

## AN UNDERACTUATED MODULAR ROBOT FOR TESTING CONTROL ALGORITHMS

*Ambrus Zelei*

HAS-BUTE Research Group on  
Dynamics of Machines and Vehicles  
Muegyetem rkp. 5. Budapest, H-1111, Hungary  
zelei@mm.bme.hu

*László Bencsik and Gábor Stépan*

Department of Applied Mechanics  
Budapest University of Technology and Economics  
Muegyetem rkp. 5. Budapest, H-1111, Hungary  
bencsik@mm.bme.hu, stepan@mm.bme.hu

### ABSTRACT

In this paper we propose a concept of a modular robot that can mainly be used for testing control algorithms in research work and in education. In our research work the main purpose is the experimental testing of computed torque control algorithms of underactuated dynamical systems, which are modeled by non-minimum set of descriptor coordinates. Various control methods are known being able to handle these type of control systems. Before the application of these methods on complex robotic structures experimental testing on simpler robotic systems is also needed additionally to the simulational tests.

The benefit of the development of a customized modular manipulator is to get a flexibly programmable and reconfigurable robot which can be built up in several various architectures. The modular structure makes possible to build up different configuration robots, including serial and closed kinematic chain robots. The design and tune of a control strategy for complex and often underactuated systems require practical knowledge and experience of sensing, actuating and data processing based on computers. This predictably low price application makes possible to investigate the control of a serial, parallel or hybrid kinematic chained robot in real environment for educational or research purposes. An other important task is the portability and compatibility, which means that the system will be able to be mobilized easily and to be controlled by any PC using commercial softwares like the Matlab.

**Index Terms**— Modular robot, underactuated robotics, computed torque control, computed desired computed torque control

---

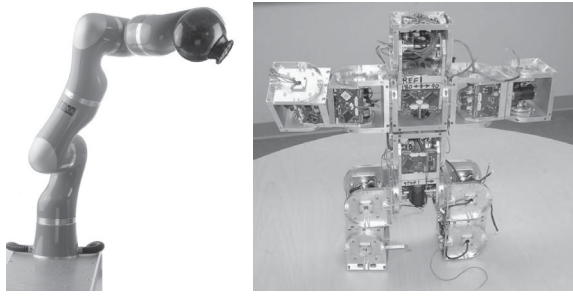
This work was supported by the Hungarian National Science Foundation (Grant No. OTKA K068910), by the HAS-BME Research Group on Dynamics of Machines and Vehicles and by the Oktatásert Kozalaptivany (Grant No. NTP OKA XXII-111)

### 1. INTRODUCTION

Modular robots usually builds up of multiple building parts of a relatively small variety, with uniform docking interfaces. The interfaces allow transfer of mechanical forces and torques, electrical power, and communication throughout the robot. The modular building blocks often consist of some primary structural actuated unit and potentially some additional specialized units such as grippers, wheels, cameras, etc. [1].

Modular robotic systems can be divided into the family of not self-reconfigurable and self-reconfigurable systems. The self-reconfigurable systems provides the capability of doing a wide variety of tasks because of their effectively changing topology.

Self-reconfigurable robot systems appeared first in the late 1980s when the concept of the common connection interface was applied in the CEBOT (short for cellular robot) modular robotic system [2]. Since then several self-reconfigurable modular robotic systems have been appeared. The modular transformer (MTRAN) series [3] incorporates the advantages of hybrid chain and lattice system [1]. The Miche system [4] has been developed at MIT. Each modules of this modular lattice system is an autonomous robot cube capable of connecting to and communicating with its immediate neighbors. The group of modules can be assembled into a complex structure with the help of distributed control algorithms. PolyBot [5] chain self-reconfiguration system was created at Palo Alto Research Center (PARC). Each cubic shaped module has one rotational degree of freedom. PolyBot has demonstrated many modes of locomotion including: biped walking, snake-like locomotion, climbing, etc. An other system with mechanically similar modules is Molecube system developed at Cornell University [6], built to physically demonstrate kinematic self-reproduction. The theoretical existence of arbitrarily sized self-replicating machines has been mathematically demonstrated. The SuperBot [7] seen in Fig.1 has been developed at the Uni-



**Fig. 1.** KUKA light weight robot (left), SuperBot from the University of Southern California (right)

versity of Southern California as a deployable self-reconfigurable robot for real-world applications outside laboratories. Its modules have a hybrid chain and lattice architecture.

