

ISSN 1590-8844
Vol. 15 No 01
2014

International Journal of Mechanics and Control

Editor: Andrea Manuello Bertetto



LIBRERIA EDITRICE UNIVERSITARIA
LEVROTTO & BELLA
TORINO

Editorial Board of the
International Journal of Mechanics and Control

Published by Levrotto&Bella – Torino – Italy E.C.

Honorary editors

Guido Belforte

Kazy Yamafuji

Editor: Andrea Manuello Bertetto

General Secretariat: Elvio Bonisoli

Atlas Akhmetzyanov
*V.A.Trapeznikov Institute of Control Sciences
of Russian Academy of Sciences
Moscow – Russia*

Domenico Appendino
*Prima Industrie
Torino – Italy*

Kenji Araki
*Saitama University
Shimo Okubo, Urawa
Saitama – Japan*

Guido Belforte
*Technical University – Politecnico di Torino
Torino – Italy*

Bruno A. Boley
*Columbia University,
New York – USA*

Marco Ceccarelli
*LARM at DIMSAT
University of Cassino
Cassino – Italy*

Amalia Ercoli Finzi
*Technical University – Politecnico di Milano
Milano – Italy*

Carlo Ferraresi
*Technical University – Politecnico di Torino
Torino – Italy*

Anindya Ghoshal
*Arizona State University
Tempe – Arizona – USA*

Nunziatino Gualtieri
*Space System Group
Alenia Spazio
Torino – Italy*

Alexandre Ivanov
*Technical University – Politecnico di Torino
Torino – Italy*

Giovanni Jacazio
*Technical University – Politecnico di Torino
Torino – Italy*

Takashi Kawamura
*Shinshu University
Nagano – Japan*

Kin Huat Low
*School of Mechanical and Aerospace Engineering
Nanyang Technological University
Singapore*

Andrea Manuello Bertetto
*University of Cagliari
Cagliari – Italy*

Stamos Papastergiou
*Jet Joint Undertaking
Abingdon – United Kingdom*

Mihailo Ristic
*Imperial College
London – United Kingdom*

János Somló
*Technical University of Budapest
Budapest – Hungary*

Jozef Suchy
*Faculty of Natural Science
Banska Bystrica – Slovakia*

Federico Thomas
*Instituto de Robótica e Informática Industrial
(CSIC-UPC)
Barcelona – Espana*

Furio Vatta
*Technical University – Politecnico di Torino
Torino – Italy*

Vladimir Viktorov
*Technical University – Politecnico di Torino
Torino – Italy*

Kazy Yamafuji
*University of Electro-Communications
Tokyo – Japan*

*Official Torino Italy Court Registration
n.5390, 5th May 2000*

*Deposito presso il Tribunale di Torino
numero 5390 del 5 maggio 2000*

Direttore responsabile:

Andrea Manuello Bertetto

International Journal of Mechanics and Control

Editor: Andrea Manuello Bertetto

**Honorary editors: Guido Belforte
Kazy Yamafuji**

General Secretariat: Elvio Bonisoli

The Journal is addressed to scientists and engineers who work in the fields of mechanics (mechanics, machines, systems, control, structures). It is edited in Turin (Northern Italy) by Levrotto&Bella Co., with an international board of editors. It will have not advertising.

Turin has a great and long tradition in mechanics and automation of mechanical systems. The journal would will to satisfy the needs of young research workers of having their work published on a qualified paper in a short time, and of the public need to read the results of researches as fast as possible.

Interested parties will be University Departments, Private or Public Research Centres, Innovative Industries.

Aims and scope

The *International Journal of Mechanics and Control* publishes as rapidly as possible manuscripts of high standards. It aims at providing a fast means of exchange of ideas among workers in Mechanics, at offering an effective method of bringing new results quickly to the public and at establishing an informal vehicle for the discussion of ideas that may still in the formative stages.

