, DESIRE, MATLAB/SIMULINK, MATRIX-X. As model is accepted as set of equations or computer program, original (real) system can be of any physical nature, even non existent in reality (very important class of virtual system). 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 done on a stage of design of mechanical parts. In our laboratory we use PATRAN/NASTRAN package for 3-dimensional geometry modelling of mechanical parts and stress analysis using finite element method. The design process is iterative, on other stages. One can go back to change model of mechanical parts if simulation shows possibility to improve the shape of this part. Such a process of generating and testing virtual (existing only in computer memory and on 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 o 30-50% shortage of design time and 30% increase of efficiency of human work. Authors claim that mechatronic approach may enlarge this numbers.

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

The characteristic feature of mechatronic approach is that it is interdisciplinary and that it demands a high integration of expertise in many technologies. The degree of integration differs at different stages of its implementation:

- 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.

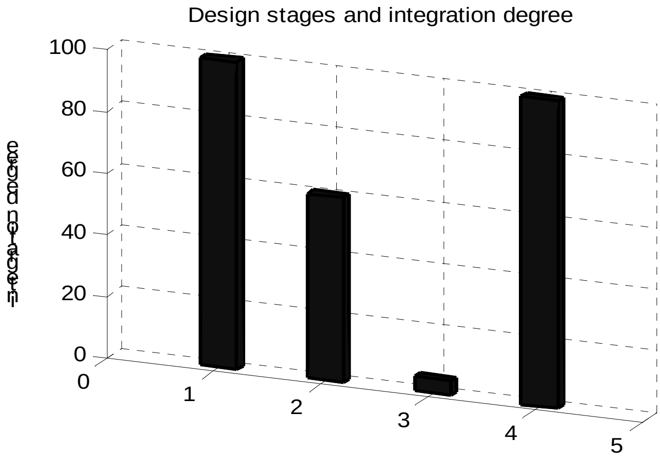


Fig. 2.1 Mechatronics: integration degree at specific design stages:
(1) concept, (2) developing, (3) designing details, (4) examining and tuning the prototype

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

The way a control system is implemented depends on the type of system (single application or a 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. This was described by Uhl et al in [RoMoCo 1999]

□ RAPID PROTOT□PIN□

Mechatronics Laboratory created in the Department of Robotics and Machine Dynamics is equipped with software and hardware implementing all of particular stages of computer-aided mechatronic design for prototyping, testing, and deploying real-time systems [Bojko 1997; Uhl 1998; Czech 1999]. It is based on workstations and PC’s connected with Ethernet network. The proposed system of computer - aided design procedure is presented in Fig. 3.1