Not self-reconfigurable systems forms the other large group of modular robots. The goal of the development of such systems is to obtain the possibility of constructing large variety of configurations mainly for research and educational purposes. Light-weight robots usually fits to this goal and generally consist of unified building parts. The DLR Institute of Robotics and Mechatronics, designed several light-weight robotic systems: DLR Light-Weight Robot I, II and III [8, 9]. DLR's light-weight robots are composed by unified modules and have an outstanding ratio of payload to total mass. These systems are mainly applied in university research work. The light-weight robot designed by the KUKA Robotics company [10] is also interesting for researchers in the field of robotics, and besides the system is also important in the industry. The contribution to the research into new fields of application for robotics co-operating with universities is aimed by the KUKA Robotics company keeping in focus the recently designed modular structured light-weight robot. An other basically modular robotic system in the industry is the modular robotic arm designed by the Robotnik company [11]. The modular arm includes modular servo-actuators composed by a motor and an attached gearbox. A power supply unit and a controller is also integrated in the system, thus, as an important feature, the modular arm does not need an external control unit. Hence, the communication between the arm and the environment is reduced to minimal: 2 wires for communication and 2 for power.

Because of the criteria explained in the following sections none of the above mentioned systems is not suitable for our purposes. In this work we design a manually reconfigurable modular robotic system for research and educational purposes. The main goal is the experimental testing of computed torque control algorithms of underactuated dynamical systems, which are modeled by non-minimum set of descriptor coordinates.

## 2. THE CONTROL ALGORITHMS AIMED TO TEST EXPERIMENTALLY

This section summarizes the control problems planned to be studied experimentally. The mechanical structure of the proposed modular system is polarized to be as suitable as possible for these control algorithms.

### 2.1. Overview

In general computed torque control (CTC) method can be used if the given trajectory of the end effector of the robot has to be followed with minimal deviation. The CTC method requires an accurate dynamical model and its inverse kinematics and dynamics [12]. The accurate following of a prescribed trajectory is a typical demand e.g. in industrial robotic systems, surgical systems [13] or in the case of domestic robots such as the ACROBOTER system [14].

In our research work we apply the CTC method for underactuated dynamical systems. Generally a system is underactuated if the rank of the input matrix is smaller than the number of degrees of freedom (DoF). Thus, if a dynamical system has less independent actuators than DoFs, it is underactuated [15].

In practice the controlled dynamical systems may be underactuated as in the case of unmanned aerospace and under-water vehicles [16]. The elasticity of the mechanical parts of a controlled dynamical system also can be handled as an underactuated problem [17]. Several specially designed robotic systems are also underactuated such as the ACROBOTER service robot [14].

In general the application of the computed torque control leads to a differential algebraic equation (DAE) problem [18, 19] because the generalized coordinates of the system as differential variables and the control inputs as algebraic variables are to be calculated from the equations results from the joined inverse dynamical and kinematical calculation.

A basic aim of our research is to apply the CTC method for underactuated multibody systems. Multibody systems, especially those which contain closed kinematic chain, cannot be efficiently modeled in the most common way, when a minimum set of generalized coordinates is chosen. Instead, redundant set of descriptor coordinates can be used with geometric constraints, in order to avoid numerically expensive computations [20]. Because of the geometric constraints between the redundant coordinates, algebraic equations arise in the resulting equation of motion.

If we consider the above mentioned phenomena we can conclude that the application of the CTC method for underactuated systems modeled by redundant coordinates is a challenging task because the control law is formulated in the form of a DAE [17, 18, 19, 21, 22]. Several methods exists for the solution of the problem which led us to develop a modular robot, on which the

computed torque control algorithms for underactuated dynamical systems modeled by non-minimum set of coordinates can be experimentally tested.

## 2.2. Problem formulation with redundant coordinates

The CTC method for underactuated systems can be generalized for systems modeled by non-minimum set of coordinates. In such case the dynamical model can be written in the form of a differential algebraic equation, which has the following general form [15, 20]:

$$\begin{aligned} \mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \Phi_{\mathbf{q}}^T(\mathbf{q})\lambda &= \mathbf{Q}(\mathbf{q}) + \mathbf{H}(\mathbf{q})\mathbf{u}, & (1) \\ \phi(\mathbf{q}) &= \mathbf{0}, & (2) \end{aligned}$$

where  $\mathbf{M} \in \mathbb{R}^{n \times n}$  is the constant massmatrix,  $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^n$  is the vector of the forces arise from the dynamics of the system, and  $\Phi_{\mathbf{q}}(\mathbf{q}) = \partial\phi(\mathbf{q})/\partial\mathbf{q} \in \mathbb{R}^{m \times n}$  is the constraint Jacobian associated with the geometric constraints  $\phi(\mathbf{q}) \in \mathbb{R}^m$ .  $\mathbf{Q}(\mathbf{q}) \in \mathbb{R}^n$  is the vector of gravitational forces.  $\mathbf{H}(\mathbf{q}) \in \mathbb{R}^{n \times l}$  is the control input matrix and  $\mathbf{u} \in \mathbb{R}^l$  is the control input vector. We assume that the dimension of the control input  $l$  is less than the degrees of freedom  $n - m$ .