Language: English

International Journal of Mechanics and Control will publish both scientific and applied contributions. The scope of the journal includes theoretical and computational methods, their applications and experimental procedures used to validate the theoretical foundations. The research reported in the journal will address the issues of new formulations, solution, algorithms, computational efficiency, analytical and computational kinematics synthesis, system dynamics, structures, flexibility effects, control, optimisation, real-time simulation, reliability and durability. Fields such as vehicle dynamics, aerospace technology, robotics and mechatronics, machine dynamics, crashworthiness, biomechanics, computer graphics, or system identification are also covered by the journal.

Please address contributions to

Prof. Guido Belforte
Prof. Andrea Manuello Bertetto
PhD Eng. Elvio Bonisoli

*Dept. of Mechanics
Technical University - Politecnico di Torino
C.so Duca degli Abruzzi, 24.
10129 - Torino - Italy - E.C.*

www.jomac.it
e_mail: jomac@polito.it

Subscription information

Subscription order must be sent to the publisher:

*Libreria Editrice Universitaria
Levrotto&Bella
C.so Luigi Einaudi 57/c – 10129 Torino – Italy*

www.levrotto-bella.net
e_mail: info@levrotto-bella.net
ph.: +39 011 4275423
mob.: +39 328 5369063
fax: +39 011 4275425

Preface for the special issue of the JoMaC dedicated to the 21st edition of the Workshop on Robotics in Alpe-Adria-Danube Region, RAAD 2013

The aim of the International Journal of Mechanisms and Control (JoMaC) is to present and promote the latest scientific achievements in the fields of mechanism, control systems and complex structures. Starting from 2003, nearly every year one issue of the International Journal of Mechanics and Control has been devoted to selected papers from the annual RAAD workshop.

The RAAD workshops started in 1992 when we realized that enhancing international cooperation, focusing on neighboring countries and unifying resources was important.

And it is also becoming more and more important today. Since then RAAD workshop has been each year organized in one of the countries in the Alpe-Adria-Danube region: 1992 Portorož (Slovenia), 1993 Krems (Austria), 1994 Bled (Slovenia), 1995 Pörschach (Austria), 1996 Budapest (Hungary), 1997 Cassino (Italy), 1998 Smolenice (Slovakia), 1998 Munich (Germany), 2000 Maribor (Slovenia), 2001 Vienna (Austria), 2002 Balatonfüred (Hungary), 2003 Cassino (Italy), 2004 Brno (Czech Republic), 2005 Bucharest (Romania), 2006 Balatonfüred (Hungary), 2007 Ljubljana (Slovenia), 2008 Ancona (Italy), 2009 Brasov (Romania), 2010 Budapest (Hungary), 2011 Brno (Czech Republic), 2012 Naples (Italy), and in 2013 in Portorož (Slovenija).

The thrust of research in the field of robotics has gradually moved from classical industrial applications to the challenging areas of service and humanoid robotics. Globalization forces us to intensify the cooperation between researchers and developers in all fields of their activities, something that is especially important for small neighborhood countries. Here, the RAAD conferences play an important role for the community in the Alpe-Adria-Danube Region.

The RAAD conference got its name from the geographical region Alpe-Adria-Danube, but was never meant to be limited to participants coming solely from this region. This year, we had 50 papers from researchers who come from 20 different countries, which were accepted for publication in the Proceedings of the RAAD 2013. The topics coincide with the contemporary trends in robotics as well as with the RAAD tradition.

According to the tradition, we have prepared a special issue of the 22nd International Workshop on Robotics in the Alpe-Adria-Danube Region, RAAD 2013, organized in Pororož, Slovenia. The selection of the contributed papers is a result of a careful reviewing process among the best presented papers, which fit the scope of the journal. This special issue brings the latest results from mechatronic design, advanced control, medical and rehabilitation robotics, and covers many challenges from contemporary service and humanoid robotics. We are convinced the readers will find this selection of excellent papers valuable and inspiring. We also believe that this special issue will promote future RAAD conferences and enhance international cooperation and scientific dissemination in Alpe-Adria-Danube region and abroad.