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

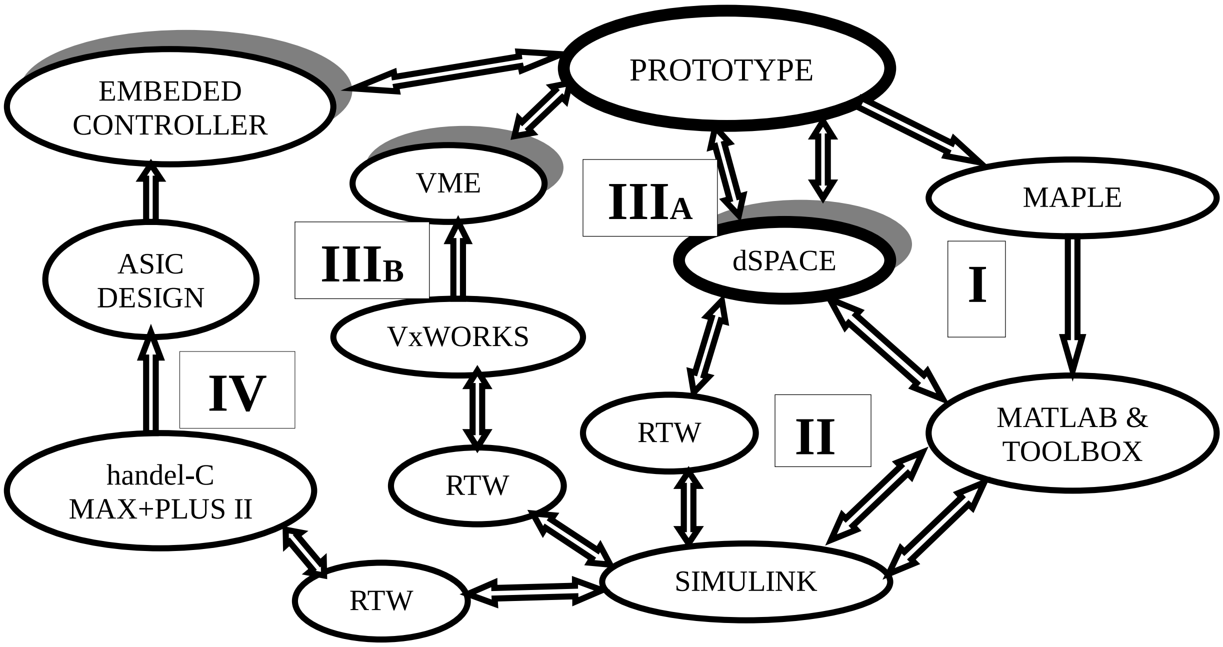


Fig. 3.1 Proposed system of computer-aided design of mechatronic products

□.1 M□□□□□□□ □□□ □□□□□□□□□□□□□□ □□□□□□□□□ □□□

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 form of documentation and the results of experiments with the prototype. The standard procedure is to build a mathematical model or block diagram of an object (a controlled object with mechanical and electrical parts) in the form of a set of ordinary differential. For this purposes *MATLAB, Simulink and RTW* from *The Mathworks* Inc. [Mrozek 1998] (one can use *MATRIXx, SystemBuild* and *AutoCode* from *ISI* instead), *EASY V* and *MAPLE* software are used in our laboratory. There is other specialised systems as *DADS, ADAMS, DYMOLA.* Those systems allow for analysis of the properties objects and their models - from the point of view of the control systems design.

□.2 □□□□□□□□□ □□□□□□□□□□□□□□□□

The **second loop** is applied for development and optimisation of new concepts of control systems. The computer software enables to design controller in form of block diagrams on computer screen. Block diagrams can be prepared with very little of programming knowledge. Then 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 functionality of the designed controller. This method is known also as *hardware in the loop simulation,* see [Uhl 1994]

The controller is not assembled from 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 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 real, physical controller.

Automatic generation of 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 data transfer in real time [Lim 1996]. One should choose prototyping hardware from following options

- Single board hardware, - stand alone or add-on card to be mounted in PC computer. dSPACE card *dS1102* is an example of single board hardware
- Modular hardware. It is scalable and flexible. Real time operating system is needed. 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 can be suitable for small and medium size problems only
- Host PC computer may also be used as a target computer. Special software as *Real Time Windows Target* from *The MathWorks Inc,* should be used. This option can be sufficient for small size problems only

□.□ AI□ □□□ □□□□□□ □□□□□□□□□□□□□□

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

The **third path (III AB)** provides possible implementations of the controller for single application. Using of DSP processor board or VME based industrial computer are most common choice. Using of micro controller or embedded computer may cut down final cost of small project -–but only if equipment and software needed during 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, on Fig.3.1, within the **fourth path**,

□ RAPID PROTOT□PIN□ O□ CONTRO□ □□□TEM □OR □□E□I□□E ARM

Presented methodology and environment has 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 described in [1].

□.1 E□□□□□□ □□□□□□□□□□□□□□□□ □□□□□□□□

All test were done using dSPACE card mounted in host computer and were repeated using remote industrial VME computer. The results are similar for botch configurations. Single step, square train or white noise signal were used on input. Signals presented in figure 4.1 and 4.2 (answer for square train signal:

- encoder signal giving angular position of *flexible arm*
- strain gauge signal

were used for identification.

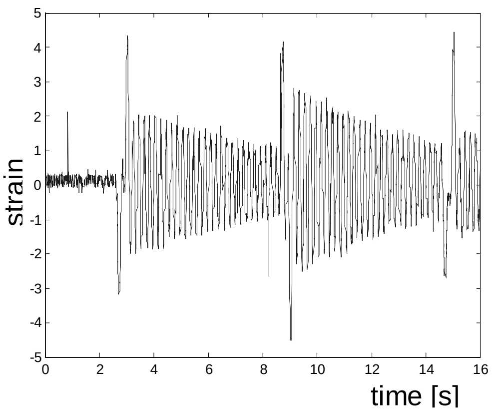


Fig. 4.1 Strain gauge signal.

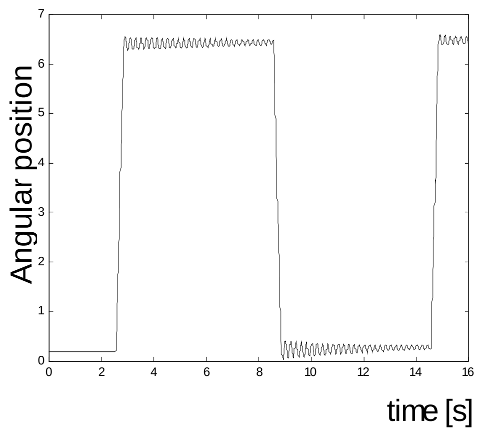


Fig. 4.2 Encoder signal of flexible arm position.