The inverse kinematical and dynamical calculations have unique solution if the number of control inputs and the dimension of the task is equal [17]. Thus we assume that the task is defined by  $l$  number of algebraic equations. This set of additional constraint equations are the so-called servo-constraints (control-constraints)  $\phi_s(\mathbf{q}, t) \in \mathbb{R}^l$ :

$$\phi_s(\mathbf{q}, t) = \mathbf{0}. \quad (3)$$

We assume that the servo-constraints can be satisfied with bounded control forces.

In special cases we can assume that these servo-constraint equations can be written in the following form:

$$\phi_s(\mathbf{q}, t) = \mathbf{g}(\mathbf{q}) - \mathbf{p}(t), \quad (4)$$

where  $\mathbf{g}(\mathbf{q})$  represents, for example, the end-effector position of the robot as the function of the descriptor coordinates and  $\mathbf{p}(t)$  is an arbitrarily prescribed function of time expressing the performance goal to be realized [17].

In some cases the servo-constraints and a well chosen subset of geometric constraints can be solved for the controlled coordinates  $\mathbf{q}_c$  in closed form. Then the task can be defined by

$$\mathbf{q}_c = \mathbf{q}_c^d, \quad (5)$$

where the superscript  $d$  refers to the desired coordinate value or one can say desired trajectory. In this formulation the controlled coordinates are a prescribed function of time. In such case one can split the descriptor coordinates into controlled and uncontrolled part as:

$$\mathbf{q}_c = \mathbf{S}_c^T \mathbf{q}, \quad (6)$$

$$\mathbf{q}_u = \mathbf{S}_u^T \mathbf{q}, \quad (7)$$

respectively, where  $\mathbf{S}_c$  and  $\mathbf{S}_u$  are task dependent selector matrices. In some of the control methods overviewed in the further sections this simplification can be utilized for saving computational time.

## 2.3. Analytical solution

In simple cases analytical solution may exist for the problem defined in section 2.2. The analytical inverse dynamical calculation of an underactuated crane model was demonstrated in [19]. This reference uses simple algebraic operations to solve the problem, however the application of Laplace-transformation is a more general method for linear systems. After the Laplace-transformation of equations (1), (2) and (3) the Laplace-transform of the control input  $\mathbf{u}$  can be calculated as a solution of a linear algebraic equation system. After that the inverse Laplace-transformation can be applied.

## 2.4. ODE form of the equation of motion

For the application of almost all control theorem, the constraint forces, mathematically the Lagrange multipliers, have to be eliminated from the equation of motion (1).

For this, one choice is the method of Lagrange multipliers [20]. In order to do this one can use the geometric constraints in the level of acceleration by differentiating the constraint equation (2) twice with respect to time:

$$\Phi_{\mathbf{q}}\ddot{\mathbf{q}} + \dot{\Phi}_{\mathbf{q}}\dot{\mathbf{q}} = \mathbf{0}. \quad (8)$$

Substituting (8) into the equation of motion (1) the Lagrange multipliers can be expressed in closed form:

$$\lambda = (\Phi_{\mathbf{q}}\mathbf{M}^{-1}\Phi_{\mathbf{q}}^T)^{-1}(\Phi_{\mathbf{q}}\mathbf{M}^{-1}(\mathbf{Q} + \mathbf{H}\mathbf{u}) + \dot{\Phi}_{\mathbf{q}}\dot{\mathbf{q}}). \quad (9)$$

After substituting (9) back into the equation of motion (1) the acceleration can be expressed directly. However, important to notice that the resulting ODE is unstable thus it cannot be used for simulations only for the calculation of the control input in a definite time instant.