Finally, we would also like to express our gratitude to everybody who contributed to this event, especially the RAAD International Scientific Committee for the support and help both in the organization of the Workshop as well as in the paper selection for this journal special issue:

- Nikos A. Aspragathos, University of Patras, Greece
- Theodor Borangiu, Univ. Bucharest, Romania
- Marco Ceccarelli, University of Cassino, Italy
- Karol Dobrovodsky, Academy of Sciences, Slovakia
- Carlo Ferraresi, Polytechnic of Turin, Italy
- Stefan Havlik, Academy of Sciences, Slovakia
- Nick Andrei Ivanescu, Univ. Bucharest, Romania
- Roman Kamnik, University of Ljubljana, Slovenia
- Gernot Kronreif, Integrated Microsystems, Austria
- Ivan Petrović, University of Zagreb, Croatia
- Doina Pîslă, Univ. of Cluj-Napoca, Romania
- Aleksandar Rodić, Mihajlo Pupin Institute, Serbia
- Andreja Rojko, University of Maribor, Slovenia
- Cesare Rossi, University of Napoli, Italy
- Imre J. Rudas, Óbuda University, Budapest, Hungary
- Leon Žlajpah, Jožef Stefan Institute, Slovenia

RAAD 2013 conference chairs

Bojan Nemeč
Leon Žlajpah

WALKING WITH ADAPTIVE OSCILLATOR AND DYNAMIC MOVEMENT PRIMITIVES

Tadej Petrič* Leon Žajpah* Gianluca Garofalo** Christian Ott**

* Department for Automation, Biocybernetics and Robotics, Jožef Stefan Institute (JSI), Ljubljana, Slovenia

** Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Wessling, Germany

ABSTRACT

In this paper we address the problem of generating a stable periodic walking pattern for a biped robot. We propose a novel control framework based on adaptive oscillator combined with dynamic movement primitives. The proposed control system is a multi-step process, where in the first step the system learns a dynamic walking trajectory. The learning is based on a control actions derived from a spring loaded inverted pendulum dynamics, which is a biomechanically inspired template model for generating running and walking gaits. The learning process is done in real time and it is completely autonomous. After learning, the periodic walking pattern is executed with a low gain feedback controller combined with feed forward torque/force control signals. This generates a stable walking pattern with properties such as compliant behaviour and smooth interaction with unstructured environment. Compared with the SLIP walking control the proposed approach is more robust since it can produce stable walking patterns on a wider interval and it allows smooth modulation of forward walking velocity.

Keywords: adaptive oscillators, learning, dynamic walking, dynamic movement primitives

1 INTRODUCTION

The interest of building robots that can help humans in every day's life is becoming increasingly popular. There are already many commercially available robots in the market, specialised for housecleaning [1] or entertainment [2]. With a rapid development of such robotic technology, the interest of building versatile robots capable of autonomous learning is becoming more and more popular. The ability of learning new skills is one of the important issues that future robot companions will have to possess [3].

In the last decade many algorithms for machine learning were proposed and adopted in robotics. Their ability to learn complex models makes them a powerful tool, especially for learning the properties of the system. Their performance has been shown for learning kinematics [4], dynamics [5], locomotion [6] or even for playing the ball-in-a-cup game [7]. Although all these algorithms can learn different task properties their main shortcoming is that a large database or a long learning period is required so that they operate properly.

Special branch of learning algorithms are the biologically inspired methods. An extensive review for a computational mechanism of sensorimotor control, which covers methods from optimal feedback control [8] to the forward models and predictive control [9], was recently published by Franklin and Wolpert [10]. Another area of biologically inspired robotic control is the use of central pattern generators for control of periodic tasks [11]. Central pattern generators are neural circuits capable of producing coordinated patterns while receiving only simple input signals [12]. Their applicability was shown on a robotic salamander [12] and in locomotion [13].

The main contribution of this paper is a novel control framework, which combines biomechanically inspired central pattern generators with dynamic movement primitives for trajectory and corresponding control signal generation. The proposed framework is an extension of a two-layered system presented in [14], [15]. Here we augmented the two-layered system with an additional set of layers for learning the commanded signals for the periodic task.