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

□.2 M□□□□□ □□□□□□ □□□□□□□□□□□□

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

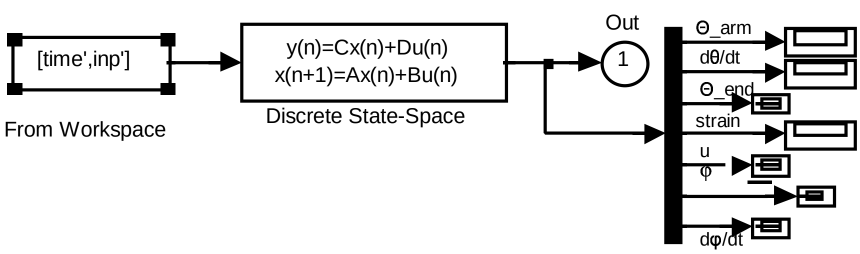


Fig. 4.3 Simulink state space block diagram

4.2.1 State space and Simulink model

Standard discrete state space model (Fig. 4.3) was derived from set of ordinary differential equations describing phenomena of DC drive (last 3 equations), gear and oscillating elastic beam (8 equations) –total model dimension was 11, as described by Lisowski [1996].

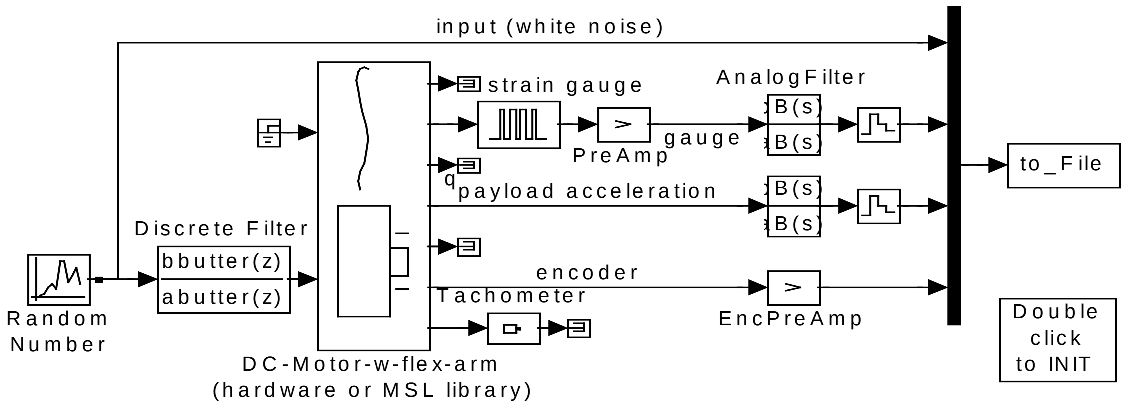
$$
\begin{aligned}
&q'_1 = d/dt\, q_1 \\
&q'_2 = d/dt\, q_2 \\
&q'_3 = d/dt\, q_3 \\
&Y_1\Theta'' + q''_1 + = 2\xi_1\omega_1 q'_1 - \omega_1^2 q_1 \\
&Y_2\Theta'' + q''_2 + = 2\xi_2\omega_2 q'_2 - \omega_2^2 q_2 \\
&Y_3\Theta'' + q''_3 + = 2\xi_3\omega_3 q'_3 - \omega_3^2 q_3 \\
&\Theta'' = d/dt\, \Theta' \\
&J\Theta'' + Y_1 q''_1 + Y_2 q''_2 + Y_3 q''_3 = k/n*\varphi - K\Theta \\
&L*di/dt = Ri - k_u \varphi' + U - k_T \varphi' \\
&\varphi' = d/dt\, \varphi \\
&J_o \varphi'' = k_i * i - K/n^2 * \varphi + K/n * \Theta
\end{aligned}
\qquad (4.1)
$$

where:

q_i		- i^{-th} modal variable, i=1,2,3
ω_r, ξ_i, Y_r	r^{-th}	- resonance frequency, damping and shape ratios
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

Simulink state space block diagram model is presented on Fig. 4.3. Main advantage of state space approach is possibility to use many of standard tools for synthesis of control systems.

The alternate model (Fig. 4.4) was designed in the Simulink environment with use of MSL mechatronic library [2]. Functional blocks representing the mechanical and electrical parts of test rig were prepared and extended the library. The drawback of this *physical* structure of mathematical model is lacking of standard methodology for design a control system. On the other hand, this model is more intuitive and gives better opportunity to introduce non-linear phenomena of real object



!-----------SOFTWARE----------|-------------------TEST BED HARDWARE------------------|

Fig.4.4 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 Matworks Inc*. was used for parametric identification of measured signals. . Square wave input signal was used. Discrete ARX model with 0.004 s time step was obtained Here are results:

Transfer function: encoder signal vs. motor input

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

Transfer function: tensometer signal vs. motor input

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

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

5 PROTOTYPING ENVIRONMENT WITH dSPACE BOARD

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

- a 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.

