Scalable Modular Control Architecture For Walking Machines

Mladen M. Milushev[1] Nikolay V. Krantov[2] and Volker Zerbe[3]

Abstract – **The development of a new generation of flexible robots is based on implementing principle movements as observed in nature, i.e. peristaltic ones or such featured by insects. The creation of a modular architecture for control and processing of information related to the mentioned robot type, which in the viewed case means the creation of a basic module, is the point of the presented paper. Final objective of the suggested approach is the building of architecture by way of generating a multitude of instances of the basic module and completing an interaction among them in higher level of hierarchy in order to secure a rapid adaptation to various configurations of independent multistage structures for the robot's actuating.**

Keywords – mechatronics, mechatronic system, biomechatronics, peristaltic, actuator, sensor, functional decomposition, programmable logical devices, VHDL model, schematic design entry, Y-diagram

I. INTRODUCTION

In recent years intensive research and development in the field of moving robots are heavily influenced by biological species. A persisting tendency towards the development of crawling (Fig.1) and walking robots (Fig.2) is on the way.

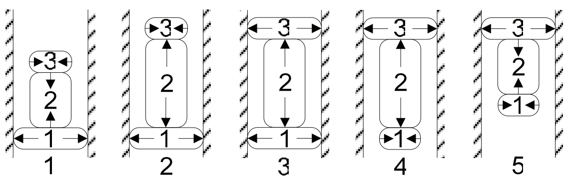

Fig. 1 Movement principle which is constructed from three segments - two holding (1 and 3) and one pushing (2)

Initially for setting robots into motion predominantly electrical motors (converted from the existing robotic engineering) were used. They transferred moments over gears, belts and levers. Disadvantage of such actuators is the lack of passive air-cushioning ability as found in biological muscles. Consequently, researches lean more and more towards fluidic muscles performing peristaltic movements – shortening and extension – mirroring those of living organisms.

[1]Mladen S. Milushev is from the Faculty of Mechanical Engineering, TU-Sofia, Kliment Ohridsky blvd. 8, 1756 Sofia, Bulgaria, e-mail: mcm@tu-sofia.bg

[2]Nikolay V. Krantov is from the Faculty of Mechanical Engineering, TU-Sofia, Kliment Ohridsky blvd. 8, 1756 Sofia, Bulgaria, e-mail: n_krantov@tu-sofia.bg

[3]Volker Zerbe is from the Faculty of Informatics and Automation, TU-Ilmenau, D-98684 Ilmenau, Germany, e-mail: volker.zerbe@tu-ilmenau.de

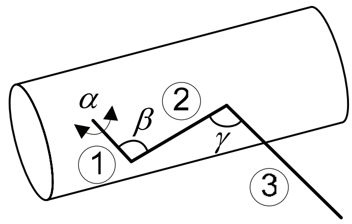

Fig. 2 Six feet structure with three joints

II. CONTROL ARCHITECTURE CHOICE

Depend upon the way they are presented throughout scientific literary sources [1],[2],[3], control concepts for mobile robots can be classified as:

- Analytical description;
- Rule-based control
- Behavior-oriented control
- Neural network-based control

Currently the approach towards mobile robots with behavior-oriented control is domineering. In turn behavior-oriented control concepts can be represented through reactive, deliberative or hybrid control. *Reactive* control systems put direct connection between sensor information and commands to the actuators, which allow robots to react immediately to environment changes. With *deliberative* systems the robot is capable of using the complete existing and current sensor information to plan its behavior, which is a rather complex one. *Hybrid* systems attempt to unite the benefits of both reactive and deliberate systems.

The hereby-suggested architecture (Fig.3) for a mobile robot control roots in the hybrid principle, which means to define the elementary behavior types along with the ways of combining them in order to achieve complex control architecture. The letter consists of four levels: joint/segment, leg/module, body and moving robot. Each level contents intelligent units for interpretation of sensor information and for providing backup actions to the upper level.

The least functional unit is the joint/segment. A control task on this level involves the managing of the pressure and force of the actuator working in antagonistic mode. Since the only alternating variable in the control loop is the fluid flowing in and out of the muscle, the valve takes up 3 positions: fill in-close up- let out. Integrating the reflexive entity on that level proves to be purposeful in regard of the contact force on the surface.

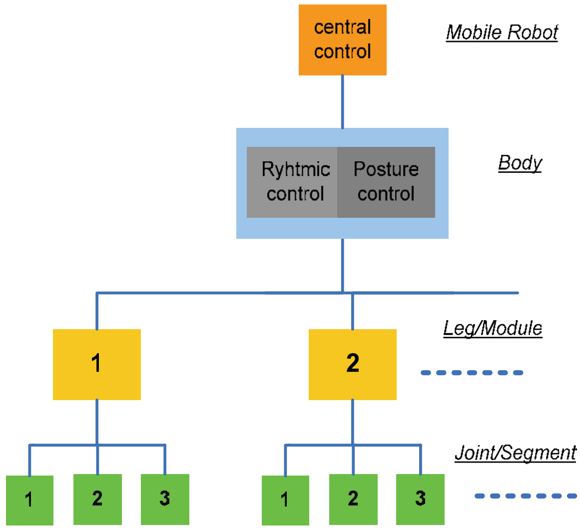

Fig. 3 Six feet structure with three joints

Next functional level is the leg/module that displays a combination of the three joints/segments. Control solutions here are focused on the originating and proceeding of information concerning the leg/module along with providing backup actions for the higher level and thus allowing the leg/module to be viewed as an independent entity. The accompanying integration of a reflexive unit would be appropriate with reference to the force of contact or collision alongside the movement direction. At this point proceeds the regulation of a communication link between the controllers for the leg and those for the body.