An other possibility to transform the equation of motion into ODE form is the projection of the equation of motion (1) into the proper subspaces [23]. Let us consider the decomposition of the variation of the descriptor coordinate velocity  $\delta\mathbf{q}$  into the admissible  $\delta\mathbf{q}_a$  and the constrained  $\delta\mathbf{q}_c$  vectors:

$$\delta\mathbf{q} = \delta\mathbf{q}_a + \delta\mathbf{q}_c. \quad (10)$$

With the assumption that the geometric constraints does not depend on time explicitly from the time derivative of the constraint equation (2) we can write that

$$\Phi_{\mathbf{q}}\delta\dot{\mathbf{q}} = \mathbf{0}. \quad (11)$$

Besides the constraint Jacobian  $\Phi_{\mathbf{q}}$  is composed by the gradient vectors of the geometric constraints which lead

to the following expression for the variation of the constrained velocities:

$$\Phi_q \delta \dot{q}_c = \mathbf{0}, \quad (12)$$

from which considering (11) one can obtain:

$$\Phi_q \delta \dot{q}_a = \mathbf{0}. \quad (13)$$

The vector  $\delta \dot{q}_a$  appearing in (13) is the difference of two possible velocities admitted by the geometric constraints, so it is the virtual velocity in the classical sense. Moreover we can state that (13) is satisfied if  $\delta \dot{q}_a$  is in the null space of  $\Phi_q$  defined as:

$$\delta \dot{q}_a = \mathbf{P}_a \delta \dot{q}, \quad (14)$$

$$\mathbf{P}_a = \mathbf{I} - \Phi_q^\dagger \Phi_q, \quad (15)$$

where  $\Phi_q^\dagger$  is the Moore-Penrose pseudoinverse of the constraint Jacobian. Finally we gain the equation of the motion admitted by the geometric constraints by using the principle of virtual power and the projection matrix defined by (15):

$$\mathbf{P}_a^T [\mathbf{M} \ddot{q} + \mathbf{C}(q, \dot{q}) - \mathbf{Q}(q) - \mathbf{H}(q)u] = \mathbf{0}, \quad (16)$$

One can observe that the calculation of the pseudoinverse  $\Phi_q^\dagger$  can lead to physically incorrect results depending of the dimensions of the descriptor coordinates in  $q$ . In [23] a modified pseudoinverse calculation was introduced and was used to calculate the projection matrix  $\mathbf{P}_a$ :

$$\tilde{\Phi}_q^\dagger = \mathbf{L}^{-1}(\Phi_q \mathbf{L}^{-1})^\dagger, \quad (17)$$

$$\mathbf{P}_a = \mathbf{I} - \tilde{\Phi}_q^\dagger \Phi_q, \quad (18)$$

where  $\mathbf{L}$  is the Cholesky decomposition of the mass matrix  $\mathbf{M}$ .

The methods that was briefly summarized in this section makes possible to use the general control methods for systems defined with geometric constraints.

## 2.5. Partial feedback linearization

The partial feedback linearization is commonly used in the case of the control of underactuated systems. The main idea of the method is to substitute the original nonlinear system with a partially equivalent linear system by a transformation. The partial feedback linearization can be applied for the systems given in the following form [24]:

$$\dot{x} = f(x) + g(x)u, \quad (19)$$

$$y = h(x), \quad (20)$$

where  $x$  is the state vector of the system,  $u$  is the control input and  $y$  is the output vector. With the partial feedback linearization the control input can be formulated as:

$$u = a(x) + b(x)v, \quad (21)$$

which results a linearized system as a cascade of  $n$  integrators, and a synthetic input  $v$  that can be chosen arbitrarily, e.g. linear compensator [24].

For the application of this theory the Lagrange multipliers have to be eliminated from the equation of motion as section 2.4 explains.

## 2.6. Computed Desired Computed Torque Control method

The CTC method for underactuated systems is already published in [25] for dynamical systems that are modeled by minimum set of generalized coordinates. The generalized method is called Computed Desired Computed Torque Control method (CDCTC), where the expression "computed desired" refers to the fact that the uncontrolled coordinates cannot be prescribed arbitrarily, since they depend on the internal dynamics of the system. In case of the CDCTC method the equations of motion is an ordinary differential equation (ODE) and the null space of the coefficient matrix of the input vector is used to project these equations into the space of uncontrolled motions. The projected set of differential equations can then be solved for the desired values of the uncontrolled coordinates and the control inputs can then be expressed from the original equation of motion.

For the application of the CDCTC the Lagrange multipliers have to be eliminated from the equation of motion as section 2.4 explains.