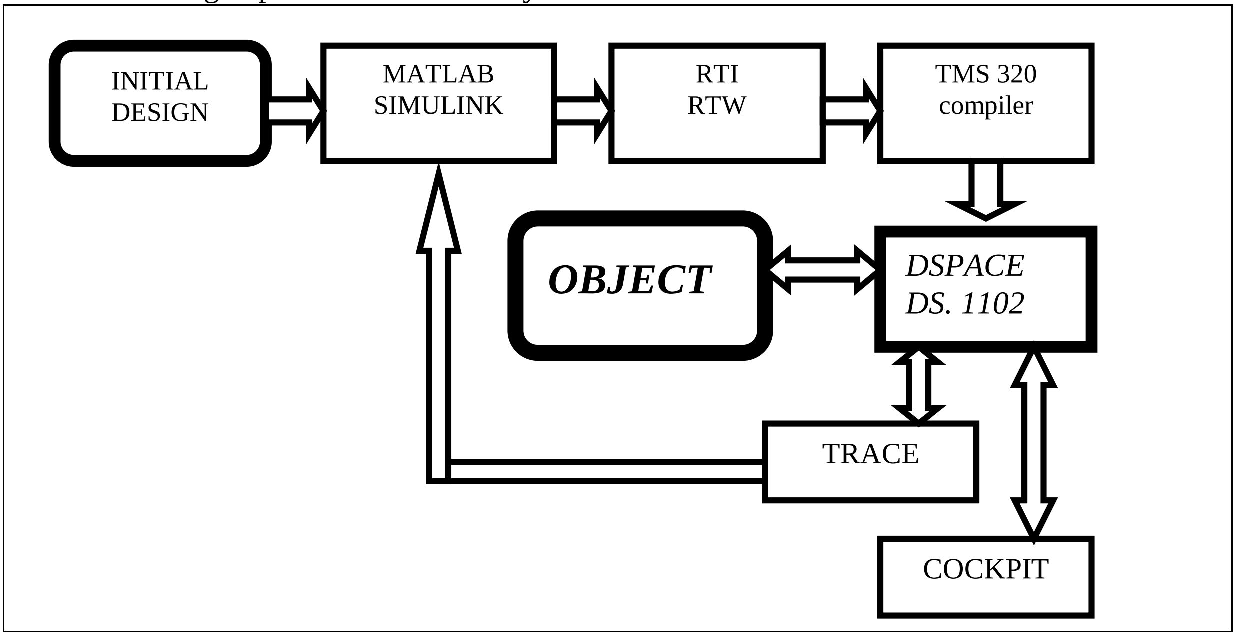


Fig. 5.1 An example of rapid controller prototyping in mechatronic laboratory

A single board *dS1102* add on card from *dSPACE GmbH.*, with *Texas Instruments TMS32OC31* digital signal processor, was used for identification and prototyping of flexible arm control system. dSPACE boards operate with *Cockpit* and *Trace* software. New versions of this software are offered with names *MLIB* and *MTRACE*. *Control Desk* is new name for 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 dSPACE is designed as temporary platform for prototyping and research. It is not suitable as final controller for industry. So another board with different processor and industry standard VME bus was designed in our laboratory [Uhl 1999; Szwabowski 1999] to be used as target controller for industrial application.