To show the applicability of the proposed system we learn the dynamics of a bipedal spring loaded inverted pendulum (biSLIP) [16] applied to the multi-body system as shown in [17]. The biSLIP dynamic model is a biomechanical template for describing walking and running motions. The motivation for utilizing the biSLIP biomechanical model as

Contact author: Tadej Petrič

Email: tadej.petric@ijs.si
Jamova cesta 39, 1000 Ljubljana, Slovenia

a central pattern generator stems from the fact, that it can produce contact force profiles that closely match the experimental human data. The contact force generation in the biSLIP model is based on a point-foot contact and thus is not expected that large torques would be required at the ankle when implemented on a multi-body robot. However, the main drawback of the biSLIP model is that the stability of a limit cycle is highly dependent on the selected parameter set. Moreover it also has a limited range of stable performance in the sense of the forward walking velocity. By applying the proposed approach, which learns the biSLIP dynamic model behaviour in a first step, we can extend the stability region of the forward walking velocity. This is done by modulating the output of the multi-layered system. The proposed approach was applied on a five linked four degrees of freedom (DOF) under-actuated planar robot. The stability of the limit cycle was shown numerically with the Poincaré section as in [17].

The paper is organized as follows. In section 2, we give a short overview of the biSLIP model. In Section 3, we briefly explain how to impose biSLIP dynamics to the multi-body robot using operation space control. In Section 4 we give a detailed description of the proposed multi-layered control system. In Section 5 we evaluate the proposed approach in a simulated environment. Conclusions and summary are given in Section 6.

2 BIPEDAL SPRING LOADED INVERTED PENDULUM DYNAMICS

The biSLIP model consist of a total mass m concentrated at the hip and two massless linear spring legs as shown in Figure 1.

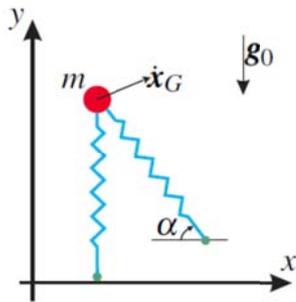


Figure 1. biSLIP model with vertically oriented support leg.

Because the legs are massless the reconfiguration to the desired angle of attack α is instantaneous. The dynamical system describing the behaviour of biSLIP model is given by

$$\ddot{\mathbf{x}}_G = \frac{1}{m}(\mathbf{f}_L(\mathbf{x}_G) + \mathbf{f}_R(\mathbf{x}_G)) + \mathbf{g}_0, \quad (1)$$

where \mathbf{g}_0 is the gravity acceleration, \mathbf{x}_G is the position for center of mass (CoM), and $\mathbf{f}_L(\mathbf{x}_G)$ and $\mathbf{f}_R(\mathbf{x}_G)$ are the left and right spring leg forces during contact. They are given by

$$\mathbf{f}_i(\mathbf{x}_G) = k(l_0 - \|\mathbf{x}_G - \mathbf{x}_{F,i}\|) \frac{\mathbf{x}_G - \mathbf{x}_{F,i}}{\|\mathbf{x}_G - \mathbf{x}_{F,i}\|}. \quad (2)$$

where i denotes the position of the left L and the right R foot contact points. l_0 is the rest length of the spring and k is the leg spring stiffness given by

$$k = \frac{k_t m g_0}{l_0}. \quad (3)$$

Here k_t is the stiffness constant. Note that, if leg is not in contact, the corresponding force is zero.

By considering the Poincaré section which corresponds to the vertical leg orientation (VLO) [18], we can numerically analyse the stability of the biSLIP model. Figure 2 shows the area with different initial velocities \dot{x}_G^0 , spring stiffness constants k_t and angle of attack α where stable forward walking can be achieved and confirmed with the Poincaré section.

In Figure 2 we can see that the forward walking velocity \dot{x}_G is bounded between ≈ 0.95 m/s and ≈ 1.3 m/s. We can also see that the relationship between different parameters, i.e. initial velocities \dot{x}_G^0 , spring stiffness constants k_t and angles of attack α , is nonlinear. This complicates the modulation of the forward walking velocity.

