Simulation Based Methodology for Selection and Integration of Real Time Control Electronics into Complex Dynamic Mechanical Systems

Mekonnen Tesfay Tesfu, J. Schlattmann, Lars Ziemen
Hamburg University of Technology
Hamburg, Germany
Mekonnen.Tesfay@tuhh.de

Abstract

Considering that innovations in modern mechanical products are increasingly of mechatronic nature, selection and integration of control electronics into mechanical systems is becoming an area of great interest for mechanical engineers. The transformation from a mechanical design to mechatronic realization can be very daunting and costly if not managed systematically. Some of the sources of difficulty are: lack of universal standardization of components and their interfaces, too many competing products in market, too many parameters involved and their complex interdependence, confusingly diverse terminologies in product specifications, the need of specialist as well as multi-disciplinary knowledge side by side, etc. The selection and/or design process of mechatronic system can easily get complex when its components have to be put together from diverse vendors with different development software platforms. In some cases, the system could be built completely from commercial off-the-shelf components, while in other cases, some components may have to be designed and built-from-scratch and then integrated to operate smoothly into the system (i.e. make-or-buy decision). To avoid painful interfacing problems at later stage, mechatronic products have to be chosen carefully and the implementation planned with clear awareness.

This contribution will first highlight and discuss the key parameters involved in the selection and integration of real-time control into complex mechanical systems and then propose simulation based methodical approach to tackle the problem. This simulation assisted approach will be demonstrated on an example of a parallel mechanism based biped walking robot Centaurob [1]. The method can greatly help in early configuration, specification, and testing of the system and/or its component parts. It can also lead to an early identification and optimization of the implementation methods and sequences.

1 Introduction

In contemporary machine design and robotics, many formerly mechanical and electromechanical functions are realized using electrical, electronic, and/or software functions e.g. precision servomotors substituting mechanical cams, touch screens replacing press-buttons, new families of MEMS based sensors such as electronic gyroscopes and accelerometers replacing the older bulky mechanical gyroscopes, cable-free connections, etc. Consequently machine design process has practically transformed from old gear/cam based design to electromechanical design process and eventually being transformed to mechatronic design.

In mechatronics design, mechanical, electrical, control, and embedded programming designs merge into a single interdisciplinary field. Also new design methodologies based on computer simulation and virtual prototyping emerge. This integrated approach can immensely reduce design risks and costs by allowing virtual testing of designs in early development process. This is of great importance considering that 60 to 80% of manageable costs in product development are preset by decisions made in early design process [2]. Another very important advantage of early digital prototyping is that it facilitates communication between design teams and also with outside customers. And lastly, but not least, this approach helps in clearly understanding the system and in defining requirement specifications for optimization of control-components selection.

In practice, however, the integrated design approach is very challenging. Despite a huge interest in mechatronics both in academics and industry, many mechanical engineers still seem to shy away from it. One of the reasons is the lack of fully integrated commercial mechatronics design tools that are ripe for the challenge. Today, one can find state of the art commercial CAE tools both in the fields of mechanical and control designs. For example, different mechanical design tools, such as 3D-CAD softwares Pro/Engineer, CATIA, and SolidWorks for visualization and collision detection, MBS softwares Adams and SimMechanics for trajectory generation and motor dimensioning, FEA softwares Abaqus and Pro/Mechanica in design and specification of some mechanical sensors, etc. has been explored during the implementation of this project. Similarly for control design and embedded programming, Matlab/Simulink, LabVIEW, and C/C++, and to some extent assembly language are explored. Most of these tools can accomplish design tasks in their respective domains greatly. However, a fully integrated tool that combines all the functionalities is still missing in the market. As one of pioneering example in this direction, one can mention the interface that National Instruments and SolidWorks have jointly developed recently to integrate their control design and 3D-CAD mechanical design software packages i.e. the beta release of LabVIEW-SolidWorks Mechatronics Toolkit [3].

A full integration of mechanical and control design tools can allow a mechatronic engineers to integrate interdisciplinary design processes without needing to have deep specialist knowledge in the involved fields and without a need to switch tools in the design process. Otherwise, a right combination of design tools from different domains has to be selected carefully based on an application at hand, which is a challenging task in its own right. Mechatronic design and simulation tools facilitate the overall product development process. Hence their selection is usually made at an early development stage. This has, in turn, a big influence on the selection of the subsequent embedded control software and electronic control hardware. Therefore, development software needs to be selected systematically.

In this paper a simulation based methodology for selection and integration of real-time control into mechanical systems will be presented on the example of the biped robot Centaurob. This design process will lead us from the most abstract mechanical system down to the late control components selection step by step.

2 The Walking Machine Centaurob

Centaurob is a statically stable walking biped robot, whose basic working principle has been successfully verified in a previous research work [1]. It is a fairly complex mechatronic machine, which requires over 12 motors, up to 100 sensors, complex control computer systems, a battery energy supply, and two serially connected Stewart platform mechanisms as its components for autonomous functioning. Each of the two Stewart platforms makes up the left and the right legs of the biped robot [Fig. 1]. The Stewart platform structure, also known as hexapod positioner, is the most popular and successful parallel mechanical structure, which was initially proposed by Stewart in 1965. Solving forward kinematic equations of motion for such parallel link mechanisms is not deterministic. Therefore, sensor-based solution methods are needed to compute the robot's forward kinematics.