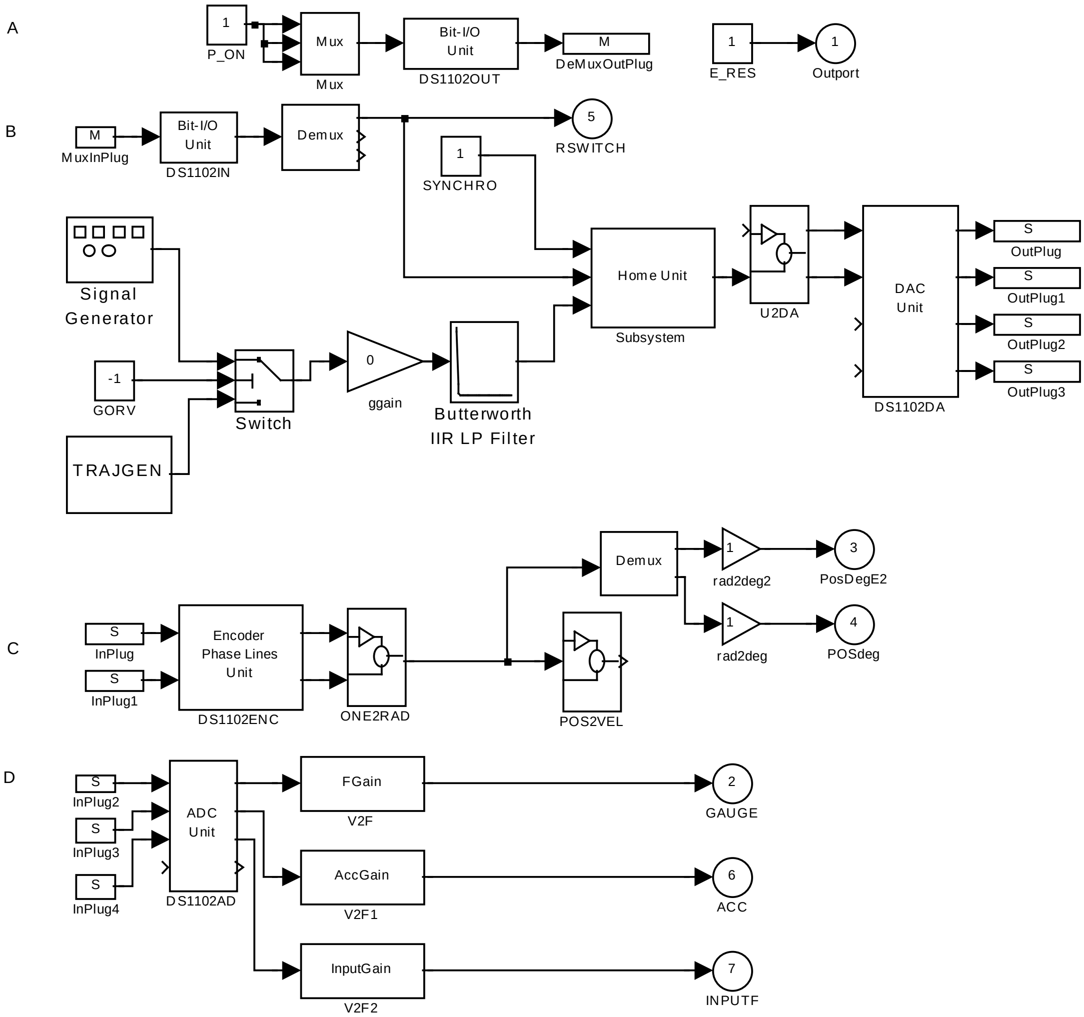


Fig. 5.2. Simulink block diagram of flexible arm test rig with dSPACE interfaces:
A –remote On/Off motor switch, B –required position generator, C –encoder interface,
D –data acquisition interface (strain, acceleration , input voltage)

The actual block diagram for flexible arm test rig is presented on Fig. 5.2. Drivers for input/output ports are imported from RTI (*Real Time Interface)* library. The library is included witch each dSPACE board. Driver names are prefixed with DS1102 name of interface board.

6 PROTOTYPING WITH MODULAR VME HARDWARE

The idea presented in this chapter differs on using the same hardware during prototyping period (second loop on Fig. 3.1) and as final solution for industry (loop III B on Fig. 3.1). So it does not generate extra cost for prototyping only hardware. Also no extra time is lost to move the solution from laboratory equipment to industrial standard hardware.

In single unit production 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 if 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 the implementation of a control system (third loop) of a robot flexible arm.

RTW program automatically generates the C language code from 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.

For the system to function it is necessary to prepare custom Simulink blocks, which represents input/output boards. Creating S functions (alternate description of 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

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 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 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 source code, to compile it, link with library and load binary code into target hardware: the VME computer. Now the system is ready and one can:

Generate source C code for 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 target VME computer – using Ethernet link

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

One or more jobs are generated in VxWorks operating system for managing loaded program. If different time steps are used for different parts of model – supplementary jobs (tRate1, tRate2, tRaten) are introduced. Job tBaseRate has 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 Simulink program, which runs on the host computer.

Simulink, if runs in external mode, can communicate with target processor (on add-on card or remote computer) using TCP/IP network or serial connection. This link is used for tuning virtual model parameters (gain, time constant, etc) on the fly, without need to recompile the model or to 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 new binary code should be loaded to remote processor memory.

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. 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 on Fig. 6.2.