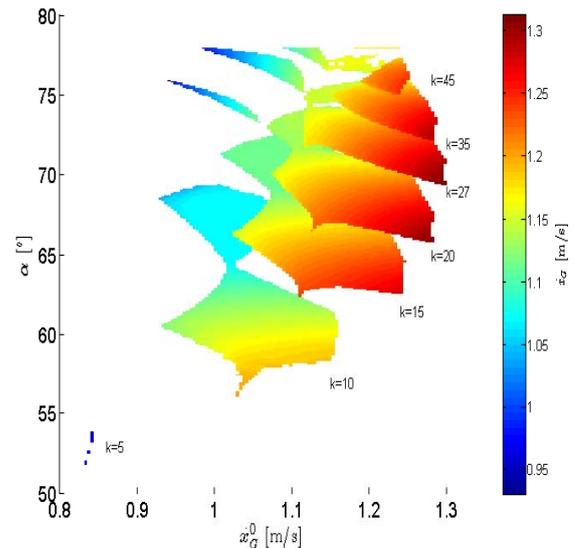


Figure 2 Stable walking regions confirmed with the Poincaré section for different initial velocities \dot{x}_G^0 , angle of attack α and spring stiffness constants k_t .

However, if the vertical movement of the biSLIP is mapped into the state space, then the state space trajectories remains similar even if the desired angle of attack α is changed. The results can see in Figure 3, where the angle of attack α was changed from 68 to 72 degrees with increments of 0.1 degree.

The similarity of responses with different angle of attack in phase plane indicates that it might be possible to simplify the process of setting the desired forward walking velocity while maintaining the properties of the biSLIP model. To achieve such a result we propose to use the adaptive oscillator combined with the dynamic movement primitives (DMPs) for learning the dynamic behaviour of the biSLIP

controller online. Once the biSLIP behaviour is learned, the DMPs modulation properties are used to change the desired walking speed.

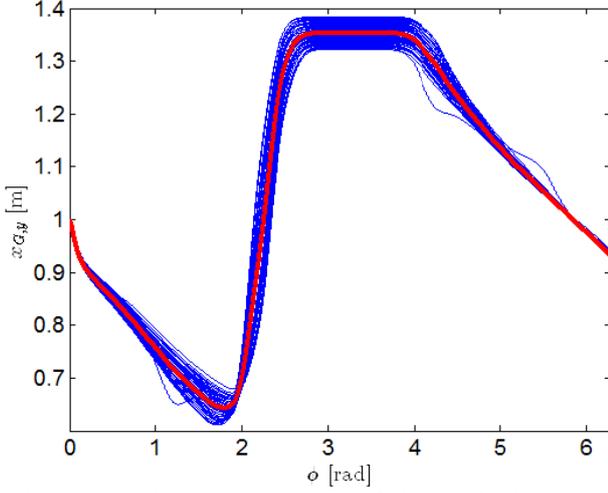


Figure 3 Motion $x_{G,y}$ (y motion of biSLIP) in the state space for $\alpha = [68, 68.1, \dots, 72]$ degrees, $\dot{x}_G^0 = 1.15$ m/s and $k_t = 20$ (blue lines). The red line is the average state space trajectory.

3 OPERATIONAL SPACE CONTROL

A five linked planar robot with the four rotational joints as shown in Figure 4 is considered. The dynamics can be described as

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{b} + \mathbf{g} = \mathbf{S}^T \boldsymbol{\tau} - \sum_{R,L} \mathbf{J}_{L,R} \mathbf{F}_{L,R}, \quad (4)$$

where \mathbf{M} is the mass matrix, \mathbf{b} is the Coriolis and centripetal force vector, \mathbf{g} is the gravitational force vector, $\mathbf{F}_{L,R}$ is the ground contact force for left and right legs, $\mathbf{J}_{L,R}$ are the corresponding Jacobian matrices, and \mathbf{S} is the selection matrix, which separates the actuated joints from generalized coordinates.