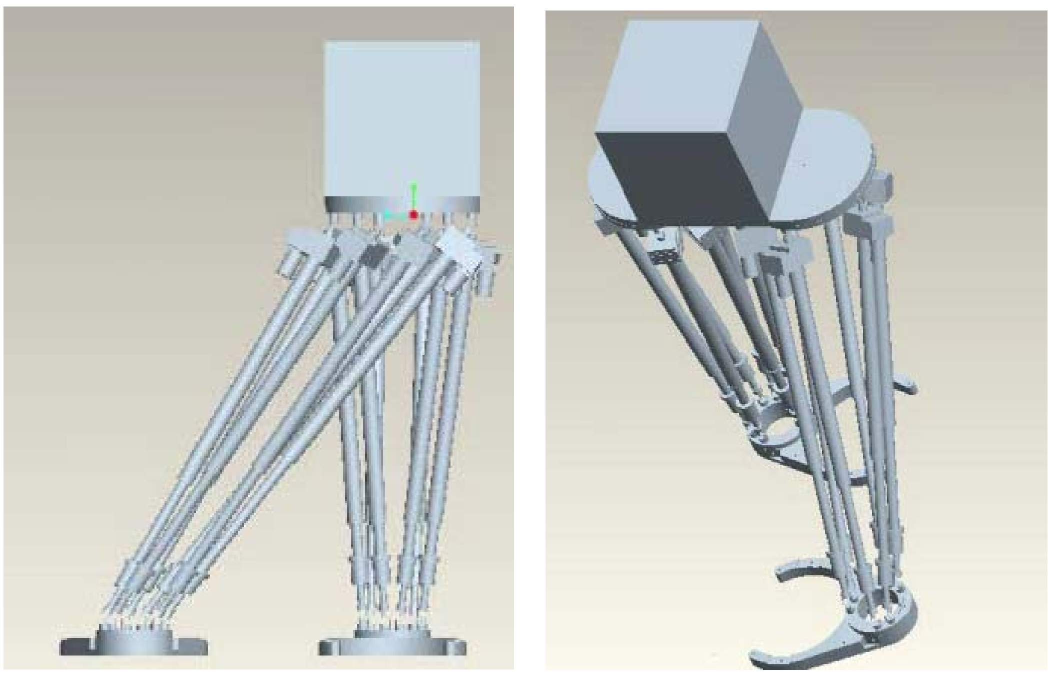

Fig. 1 3D-CAD Views of Centaurob's basic mechanical structure

2.1 Walking Trajectory Planning

The first step in the walking trajectory generation in Centaurob is to plan the swing-foot motion based on the obstacle information [Fig. 2]. The environment information is to be obtained in real-time with localization sensors. This information is to be input in real-time for trajectory planning. Once the swing-foot motion is set, a hip trajectory is determined taking into consideration the stability imbalances that are caused due to the swing-foot motion. Then, based on the local information and other preset waking parameters, the overall stability of the robot is to be maintained using torso and hip motions. Pure mathematical trajectory computation is an iterative process since the motion of the two hexapods, the hip, and the torso are coupled through the kinematic and dynamic equations of motion. Therefore, appropriate sensors have to be integrated to make trajectory computation deterministic.

ISBN 978-1-60643-379-9

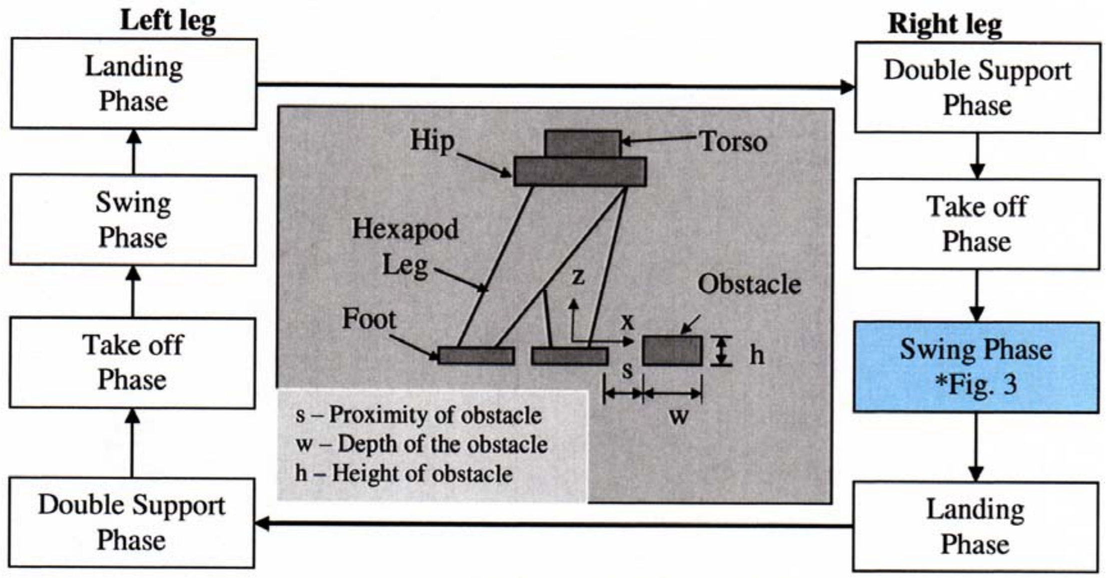

Fig. 2 Walking phases and an obstacle

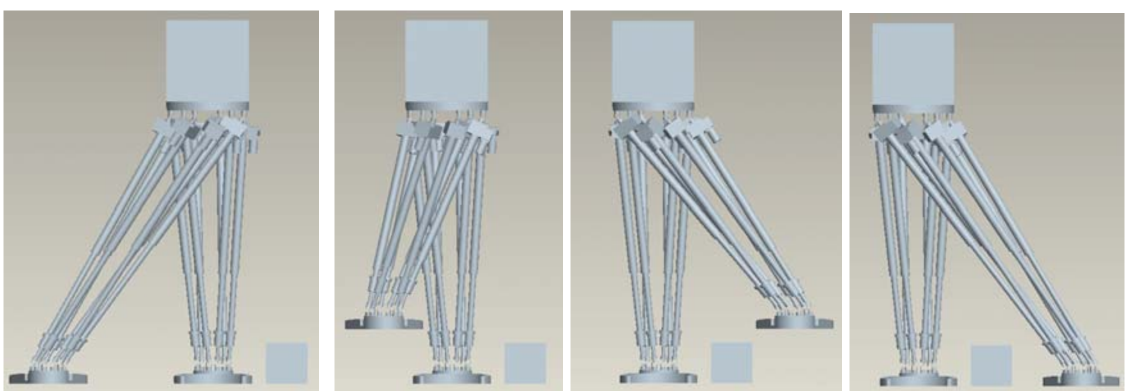

***Fig. 3** States of right leg swing phase

Once the swing-foot trajectory and the corresponding hip trajectory are defined, the joint trajectories are computed using inverse kinematics, from which the required joint speeds, accelerations, and torques can be determined. Inverse kinematics and dynamics of parallel link mechanisms can be deterministically computed. This can be performed either using commercially available multi-body simulation softwares or by self-programmed numerical solution. Here, the two hexapods of Centaurob are modeled using the Newton-Euler-Approach and a numerical solution is programmed in Matlab/Simulink [Fig. 5]. From the inverse kinematics of the hexapods we obtain positions, velocity, and accelerations for each joint/link. The corresponding forces and drive torques acting on each link are similarly determined using inverse dynamics. Fig. 4 below shows a sample plot of link lengths for the six DC-motor driven ball screw actuators of a swinging foot for a given walking-step. This graph is computed from the trajectory generation algorithm presented in Fig. 5.