## 2.7. Method of Lagrange multipliers with servo-constraint stabilization

In this approach the servo-constraints are handled similarly to the geometric constraints. The geometric constraints can be expressed in acceleration level as equation (8) shows. Similarly the servo-constraint equation (3) also can be differentiated twice with respect to time:

$$\mathbf{G}_q \ddot{q} + \dot{\mathbf{G}}_q \dot{q} + \dot{c} = \mathbf{0}, \quad (22)$$

where  $\mathbf{G}_q \in \mathbb{R}^{l \times n}$  is the Jacobian of the servo-constraint and  $c \in \mathbb{R}^l$  is the time derivative of the explicitly time dependent part of the servo-constraint. In the application of the method of Lagrange multipliers the geometric constraint equations are stabilized by the Baumgarte method [20, 26]. Similarly, here we extend the acceleration level servo-constraint equation (22) as follows:

$$\mathbf{G}_q \ddot{q} + \dot{\mathbf{G}}_q \dot{q} + \dot{c} + \mathbf{K}_D(\mathbf{G}_q \dot{q} + c) + \mathbf{K}_P \phi_s = \mathbf{0}, \quad (23)$$

The equation of motion (1), the acceleration level geometric constraint equation (8) and equation (23) can be incorporated in hyper-matrix form as follows:

$$\begin{bmatrix} \mathbf{M} & \Phi_q^T & -\mathbf{H} \\ \Phi_q & \mathbf{0} & \mathbf{0} \\ \mathbf{G}_q & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{q} \\ \lambda \\ u \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_g - \mathbf{C} \\ -\dot{\Phi}_q \\ -\dot{\mathbf{G}}_q \dot{q} - \dot{c} - \mathbf{K}_D(\mathbf{G}_q \dot{q} + c) - \mathbf{K}_P \phi_s \end{bmatrix}, \quad (24)$$

from which the control input can be calculated as the function of the measured state  $q$  and  $\dot{q}$  of the system. It has to be noticed that the coefficient hyper-matrix of the unknowns  $\ddot{q}$ ,  $\lambda$  and  $u$  is not invertible if the system is non-collocated. The definition of collocated and non-collocated underactuated systems can be found in [25].

## 2.8. Direct discretization

This method is similar to the CDCTC method summarized in section 2.6 from the viewpoint that the desired coordinates are also calculated. In this method we apply the backward Euler discretization of the DAE system directly and the resulting set of nonlinear algebraic equations are solved by the Newton-Raphson method for the desired actuator forces, uncontrolled coordinates and Lagrange multipliers [19, 22].

In the present formulation we assume that the servo-constraints with the geometric constraints can be solved for the controlled set of descriptor coordinates  $\mathbf{q}_c$ , see equations (5), (6) and (7). Considering a PD controller with gain matrices  $\mathbf{K}_P$  and  $\mathbf{K}_D$  the control law can be formulated as:

$$\mathbf{M}\ddot{\mathbf{q}}^d + \mathbf{C}(\mathbf{q}^d, \dot{\mathbf{q}}^d) + \Phi_{\mathbf{q}}^T(\mathbf{q}^d)\boldsymbol{\lambda} = \mathbf{Q}(\mathbf{q}^d) + \mathbf{H}(\mathbf{q}^d)\mathbf{u} + \mathbf{K}_P(\mathbf{q} - \mathbf{q}^d) + \mathbf{K}_D(\dot{\mathbf{q}} - \dot{\mathbf{q}}^d), \quad (25)$$

$$\boldsymbol{\phi}(\mathbf{q}^d) = \mathbf{0}. \quad (26)$$

Introducing  $\mathbf{y}^d = \dot{\mathbf{q}}^d$  we derive the first order form of (25). After the decomposition of the controlled and uncontrolled coordinates the control law can be written as:

$$\dot{\mathbf{q}}_c^d = \mathbf{y}_c^d, \quad (27)$$

$$\dot{\mathbf{q}}_u^d = \mathbf{y}_u^d, \quad (28)$$

$$\mathbf{y}_c^d = \mathbf{S}_c^T \mathbf{M}^{-1} [-\Phi_{\mathbf{q}}^T(\mathbf{q}^d)\boldsymbol{\lambda} - \mathbf{C}(\mathbf{q}^d, \dot{\mathbf{q}}^d) + \mathbf{Q}(\mathbf{q}^d) + \mathbf{H}(\mathbf{q}^d)\mathbf{u} + \mathbf{K}_P(\mathbf{q} - \mathbf{q}^d) + \mathbf{K}_D(\dot{\mathbf{q}} - \dot{\mathbf{q}}^d)], \quad (29)$$

$$\mathbf{y}_u^d = \mathbf{S}_u^T \mathbf{M}^{-1} [-\Phi_{\mathbf{q}}^T(\mathbf{q}^d)\boldsymbol{\lambda} - \mathbf{C}(\mathbf{q}^d, \dot{\mathbf{q}}^d) + \mathbf{Q}(\mathbf{q}^d) + \mathbf{H}(\mathbf{q}^d)\mathbf{u} + \mathbf{K}_P(\mathbf{q} - \mathbf{q}^d) + \mathbf{K}_D(\dot{\mathbf{q}} - \dot{\mathbf{q}}^d)], \quad (30)$$