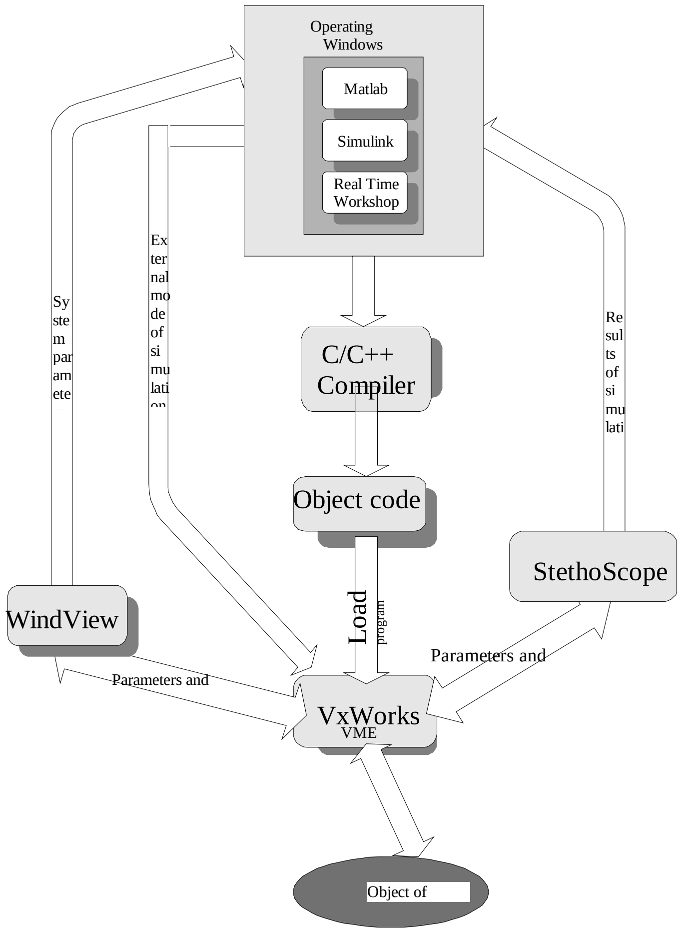


Fig. 6.1 Fast prototyping using an industrial computer with VME bus

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

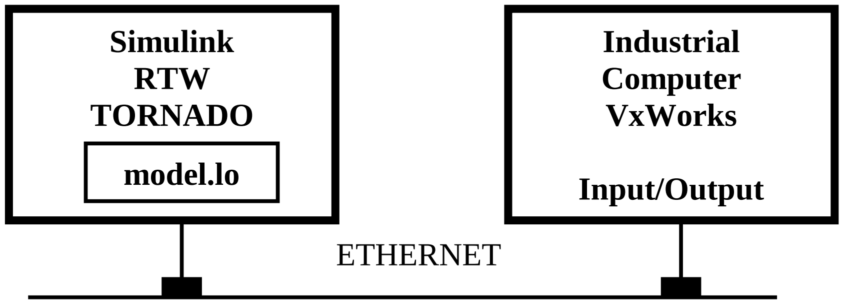


Fig. 6.2 . Scheme of hardware connection.

6.2 StethoScope and WindView

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

- StethoScope [1996] 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) values of program variables and chosen parameters. Results can be presented on line on host computer (in graphic mode) and saved in file for later examination.

WindView program visualises state 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 hazardous situation

6.3 Hardware description

For controller implementation an industrial computer with VME bus is employed [Marzec 1994; Jasiński 1995]. The VM42 processor board (part of VME based modular industrial computer) is equipped with Motorola 68040 processor. In order to communicate with sensors and actuators, following input /output boards were used (see Fig.6.3):

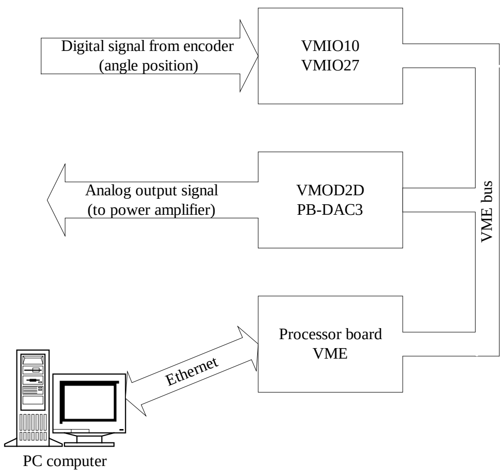


Fig. 6.3 General structure of control system for positioning of *flexible arm* of robot

- the PB-ADC3 analog input overlay board including eight 12-bits analog inputs with approximately 30 μs per channel conversion time,
- the 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.
- the 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 drivers software that handles 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. This means 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

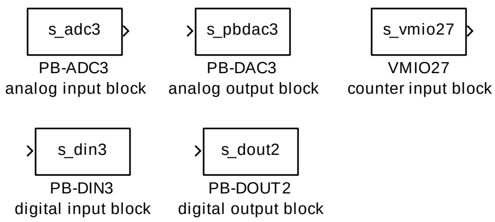


Fig. 6.4 Device drivers blocks used in fast prototyping

Drivers can be imported from our own custom library (Fig. 6.4) to any Simulink block diagram using simple drag and drop procedures. All drivers were carefully tested in real time environment and are very reliable now.

6.4 Timing

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

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

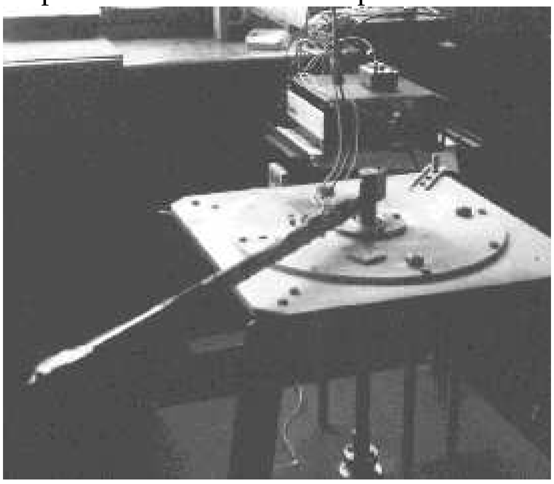
6.5 Testing VME bus computer

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

7 VERIFICATION OF THE RESULTS USING TEST RIG

In order to verify the worked out procedures in experimental way a laboratory rig was build, which is presented in Fig. 7.1. The flexible arm in this station is a steal beam with the dimensions: 900x20x6 mm. DC motor drive and harmonic gear of 1:82 ratio is used.