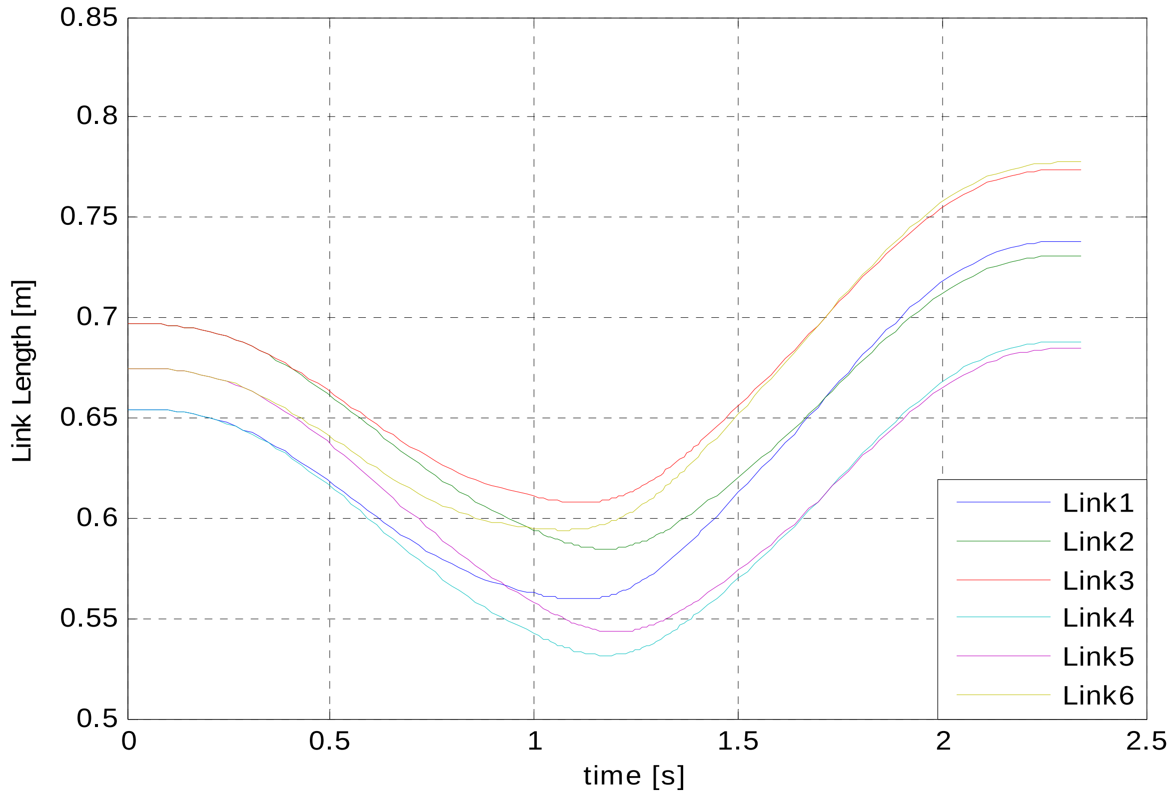

Fig. 4 Position trajectory of the six hexapod links of the standing leg

The control strategy is to generate reference trajectories of the swing foot, the hip, and the torso based on Center of Mass (COM) and/or Zero Moment Point (ZMP) criteria and use feed-forward control for tracking the generated reference trajectories (see Fig. 5). Knowing the location of the COM and ZMP in real-time provides enough information to build a closed loop balance (or stabilization) control for the robot. But a real-time localization of COM/ZMP, based on theoretical kinematic and dynamic models is not deterministic. Real time applications, however, require a deterministic system. An alternative way, therefore, is to compute COM/ZMP from actual state information of the robot, which has to be obtained using sensors. For this purpose, a number of position and acceleration sensors have to be integrated into the Centaurob mechanisms. These sensors are used to determine the absolute position of each component of the robot structure. Stabilization control computations are then performed based on these sensor data.

Hardware wise, the control system is to be constructed on the top of the robot so that the robot can walk autonomously. Because there is a very limited space available for the hardware, the control computers should be as compact and as light as possible. Moreover, the limited availability of energy, which is intended to come from a battery carried by the robot, demands that the power consumption of the computers and other control electronics to be as minimal as possible. Hence, these constraints have to be kept in mind during the control system selection process.