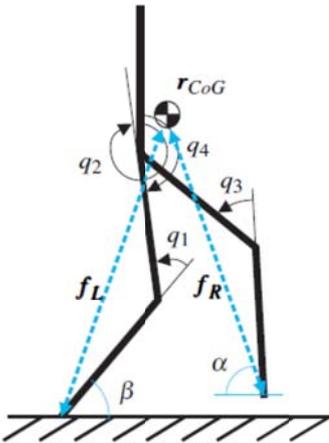


Figure 4 Multi-body robot with joints definitions. The angle β is used to parametrise the swing foot trajectory with respect to center of mass CoM as in [17].

Note that the contact force $\mathbf{F}_{L,R}$ is zero when the leg is not in contact. Otherwise, the following conditions hold

$$\dot{\mathbf{x}}_{L,R} = \mathbf{J}_{L,R} \dot{\mathbf{q}} = 0, \ddot{\mathbf{x}}_{L,R} = \mathbf{J}_{L,R} \ddot{\mathbf{q}} + \dot{\mathbf{J}}_{L,R} \dot{\mathbf{q}} = 0 \quad (5)$$

which leads to the solution from [19],

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{N}_{L,R} (\mathbf{b} + \mathbf{g}) + \mathbf{J}_{L,R}^T \boldsymbol{\Lambda}_{L,R} \dot{\mathbf{J}}_{L,R} \dot{\mathbf{q}} = (\mathbf{S} \mathbf{N}_{L,R})^T \boldsymbol{\tau}, \quad (6)$$

where $\boldsymbol{\Lambda}_{L,R}$ is the support space inertia matrix given by

$$\boldsymbol{\Lambda}_{L,R} = (\mathbf{J}_{L,R} \mathbf{M}^{-1} \mathbf{J}_{L,R}^T)^{-1}, \quad (7)$$

and $\mathbf{N}_{L,R}$ is the dynamically consistent support null-space.

To impose the dynamics of the biSLIP model given by (1)-(2) to the multi-body system, an operational space formulation [20] is used to project the stance dynamic with support constrain into the robot's center of mass \mathbf{r}_{CoM} . The control block diagram is shown on Figure 5. The operational space dynamics is given by

$$\boldsymbol{\Lambda}^* \ddot{\mathbf{r}}_{CoM} + \boldsymbol{\mu}^* + \mathbf{p}^* = \mathbf{F}. \quad (8)$$

This formulation can be used to compute the necessary operational space force \mathbf{F} , which is related to actuator torques $\boldsymbol{\tau} = \mathbf{J}^* \mathbf{F}$. For a detailed derivation of individual components of this projection, please refer to [16], [19], [21].



Figure 5 Block diagram of the operational space control that maps the biSLIP dynamic to the robot dynamics.

Inserting (1) into (8) leads to a biSLIP motion control

$$\boldsymbol{\tau} = \mathbf{J}^* \mathbf{T} \left(\boldsymbol{\Lambda}^* \left(\frac{1}{m} (\mathbf{f}_L(\mathbf{x}_G) + \mathbf{f}_R(\mathbf{x}_G)) + \mathbf{g}_0 \right) + \boldsymbol{\mu}^* + \mathbf{p}^* \right). \quad (9)$$

If the system would be perfectly modelled then the CoM motion will exactly follow the motion prescribed by the biSLIP dynamics. In addition to the CoM motion, which is our primary concern, we have to consider also the swing leg trajectory $\boldsymbol{\xi}$ and trunk orientation $\boldsymbol{\theta}$. For the exact derivation of those constrains see [17]. We stack both constrains into vector $\dot{\mathbf{x}}_t = [\dot{\boldsymbol{\xi}}^T \ \dot{\boldsymbol{\theta}}^T]^T$ and corresponding Jacobians into matrix $\mathbf{J}_t = [\mathbf{J}_\xi \ \mathbf{J}_\theta]$. To add these two additional constrains to lower priority tasks, a null-space projection \mathbf{N}^* is used. Therefore (9) yields

$$\boldsymbol{\tau} = \mathbf{J}^* \mathbf{T} \left(\boldsymbol{\Lambda}^* \left(\frac{1}{m} (\mathbf{f}_L(\mathbf{x}_G) + \mathbf{f}_R(\mathbf{x}_G)) + \mathbf{g}_0 \right) + \boldsymbol{\mu}^* + \mathbf{p}^* \right) + \mathbf{N}^* \mathbf{J}_t^* \dot{\mathbf{x}}_t. \quad (10)$$