$$\mathbf{0} = \boldsymbol{\phi}(\mathbf{q}^d). \quad (31)$$

Equation (27) is identity because the controlled coordinates are prescribed thus it can be left out from the equation set. After the backward Euler discretization of equations (28-31) we obtain a system of  $2n - l + m$  number of nonlinear algebraic equations for the  $i$ -th value of the desired uncontrolled coordinates  $\mathbf{q}_{u,i}^d$ , their time derivatives  $\mathbf{y}_{u,i}^d$ , the control inputs  $\mathbf{u}_i$  and the Lagrange multipliers  $\boldsymbol{\lambda}_i$ . It can be formulated as a function  $\mathbf{F}(\mathbf{z}_i)$  of the vector of unknowns  $\mathbf{z}_i$ :

$$\mathbf{z}_i = [\mathbf{q}_{u,i}^d, \mathbf{y}_{u,i}^d, \mathbf{u}_i, \boldsymbol{\lambda}_i]^T. \quad (32)$$

The system of nonlinear algebraic equations is solved by Newton-Raphson method. The  $j$ -th approximation of the unknowns in the  $i$ -th time step can be formulated as:

$$\mathbf{z}_i^j = \mathbf{z}_i^{j-1} - \mathbf{J}^{-1}(\mathbf{z}_i^{j-1})\mathbf{F}(\mathbf{z}_i^{j-1}), \quad (33)$$

where  $\mathbf{J}(\mathbf{z}_i)$  is the Jacobian of  $\mathbf{F}(\mathbf{z}_i)$ . Usually Newton-Raphson iteration gives accurate result in very few steps because the initial estimation  $\mathbf{z}_i^0$  comes from the solution  $\mathbf{z}_{i-1}$  calculated in the previous time step.

The calculation of the Jacobian can be accomplished analytically and also numerically. In order to save computational time it is enough to calculate the Jacobian one or maximum two times in each time step.

In some cases the Jacobian matrix may be ill-conditioned, but the problem can be handled by singular value decomposition.

## 3. THE DESIGN OF THE MODULAR ROBOT

We propose the design of a flexibly programmable and manually reconfigurable modular robotic system. The main requirements the system has to meet are summarized as:

- The robot has to be flexibly reconfigurable including the possibility to build serial and parallel kinematic chain manipulators.
- The system has to be able to form underactuated systems, which means that active and passive joints have to be replaceable easily. The passive joint have to provide the rotation with minimal friction and the measurement of the joint angle at the same time.
- The inertial forces in the system has to be significantly larger that the friction forces arising by the actuators.
- The centralized control programme of the robot is running on an external PC with which the robot communicates via a standardized interface. This provides the possibility to use commercial softwares to the control algorithm development, e.g. MATLAB, LabVIEW, etc.
- The system has to be able to be extended and replicated easily with unified building parts.
- The hardware has to be developed and remanufactured within relatively low budget. It is important in educational purposes.

In order to test the control methods explained in section 2 these requirements has to be fulfilled. The systems overviewed in section 1 do not meet with all of the above requirements at the same time. It is partially obvious for self-reconfigurable robotic systems because the aim of the development of those systems is not for testing of control algorithms in general. This led us to design a custom robotic system.

The robot builds up by unified modular components. The modular structure allows to build up different configuration robots, including serial (see Fig.3) and closed kinematic chain robots (see Fig.4). Underactuated configurations also can be set up as Fig.3 shows.

The links of the robot are connected to each other by rotation modules, which are driven by motors or employed as free joints where encoders are placed. The rotation module can be seen on Fig.2. Motors (M1 on Fig.3 and M1, M2, M6 on Fig.4) and encoders (E1 on Fig.3 and E3, E4, E5 on Fig.4) can be attached to the hollow shaft of the rotation module. The design of the rotation module and the bar even allows 90° angular deflection between the joint axes as shown on Fig.3.