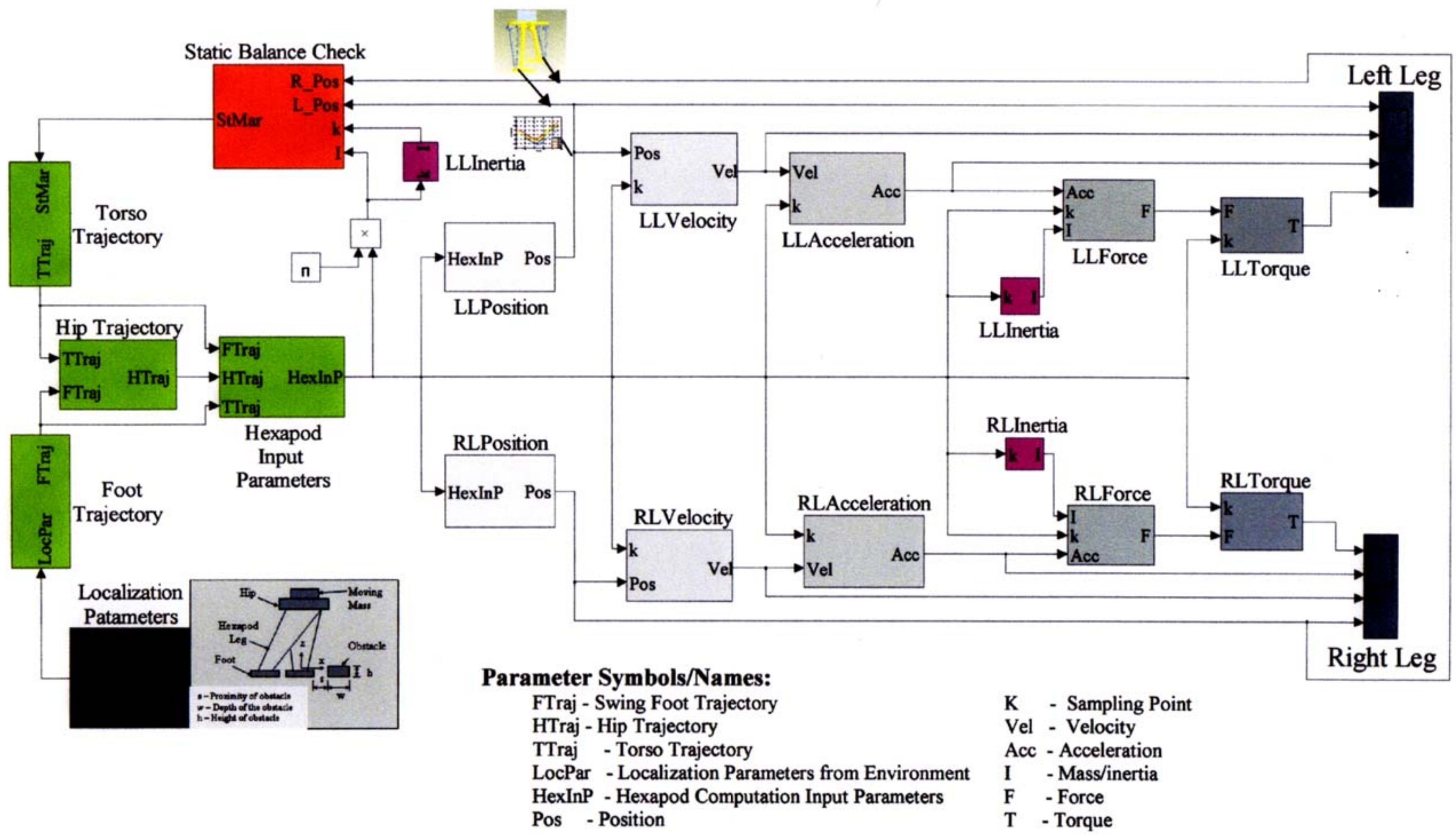

Fig. 5 Walking trajectory generation algorithm using MATLAB/SIMULINK

2.2 Control Architecture

Fig. 6 below shows the overall concept of system control architecture developed on the bases of the Centaurob's walking control strategies discussed above. Here, the control system should be real-time capable i.e. the operating system and bus systems should fulfill the hard real-time requirements, the latency time of sensors and input/output operations should be low enough, etc. The control loop cycle times should now be estimated as accurately as possible based on the critical loops in the previous topic.