The body level is in charge of coordinating the legs in a way granting a static stability of different walking modes, e. g. tripod or tetrapod. Depending on the walk velocity the phase and speed for each of the legs are chosen (Rhythmic Control). Additional function on this level is the control of body alterations (Posture Control). All data required to control horizontal changes and body velocities are calculated using the valid joints/segments angles dimensions or are valued on base of additional body sensor information.

The moving-robot-level performs the highest level of moving systems control. It accounts for the motion's trajectory set up and for body changes as the main control algorithm – coordinating sub-levels. The moving-robot-level should provide for a data exchange in order to:

- Obtain commands, e. g. move to a premeditated position through detection by the own sensors, or ;
- Communicate data about the robot's status quo.

III. Hardware Platform Choice

There are many approaches for implementation of digital systems design: microprocessor and microcontroller based (or software), dedicated ASICs and FPGA-based solutions. Maximum flexibility can be reached with software specifications of the full system; however for highly parallel systems microprocessor based solutions are not suitable. Dedicated ASICs is the best solution for achieving good performance but prototype development makes no sense for using ASICs. At the same time such an approach dramatically reduces the adaptability of the system. FPGA based systems provide both higher performance for parallel computation than software solutions, enhanced flexibility and low development cost compared to ASICs. Moreover it is possible to reduce execution time by implementing intensive computational parts of the algorithm in hardware.

Hardware implementation of electronic systems requires the extensive use of verification stages throughout the design process to guarantee correct functionality of the final product. The use of high-level VHDL for modeling and simulation is especially appealing since it provides a formal description of the system and allows the use of specific description styles to cover the different abstraction levels used in the design (architectural, algorithmic, RTL and logic level as is shown in the wide-spread Y-diagram).

IV. Joint Block Architecture

In the present paper the whole logic design is hierarchical which means that the top-level design file is a VHDL based including component instances that refer to schematic based one.

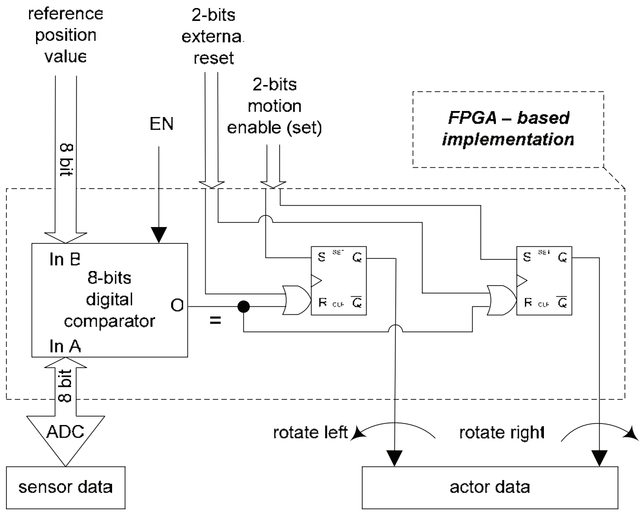

Fig. 4 Block diagram for implementing a joint control

Register transfer level architecture of the joint block follows the logic shown on the figure 4. The main unit in the design is an 8-bits digital comparator and two RS Flip-Flops. The comparator gives us a relation between the predefined value which is set up externally and the current position of the joint achieved from external sensor also with 8 bits of resolution. Equation event between two values activate an output of the comparator which is used in conjunction with two external signals to act as reset for flip-flops. For setting purposes of the logic blocks is used also a 2 bits external signal for motion control. Outputs of the flip-flops are used respectively for enable the action of external actors for left or right rotation movement of the joint as can be seen from the figure. There is included a global enable signal for the digital comparator which enable the entire joint block. So until the current position values of joint reach the predefined position value the actors for rotation and linear movement are active.

V. Implementation

For development was used integrated synthesis environment (ISE) WebPACK which is the ideal solution for FPGA and CPLD design. It offers HDL synthesis and simulation, implementation, device fitting, and JTAG programming. This ISE allows convenient productivity by providing a design solution that is always up-to-date and provides features and functionality at no cost.

The digital architecture of the joint control was entered by using schematic editor which is integrated into ISE. With ready to use library elements translation of the control idea into synthesizable form is very straight forward process similar to conventional CAD tools such as Protel and OrCAD. The ready joint architecture is shown on the following figure.