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 rig it is possible to connect another encoder (18000 pulses per turn) to flexible arm holder. Then 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. Accelerator is mounted on other end of the beam. Static experiment was carried out to determine a scale coefficient between the strain gauge output voltage and the angular displacement of the beam tip.



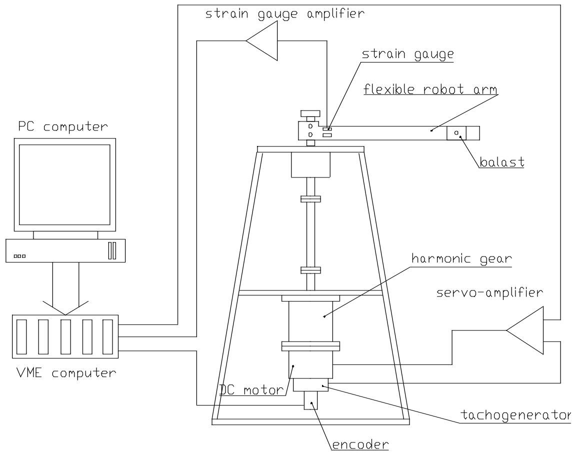


Fig. 7.1 Test rig with flexible arm, actuator (DC motor drive) and sensors (encoders, strain gauges and accelerometer).

Simulink block diagram used for prototyping control system for flexible arm test rig was presented on Fig. 7.2. Using VME modular computer one can use similar block diagram, but driver should be imported from our custom library (see Fig. 6.4) 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 under research now. Figure 14 presents block diagram used 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 all* in [1999, RoMoCo]

Comparing actual beam position plots on Fig. 7.4 and 7.5 one can see the difference in oscillation damping. Including only encoder signal in control loop (Fig 7.5) is not effective in oscillation damping.. Much better result was achieved when both: encoder and tensometer gauge signal are included in control loop (see block diagram Fig. 7.2 and plot on Fig. 7.4)

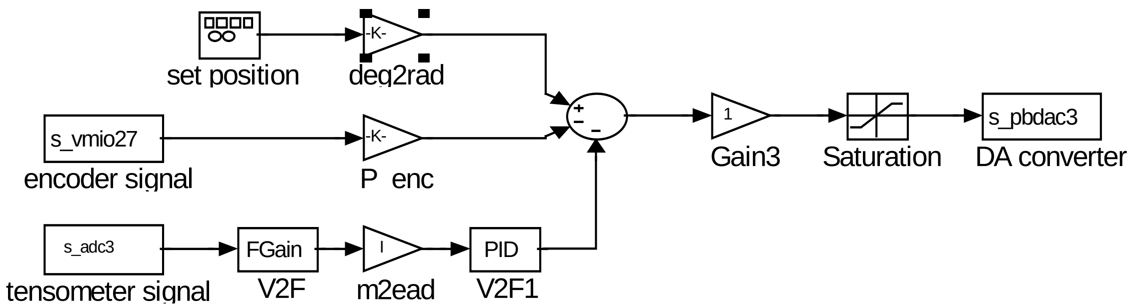


Fig.7.2 Simulink: external simulation with feedback from angle position and strain gauge

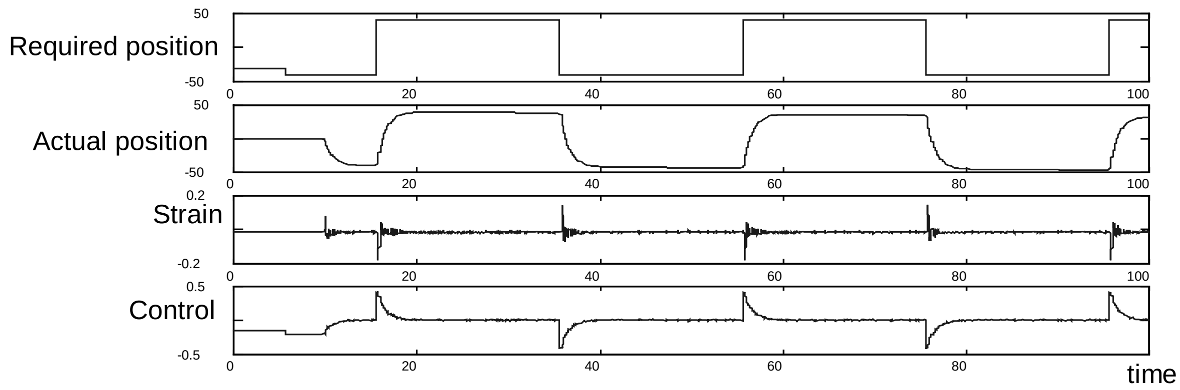


Fig.7.3 Results of experiment (signals vs. time):Desired angle position, Actual angle position, Signal from tensometer (strain gauge), Control signal (to power amplifier).