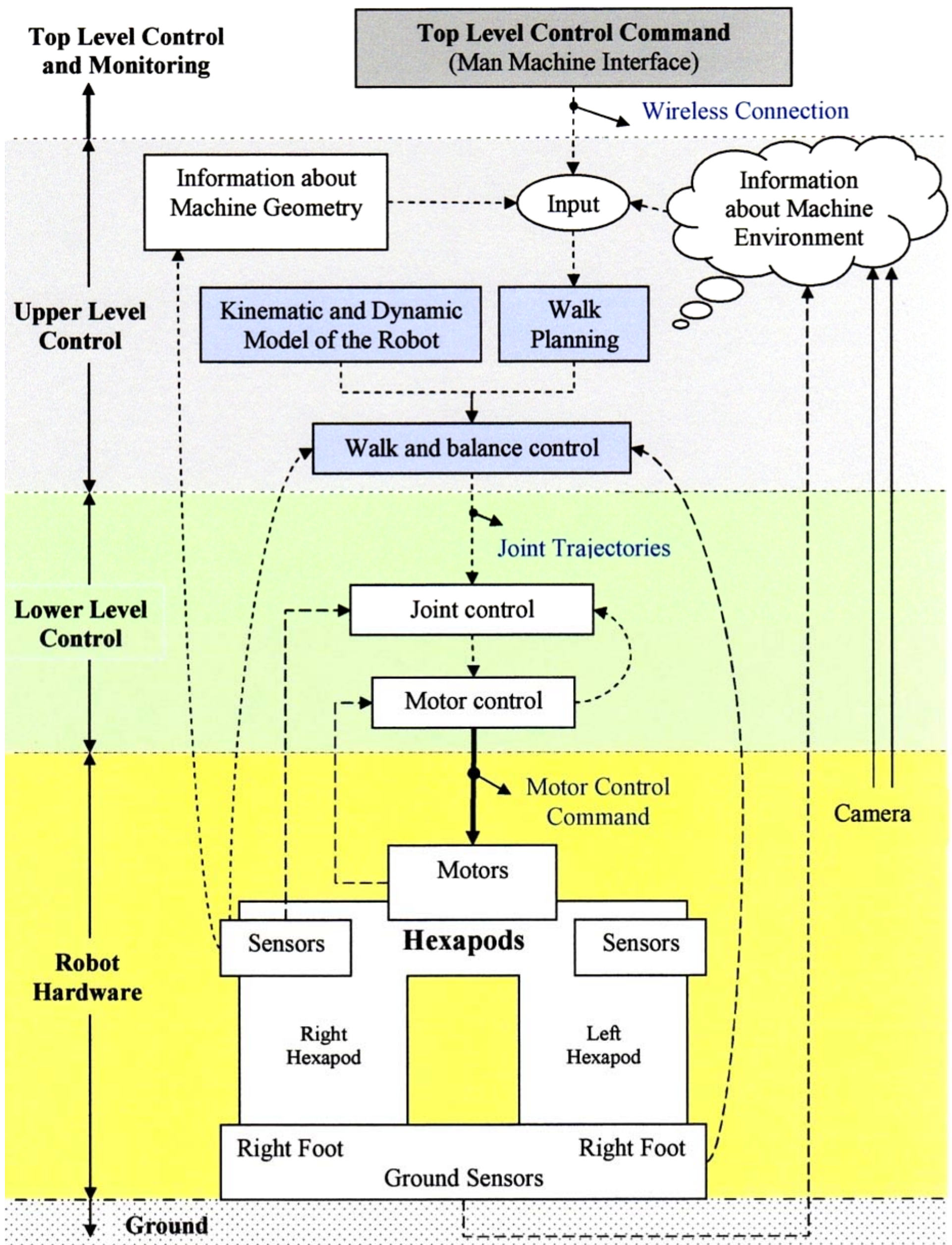

Fig. 6 Centaurob control configuration concept

The method of feedback linearization of the overall control system dynamics requires fast computation, which can best be performed with one powerful central computer. On the other hand a single computer can hardly manage to compute all involved upper level, lower level, and additional peripheral control algorithms in real-time. Taking these two seemingly conflicting requirements into consideration, two basic hardware configuration concepts are considered for the above shown control architecture. The first configuration concept is the use of a central computer and a number of decentralized microcontrollers connected to each other through a bus system. The second configuration concept is the use of a single powerful central computer for the whole control. The difference between the two configurations basically lies at the lower end of the control configuration i.e. the capturing of sensor data and driving of the motors. In both cases the computation of the trajectories and the dynamics of the robot model take place on the central computer, which can compute the overall control routines at high, fixed frequencies. In controlling Centaurob, the time span of the upper level input-output control loop is desired to lie below 4ms (250Hz). This means, any delay in computation time above 4ms or any failure in computing would disturb the control system significantly. This puts a hard real-time requirement on the computer system that can only be fulfilled by real time operating system (e.g. RT-Linux, VxWorks, etc. [4]). For detailed specification of time requirements and communication networks the system should be modeled and simulated in real time [5].

3 Control Components Selection and Integration Tools and Methodologies

Now that the mechanical system and its basic control strategy of Centaurob are specified, it is a good starting point for selecting the control system components. The first step in components selection is usually to list all the inputs and outputs to the controller. From the overall design loops in Fig. 6, it is possible to approximately specify the number of sensors and actuators to be integrated into the system, which is very important information to start with in designing controllers. A reliable approach for detailed specification of the sensors and actuators is system simulation. In turn, for simulation of mechatronic systems one should carefully choose appropriate development software system because choice of a specific tool influences the subsequent selection of system components. The development software, sensors, motors, IO boards, communication buses and interfaces, controller boards and control computers, etc. are all interdependent in some way or another and they should be chosen for smooth interoperability. The software components play equally important role. Ease of assembly is a well known design concept in mechanical engineering as compatibility is in software development. When a system with mechanical, electronic, and software components are to be assembled the combined meaning should be fulfilled. Computer modeling and simulation are nowadays indispensable tools in control system design and their integration into mechatronic systems. In the following section, a simulation platform Matlab/Simulink from Mathworks is used in modeling and simulation of linear units.

3.1 Modular Control System Modeling

The main elements in the control system model shown below [Fig. 7] are the *Mechanical Plant Model* and *Cascaded P-PI-PI Controller*, which are modeled in Simulink. The *DC Motor Model* is a simplified Simulink model of Maxon Brushless DC Motor. The system variables and *Reference Trajectories*, which are computed using a separate path planning and waking-trajectory-computing model [see Fig. 5], are loaded automatically from m-Files. A *3D Model* of the mechanical system can be realized using Virtual Reality Toolbox. Here, *Limit-Checks* can be integrated to observe the position and velocity limits of the moving linear units using Stateflow. Moreover, the *Energy/Power* requirement of the system can be computed online to observe whether the system can get enough power at a given operating point.