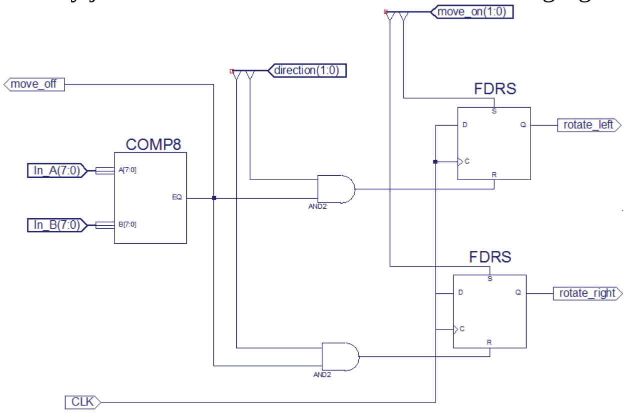

Fig. 5 Joint architecture schematic

Instantiating of the joint module at higher level of the logic design is possible through the VHDL source code shown on the figure 6.

```
COMPONENT Joint
  GENERIC(width  : natural := 8);
  PORT( reset_ext   :  IN STD_LOGIC_VECTOR (1 DOWNTO 0);
          move_en :  IN STD_LOGIC_VECTOR (1 DOWNTO 0);
          GCLK :  IN STD_LOGIC;
          angle_control :  OUT   STD_LOGIC;
          position_control :  OUT   STD_LOGIC;
          In_A :  IN STD_LOGIC_VECTOR (width-1 DOWNTO 0);
          In_B :  IN STD_LOGIC_VECTOR (width-1 DOWNTO 0));
  END COMPONENT;

Joint_1: Joint PORT MAP(
      reset_ext => ,
      move_en => ,
      GCLK => ,
      angle_control => ,
      position_control => ,
      In_A => ,
      In_B =>
  );
```

Fig. 6 Piece of code identifying joint instance

Adding component declaration at the beginning of the top architecture gives us ability to instantiate this component in the body of the architecture infinite times.Also with generics which are especially powerful where there is a reuse policy in mind, we can easy create parameterized in width ADC interface depending on the one in use.

As final step there was examined simulation of the described system using third-party simulator from Cadence – ModelSIM (Fig. 7). The results show generation of control signals for rotating at two directions as was mentioned earlier.

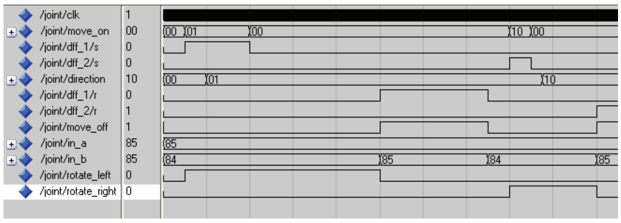

Fig. 7 Logical design simulation

VI. Conclusion

The presented paper offers a structure intended at significant enhancement in the modularity of control for mobile robots, which use the principle of peristaltic motion through antagonistically functioning muscles. The suggested reflexive control mechanisms can be applied to overcome a wide range of obstacles without modelling the latter. The described reflexes can be considered as a first-time application in those systems. The system opens a possibility to widen the presence of reflexes during experiments, to combine them on different levels with reflex chains. For that, types of elementary behavior modes have to be defined as well as models for their combination.

There was shown the full top-down design cycle for implementing digital behavior. Using schematic entry at the lower level is very straightforward approach to describe basic functionality. VHDL at higher level allow flexibility and adaptability of the full body system. Description language also minimizes time for verification and post-processing stages and reduces significantly implementing of fully functional logic system. Trade-off between price and achieved functionality is obvious.

Acknowledgement

Every research this paper is accounting for has been done under the assignment of Contract ВУ – ТН – 201/2006 entitled “Research of a Modular Architecture for the Control of Mechatronic Elastic Multi-Link Devices”

References

[1] O. Matsumoto, W. Ilg, K. Berns and R. Dillmann. Dynamically stable control of the fourleggedwalking machine BISAM in trott motion using foot force sensors. In prodeedings of the International Conference of Intelligent Autonomous Systems (IAS 6), p. 301-306. Venice, July 2000

[2] H. Miura and I. Shimoyama. Dynamic Walking of a biped, International Journal of RoboticsResearch, 3(2):60-74, 1984

[3] Marc H. Raibert, Legged Robots That Balance. MIT Press Cambridge, MA, 1986

[4] M. Vukobratovic, B. Borovac, D. Surla and D. Stokic. Biped Locomotion. Springer-Verlag, Heidelberg, Berlin, New York, 1990

[5] Armstrong J. R., F. G. Gray: "Structured Logic Design with VHDL", Prentice Hall, Englewood Cliffs, 1993

[6] Selz, M., "Untersuchungen zur synthesegerechten Verhaltensbeschreibung mit VHDL", Dissertation am LRS, Universität Erlangen-Nürnberg, 1994

[7] Burriel, R., "Industrial Use of VHDL in ASIC-Design", EURO-VHDL, Hamburg, 1992

[8] Darringer, J., F. Ramming, "Computer Hardware Description Languages and their Applications", North Holland, Amsterdam, 1990D

[9] Cohen, B., “VHDL coding styles and methodologies”, Second. Boston, MA: Kluwer Academic Publishers, 1999. – ISBN 0–7923–8474–1