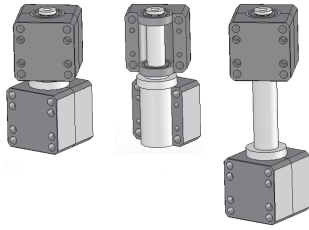


Fig. 2. Rotation module

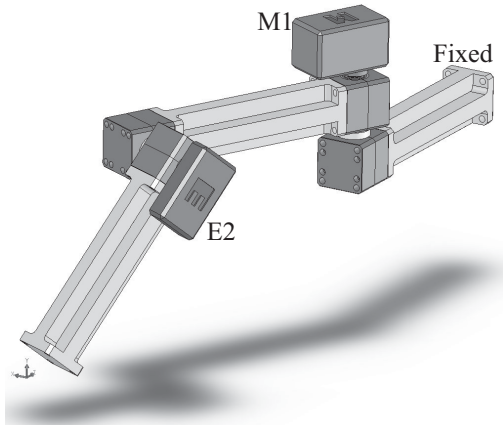


Fig. 3. An open kinematic chain underactuated configuration

The block diagram of the control architecture is shown in Fig.5) for the case of two motors and two encoders, however several motors and encoders could be handled. The control unit communicates with a computer, where the main control algorithm runs. The task of the control unit is to transfer the digital control signal from PC to the motors applying pulse width modulation (PWM). The measured encoder signals are transferred to the computer in digital form. So the control unit integrates the motor and encoder drivers with several input and output interface. Besides the power supply is also provided by the same hardware unit. It provides power supply for the motors, encoders and control unit.

#### 4. CONCLUSION

A modular robot design was proposed to provide an experimental tool for testing different control algorithms in university research work. The developed system is also useful in education to study basic control problems and higher level control methods experimentally.

The computed torque control algorithms applicable for underactuated dynamical systems, which are modeled by non-minimum set of descriptor coordinates were overviewed. The requirement for a robotic system which is suitable for the testing of these algorithms was summarized. The state of the modular robotic sys-

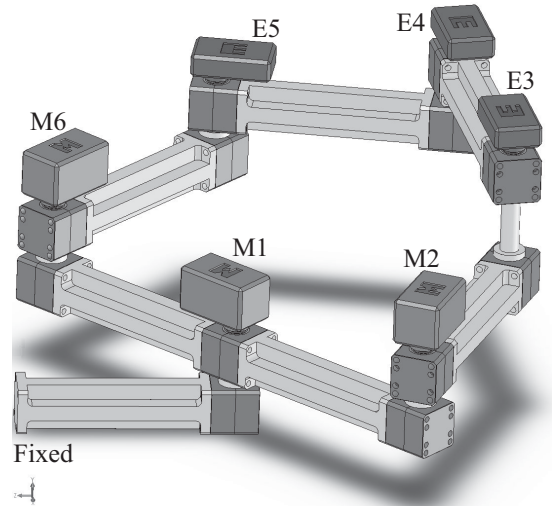


Fig. 4. A closed kinematic chain fully actuated configuration

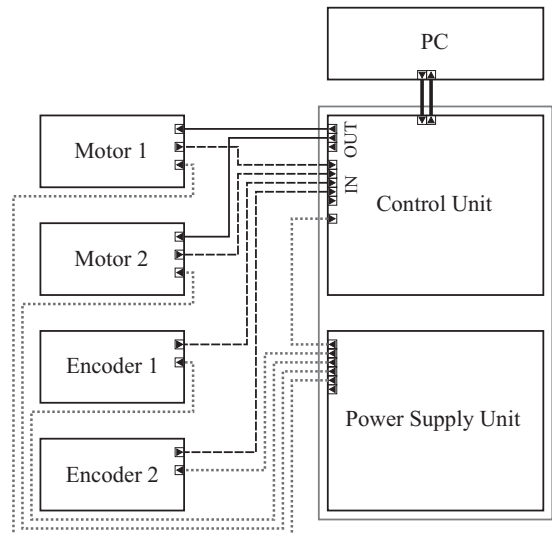


Fig. 5. Centralized control architecture

tems also was overviewed, and we concluded that the desing of a new low cost system is beneficial based on the state-of-the art modular robotic systems.

#### 5. REFERENCES

- [1] M. Yim, Wei-Min Shen, B. Salemi, D. Rus, M. Moll, H. Lipson, E. Klavins, and G.S. Chirikjian, "Modular self-reconfigurable robot systems," *Robotics and Automation Magazine, IEEE*, vol. 14, no. 1, pp. 43–52, 2007.
- [2] T. Fukuda, S. Nakagawa, Y. Kawauchi, and M. Buss, "Self organizing robots based on cell structurescebot," in *Proceedings of IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 1988, pp. 145–150.