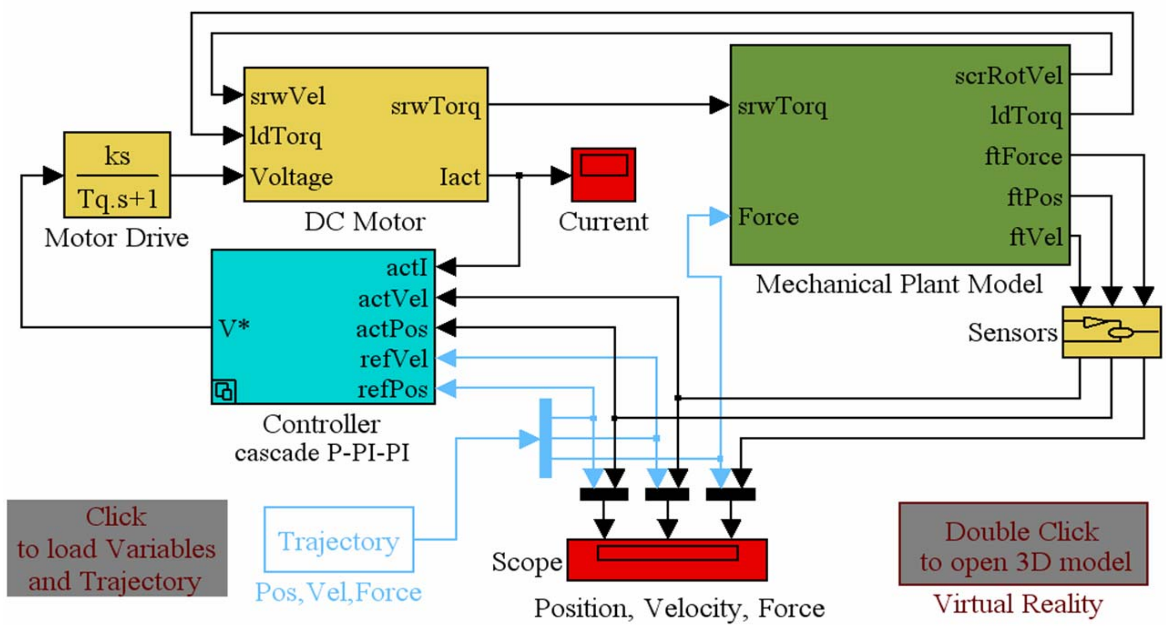

Fig. 7 An example of a modular control system model

3.2 Real-time Programming Software

In the past, the programming languages used for implementing real time embedded systems were basically assembler and/or C. Today, however, there have emerged special programming languages called graphical development environments, which have typically modeling, simulation, as well as platform specific C-code generation capabilities. These include general purpose high level modeling and simulation languages such as Matlab/Simulink/Stateflow from the MathWorks [6] and LabVIEW from National Instruments [7], which have integrated C/C++ code generators. Targetlink from dSPACE [8] can, for instance, generate a production code (C/C++ code) straight from Matlab/Simulink/Stateflow graphical development environment. Along with this, model based control design has also become an established development method across many industries. The main advantages of these types of tools are code repeatability and reduction of development time (up to 50% [8]). For example, it is much easier to implement parallelism with high level graphical programming languages such as Simulink and LabVIEW compared to text-based coding languages. There are also other evolving development platforms that primarily focus on how to quickly get developments started. These platforms provide the much needed mechanisms to reuse the software components from one project to another [9].

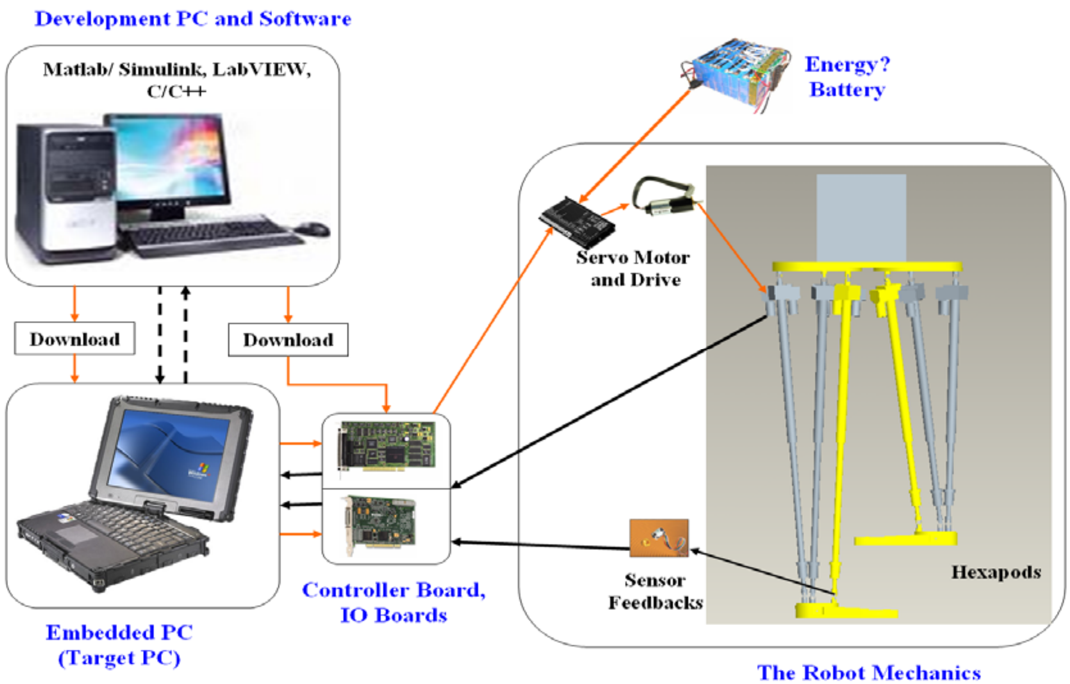

Fig. 8 A schematic example showing a development system configuration for code generation, deployment, and virtual prototyping of embedded control system.

3.3 Virtual Testing and Simulation

Here, special attention is to be placed on multi-domain simulation of the system with the goal of implementing simulated real time operation. Since a control performance of a system is limited by the model accuracy, a great portion of the design effort should be spent on modeling. Virtual prototyping can involve different functionalities ranging from planning of motion trajectories and collision detection using 3D-CAD tools to determining throughput time as well as sizing of actuators. In virtual testing of motion trajectories, for instance, each motion axis can be mapped to a joint of a 3D-CAD model. Then by animating 3D system models, designers can quickly evaluate the feasibility of the overall conceptual design in machine design process. Apart from that, motion profiles can be validate on a 3D-CAD model with the help of collision detection features in 3D-CAD tools before implementing them on a real machine without risking the physical systems.

3.4 Hardware in the Loop Simulation

After testing purely computer simulated system, real hardware components can be integrated step by step into the virtual system. This mixture of virtual and real systems is known as Hardware-in-the-Loop (HiL) simulation or in other cases Software-in-the-Loop (SiL) simulation can also be applied. Today, HiL simulation is common but relatively limited in control design. Extension of the HiL technology to design and testing of general mechatronic systems and components can have huge benefits to the design of future. Fig. 9 below shows schematic setup for virtual testing and implementation approach to be used in Centaurob development.