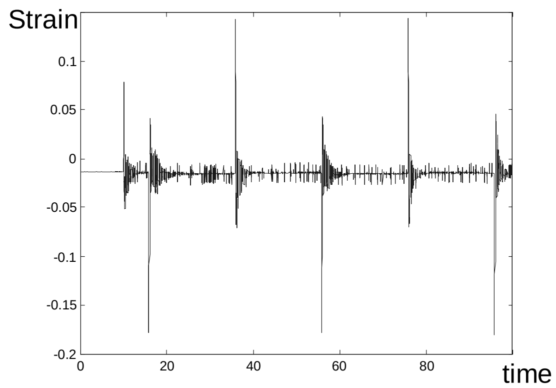


Fig.7.4 Vibrations of flexible arm. Encoder and tensometer signals in feedback loop

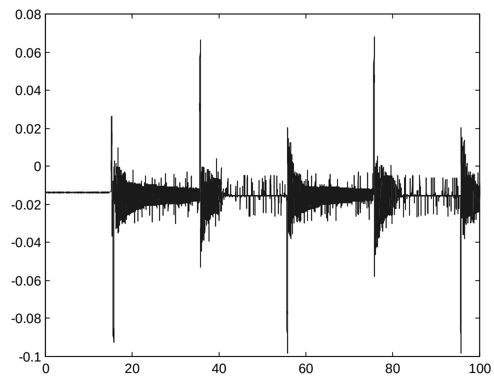


Fig.7.5 Vibrations of flexible arm while only controlling angle position.

FINAL 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 the 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 a 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).

□.□ AC□ NO□ LE□GEMENTS

Authors would like to express their gratitude to the KBN, Scientific Research Committee for financing the research work presented in this paper within the framework of a research project No 7TO7B014 14.

□ REFERENCES

1. .Auslander D.M. Kempf C.J. 1996, *MECHATRONICS*..., Prentice Hall,
2. Arai F. and Fukuda T.: 1998 " *Micromanipulation – Three-Dimensional High Speed Micromanipulation* -", Tutorial notes, Tutorial on Micro Mechatronics and Micro Robotics, IEEE ICRA98,
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, pages.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, Oct. 25-27,.
5. dSPACE information leaflet
6. Jasiński S, Maciejewski M., Szydłowski J, 1995, *VME is young again* (in Polish), Elektronika, nr 6.
7. Lim S.Y., Dawson D.M., Vedagarbha P 1996., *Advanced Motion Control of Mechatronic systems via a high-speed DSP*, Mechatronics vol.65. No.1, pp. 101-122,.
8. Lisowski W. 1996, *Identification of flexible arm of robot*, PhD thesis, (in Polish), AGH Kraków,.
9. Marzec B 1994, *Introduction to VMEbus standard*, (in Polish), Warszawa.
10. Mrozek B, Mrozek M, 1998, *Matlab 5x Simulink 2x*, (in Polish), PLJ Warszawa.
11. Pjetursson, A. and Ravn, O. 1995. *Mechatronic Simulink Library, Reference Manual*. Institute of Automation, Technical University of Denmark, 2. edition.
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. *Real-Time Workshop*., User's Guide. Version 3. The MathWorks Inc.. 1999
15. Szwabowski W, Bogacz, Uhl T: 1999, *Fast prototyping of controller in real time systems*, (in Polish), 2-nd National Conference: Metody i Systemy Komputerowe, Kraków, Oct. 25-27,.
16. Szymkat, M, Uhl, T.. 1994, *Matlab based environments for mechatronic robot modelling*. In European Robotics and Intelligent System Conference -EURISCON'94,pages vol II, pp. 817–829, Malaga, Spain.
17. *Toranado* User Guide, User's Manual, Wind River Systems Inc., 1997.
18. 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, , pages. 17 – 25
19. Uhl T., *Hardware in the loop simulation – new method in prototyping for control and monitoring* 1994 (in Polish), Pomiary Automatyka Kontrola, no. 5, , pages.118 –124 (reprinted in Komputerowe wspomaganie prac projektowych,CCATIE Kraków 1995).
20. Uhl T, Bojko T, Mrozek Z, 1999, Mechatronic Approach Towards Designing and Implementing of Control Systems, , RoMoCo, Poznan , June 28-29,

21. *VxWorks*, Programming Guide 5.1, Wind River Systems, 1996.