- [3] S. Murata, E. Yoshida, A. Kamimura, H. Kurokawa, K. Tomita, and S. Kokaji, "M-tran: Self-reconfigurable modular robotic system," *IEEE/ASME Trans. Mech.*, vol. 7, no. 4, pp. 431–441, 2002.
- [4] D. Rus and G.S. Chirikjian, "Autonom. Robots (Special Issue on Modular Robots)," vol.10, no.1, 2001.
- [5] M. Yim, Y. Zhang, and D.G. Duff, "Modular robots," *IEEE Spectr*, vol. 39, no. 2, pp. 30–34, 2002.
- [6] V. Zykov, E. Mytilinaios, M. Desnoyer, and H. Lipson, "Evolved and designed self-reproducing modular robotics," *IEEE Transactions on Robotics*, vol. 23, no. 2, pp. 308–319, 2007.
- [7] B. Salemi, M. Moll, and W.-M. Shen, "Superbot: A deployable, multi-functional, and modular self-reconfigurable robotic system," in *Proceedings of IEEE/RSJ Intl. Conf. Intelligent Robots Systems*, 2006, pp. 3636–3641.
- [8] G. Hirzinger, A. Baader, R. Koeppe, and M. Sched, "Towards a new generation of multisensory light-weight robots with learning capabilities," in *Proc. of IFAC93 World Congress*, 1993, pp. 18–23.
- [9] G. Hirzinger, J. Butterfaß, M. Fischer, M. Grebenstein, M. Hähnle, H. Liu, I. Schäfer, and N. Sporer, "A mechatronics approach to the design of light-weight arms and multifingered hands," in *Proceedings of the 2000 IEEE International Conference on Robotics and Automation (ICRA)*, 2000, pp. 46–54.
- [10] KUKA Robot Group, "Light weight robot," <http://www.kuka-robotics.com/en/products/addons/lwr/>, Last accessed: July 2011.
- [11] Robotnik, "Modular robotic arm," <http://www.robotnik.es/en/products/robotic-arms/modular-robotic-arm/>, Last accessed: July 2011.
- [12] Mark W. Spong and M. Vidyasagar, *Robot Dynamics and Control*, John Wiley & Sons, 1989.
- [13] da Vinci Surgery, "da Vinci Surgical System," <http://www.davincisurgery.com/>, Last accessed: July 2011.
- [14] G. Stépán et al, "A ceiling based crawling, hoisting and swinging service robot platform," in *Proceedings of Beyond Gray Droids: Domestic Robot Design for the 21st Century Workshop at HCI 2009*, 2009.
- [15] H. Goldstein, *Classical Mechanics*, 2nd ed. Addison-Wesley, 1980.
- [16] Reza Olfati-Saber, *Nonlinear Control of Under-actuated Mechanical Systems with Application to Robotics and Aerospace Vehicles*, Ph.D. thesis, Massachusetts Institute of Technology, 2001.
- [17] W. Blajer and K. Kolodziejczyk, "Modeling of underactuated mechanical systems in partly specified motion," *Journal of Theoretical and Applied Mechanics*, vol. 46, no. 2, pp. 383–394, 2008.
- [18] W. Blajer and K. Kolodziejczyk, "Control of underactuated mechanical systems with servo-constraints," *Nonlinear Dynamics*, vol. 50, no. 4, pp. 781–791, 2007, From the issue entitled "Dynamical Systems: Theory and Applications".
- [19] A. Zelei, L. L. Kovács, and G. Stépán, "Computed torque control of an under-actuated service robot platform modeled by natural coordinates," *Commun Nonlinear Sci Numer Simulat*, vol. 16, no. 5, pp. 2205–2217, 2011.
- [20] J.G. de Jalón and E. Bayo, *Kinematic and dynamic simulation of multibody systems: the real-time challenge*, Springer-Verlag, 1994.
- [21] W. Blajer and K. Kolodziejczyk, "A geometric approach to solving problems of control constraints: Theory and a dae framework," *Multibody System Dynamics*, vol. 11, no. 4, pp. 343–364, 2004.
- [22] L. L. Kovács, A. Zelei, L. Bencsik, J. Turi, and G. Stépán, "Motion control of an under-actuated service robot using natural coordinates," in *Proceedings of ROMANSY 18 Robot Design, Dynamics and Control*, 2010, pp. 331–338.
- [23] J. Kövecses, J.-C. Piedoboeuf, and C. Lange, "Dynamic modeling and simulation of constrained robotic systems," *IEEE/ASME Transactions on mechatronics*, vol. 8, no. 2, pp. 165–177, 2003.
- [24] A. Isidori, *Nonlinear Control Systems*, 2nd ed. Springer, 1999.
- [25] I. M. M. Lammerts, *Adaptive Computed Reference Computed Torque Control*, Ph.D. thesis, Eindhoven University of Technology, 1993.
- [26] J. Baumgarte, "Stabilization of constraints and integrals of motion in dynamical systems," *Computer methods in applied mechanics and engineering*, vol. 1, no. 1, pp. 1–16, 1972.