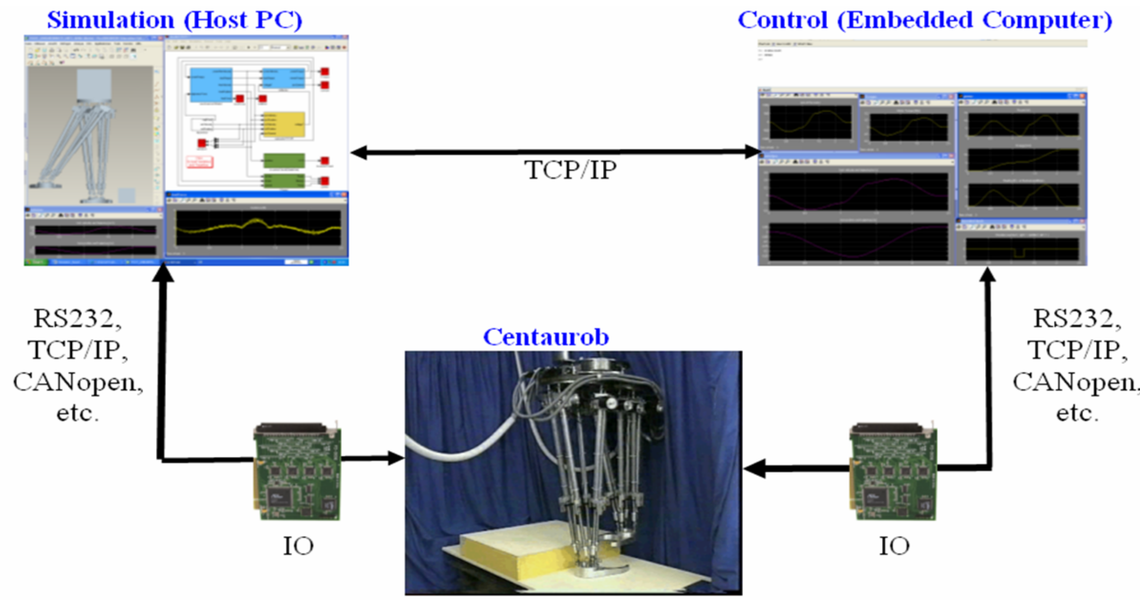

Fig. 9 A schematic system setup for virtual testing, and implementation of Centaurob

4 Components Specification and Selection

The system components involved here are mainly the motor drive electronics, sensors, communication buses, controller computers, IO boards, and embedded software systems. These components should be carefully evaluated and selected on the bases of the control algorithms, control architecture, and implementation strategies described in previous sections. The full details of the systematic evaluation and selection methodologies can not be presented in this short paper but there are effective domain independent methodologies, which can be conveniently used in the evaluation and selection of commercial of-the-shelf components [2, 10, and 11]. Components selection parameters are many and highly interdependent in complex systems like Centaurob. However, there are some general criteria that can be used in modern electronic system specification, evaluation, and selection e.g. complexity, interoperability, technology trend, ease of development, performance and thermal management [12], size or miniaturization of components [13], cost, availability, etc.

Motor Dimensioning and Selection: - In selecting motors one takes into account evaluation criteria such as power behavior, electrical behavior, controlling possibilities etc. in addition to mechanical and thermal characteristics. One of the most important motor parameter from control design point of view is, however, the art and means of sending control commands to the motors. This plays a significant role in selecting a controlling computer system, especially the controller input/outputs (IOs).

Sensor Selection: - Sensors are by nature very diverse system components both functionally and constructionwise. Depending on the type and application of a given sensor, various parameters play a role in its selection and evaluation. Size or dimension, orientation, direction of movement, portability, compatibility with other hardware and software components, analog or digital, types of interfaces (RS232, PWM, CAN, etc), incremental or absolute, power supply, wear and tear, ease of installation, ease of calibration, measurement

ISBN 978-1-60643-379-9

range, measurement frequency, resolution, accuracy, dynamic behavior, operating environment, signal sensitivity, noise sensitivity, price, availability, technological trend, etc. could all play a role to some extents. Thorough understanding of these parameters is required in order to be able to make informed tradeoffs among them in the selection process. Here, one may follow systematic evaluation methodology by ranking different sensor solutions against weighted evaluation criteria [10]. Various trends in sensor technology may also be used as a guide in searching modern and future oriented sensor solution, such as:

- USB 2.0 data acquisition (Laptop based).
- Improved performance and declining prices of ADCs [14],
- Onboard Field-Programmable Gateway Arrays (FPGA) [15],
- New developments in MEMS technology e.g. MEMS accelerometers [16].
- Wireless technologies e.g. WiFi, WLAN, Bluetooth, ZigBee, RFID, etc. [17]

5 Challenges in Electronic Control Hardware and Software Selections

From the discussions above, one can observe how complex a control system can grow if not managed systematically. The selection process gets especially more complex, when the components come from different vendors and have differing development software environments. In some cases, the system could be built completely from off-the-shelf components, while in other cases, components may have to be designed from a scratch and built by own-self and then integrated to operate smoothly into the system. Here, make or buy decision have to be made based on a number of technical and economic criteria. Many times, incompatible operating concepts and data formats hinder smooth integration of components and the workflow as a whole. In addition to massive time invested in converting data from one format to another, there is also the risk of error introduction due to repeated conversions. So what are the basic challenges and how do we tackle them?

5.1 Problem of complexity

Two major reasons among others can be cited for the growing complexity of modern electronic control-hardware systems. One reason is the increased functionality of the hardware from manufacturer’s side, which is made possible due to dynamic innovations in state of the art electronics. The other reason is differing system environments on the user’s side (e.g. differing digital field bus systems, differing host systems, different PC supported tools, etc.). Implementation of components from different manufactures can get very complex and may involve huge data flooding. Frequently, users are forced to spend more of their valuable time to solve problems created by new innovations and/or software releases by producers than doing the “real intended job”. Complexity of a system should, therefore, be kept as low as possible. By doing that one can understand the system better and as a result the design gets less prone to human error. Simulation of a system is a proven method in fighting complexity because it increases understandability.

5.2 Problem of Compatibility/Interoperability

Performance and fitness of each individual component of the control hardware system can be dependent on other hardware components in the system. Moreover, it may depend on development software used. Therefore, hardware selection process of electronic components should be made considering both the hardware and software environments under which it is to be developed and operated. Otherwise, interfacing process for mismatched hardware-software components can be painfully complex, costly, and time consuming. Using hardware and software components from a single vendor can significantly reduce incompatibility problem. But a single source system is usually less flexible and leads to more dependence on the vendor. Moreover, some component parts from that vendor may not necessarily be the best in the market. The dilemma here is, therefore**,** *whether* to adapt a complete compatible system from a single source [13], which is possibly less flexible, sub-optimal, and source dependent, *or* to put together functionally optimal components from different sources and go the risk of interfacing complexity [18]. The industry response to the compatibility problem could be a creation of common industry-platforms (e.g. AUTOSAR in automotive industry [19]). If there were an industry-wise common platform, most of the problems in interfacing, compatibility, modularity, etc. would have been mitigated. If we consider that every component plugs and plays with every other component, the system design and integration would have been an easier task, but most probability at the expense of innovativeness.

5.3 Recommendations and Observations

This paper has explored multi-dimensional components selection process and strategies with the goal of optimizing control-hardware-software selection for a complex mechanical application. In this top-down approach, a general format used in preparing requirements and specification of control hardware and software components can be of great help. Control system requirements specification involves identification and detailed understanding of the parameters relevant to electronic-components design and/or selection. Parameters used in requirement specification should correspond to parameters used in manufacturer's component data sheets. They should also be made understandable for both mechanical and electrical designers. It is not uncommon to find identical terms and abbreviations used by specialists in different fields with differing meanings. Vendor documentation of electronic or software components are usually bulky and they are intermingled with advertisements. Performance indicators in the vendor's documentation are usually "the best case scenarios", and may not be delivered in real application environment. Moreover, the terminologies and even performance parameters used by different producers in the specification of seemingly comparable components can vary and get confusing. Therefore, requirements specification should take these into consideration.

Computer simulation is an invaluable tool in system specification and components selection process. Especially virtual prototyping and hardware-in-the loop simulation are highly recommended because they bring theory closer to reality early in the design process. But, on the other hand, since pure computer simulation can not substitute physical experimentation, one should be careful when using numerical output from simulation models. Another important issue to consider here is the 'time cost' of building complex and detailed computer simulation of a system i.e. some simulations may simply not be worth the time and money spent on them.

6 Conclusion

In spite of serious challenges involved in finding a fitting hardware-software solution for a given mechanical application, it is difficult to find comprehensive research works on optimization of the selection and integration process of electronic components in this area. Most of the works available are either tailored to specific types of applications, specific types components, and/or manufacturer or they are incomplete or in other cases too general. Catalogs get out of date too soon and too often in this highly innovative and dynamic system.

This paper explores some important design techniques and tools applicable in mechatronic design process. The discussed approach helps to get a clearer view of the design problems from a function oriented interdisciplinary point of view. It investigates the state of the art methods through which the mechatronic designer can cross the boundaries of technical disciplines and optimizes the interdisciplinary system components, thereby improving the overall system performance. The paper also presents a case study, on the example of a biped robot, where it takes the design problem from the abstract mechanical structure level step by step down to the late electronic control components selection level. In the Centaurob application, a PC-based virtual testing and hardware-in-the loop simulation methods are being experimented. Here, Matlab/Simulink/Stateflow graphical development tools, dSPACE control hardware, and compatible IO boards from National Instruments are used to interact with various sensors and DC Servomotors. This approach is efficient and very effective in step-by-step integration of control components as they arrive. In the oral presentation of the article, some of the simulation and prototyping results of the experimentation along with the hardware and software technologies used will be presented and discussed.

References

[1] J. Schlattmann, Mekonnen Tesfay Tesfu, Lars Ziemen. New Steps in the Development of the Two Legged Robot CENTAUROB. International Journal of Modern Engineering (IJME), Volume 7, Number 2, Spring 2007.

[2] Ehrlenspiel, K.; Kiewert, A.; Lindemann, U. Kostengünstig Entwickeln und Konstruieren. Berlin, Springer 2002.

[3] http://zone.ni.com/devzone/cda/tut/p/id/6183, June, 2008

[4] Gerd Frick. Eine funktionale, synchrone programmiersprache fur eingebettete Ectzeitsysteme. FZI Forschungszentrum Information an der Universität Karlsruhe. FZI-Publikation 2/2006.

[5] Daniel Witsch, Birgit Vogel-Heuser. Modeling Approach for time requirements and Communication Networks. Atp - Automatisierungstechnische Praxis, 6/2008:44–52, 2008.

[6] http://www.mathworks.com/products/simulink/. June 2008.

[7] http://www.ni.com/labview85/. June 2008.

[8] dSPACE Catalogue 2008.

[9] Robert Cravotta. Robots on the March. EDN Europe, Issue 03/2008

[10] Mekonnen Tesfay Tesfu. Investigation of Optimization Strategy for Design Component-Selection. Master-Thesis, not published, Hamburg University of Technology, 2005.
[11] M. Meß, J. Schlattmann. Auswahl und Integration existierende elektronischer Komponenten mechatronischer Systeme am Beispiel von Sensoren. VDI-Berichte Nr. 1892, 2005.
[12] Wolfgang Scheel, Martin Schneider-Ramelow. Optimized Methods to cool down Electronic Systems. Productronica Magazine, B 19126 Nov.2007, 2007.
[13] Wolfgang Binder, Matthias Geiger. Approach and Implementation Options for SiP Assemblies; Genuine Miniaturization. Productronica Magazine, B 19126 Nov.2007, 2007.
[14] F. Mayer-Lindenberg. Dedicated Digital Processors; methods in hardware/software system design. John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England, Copyright @ 2004.
[15] http://www.ni.com/fpga_technology/. June 2008. FPGA (National Instruments)
[16] Bob Scannell. Integrierte MEMS-Sensoren für Industriesteuerungen. Elektronik Praxis Nr.21, 7.November 2007.
[17] Heinrich Milosiu. Low-current solutions for Wireless Point-to-Point Sensor Data Transmission for Long-term Application. Frauenhofer-Institut für Integrierte Schaltungen; Atp - Automatisierungstechnische Praxis, 3/2008:51–54, 2008.
[18] Mike Hertz. Programmierbare Oszilloskope; Eigene Funktionen Implementieren. Elektronik Praxis Nr.21, 7.November 2007.
[19] http://www.autosar.org, June, 2008

Biography

MEKONNEN TESFAY TESFU (MSc) is research assistant working on “systematic development and integration of control systems into dynamic mechanical systems” in workgroup System Technologies and Design Methodology (AmP) at the Hamburg University of Technology. He earned his B.Sc. in Mechanical Engineering from Addis Ababa University and MSc. in International Production Management from Hamburg University of Technology: Mekonnen.Tesfay@Tuhh.de

Prof. Dr. E.h. Prof. h.c Dr.-Ing. JOSEF SCHLATTMANN is head of the workgroup System Technologies and Design Methodology at the Hamburg University of Technology. He received several honorable research awards and is the originator of numerous patents. j.schlattmann@tuhh.de

Dipl.-Ing. LARS ZIEMEN received his diploma in Mechanical Engineering in 2005 at the Hamburg University of Technology. He is currently working as research assistant on design and control of mechanical systems for service robots at the AmP. lziemen@tuhh.de