4 MULTI-LAYERED CONTROL SYSTEM

The most important aspect of the proposed approach is the ability to learn the task trajectories and corresponding control signals. To achieve this, we use a machine learning

method, which incrementally updates the approximated models in real-time. The proposed control system framework is an extension of our previous work described in [14], [15]. The block scheme of the proposed control system is shown in Figure 6.

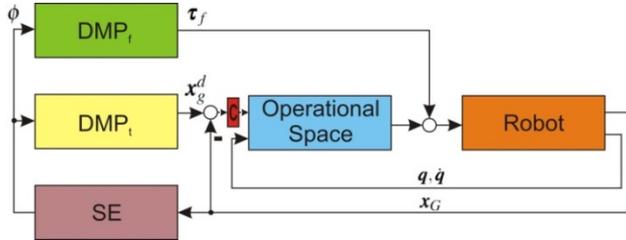


Figure 6 Block diagram of the proposed control system, where SE is the state estimator and C is the control feedback gain.

Here, instead of mapping complete dynamical models to the multi-body system, we use a set of layers based on dynamic movement primitives that learn the specific task trajectories and corresponding control signals. After learning, these trajectories with corresponding control signals are executed. This further allows natural compliant behaviour while maintaining sufficient tracking accuracy of the desired movement. With the *natural compliance* we denote the structural compliance of the mechanism, i.e. the compliance of the mechanism when the actuator torque is zero. The main advantage of the proposed control architecture is essentially the model free approach.

The stepping process of the algorithm is as follows. In the first step, the goal is to learn motion trajectories and corresponding control signals and in the second step these trajectories with corresponding control signals are executed. Different techniques exist for learning of the desired motion, e.g., the use of kinesthetic guidance [22], haptic interfaces [23], motion capture systems [24], or using models [25], etc. In the example presented in this paper, biSLIP model template was used in the learning phase.

To learn a periodic behaviour, a combination of adaptive oscillator and dynamic movement primitives was used. In our previous work [14], [15], [26], we explained that the adaptive oscillator can be used to extract the basic frequency from the demonstrated periodic movement and that DMPs can learn the waveform at the same time. It was also shown that imitating the desired waveform at an arbitrary frequency is possible.

Here the multi-layered movement imitation system consists of an adaptive oscillator (first layer) combined with output dynamical system (second and third layer) based on dynamic movement primitives denoted by $DMP_t(\Omega, \phi)$, where subscript t denotes the task trajectory for the second layer and by $DMP_f(\Omega, \phi)$ where subscript f denotes the control signals trajectories for the third layer. Note that the following equations are valid for one degree of freedom (DOF). For multiple DOFs these equations can be used in parallel.

The first layer, also called the state estimator, is based on an adaptive frequency oscillator combined with an adaptive

Fourier series [15]. It is a second order system consisting of differential equations given by

$$\dot{\phi} = \Omega - Ke \sin(\phi), \quad (11)$$

$$\dot{\Omega} = -Ke \sin(\phi), \quad (12)$$

where Ω is the extracted frequency, ϕ is the phase, K is the coupling constant and e is the difference between actual CoM position x_G and estimated \hat{x}_G CoM position, which is given by

$$\hat{x}_G = \sum_{c=0}^m (\alpha_c \cos(c\phi) + \beta_c \sin(c\phi)). \quad (13)$$

Here, m is the size of the Fourier series. The weights α_c and β_c are updated according to the following learning rule

$$\dot{\alpha}_c = \eta \cos(c\phi) e, \quad (14)$$

$$\dot{\beta}_c = \eta \sin(c\phi) e, \quad (15)$$

where η is the learning constant (see [15] for details). In this paper, the size of the Fourier series is $m=10$.

Augmenting the state estimator by anchoring the DMPs to the phase signal ϕ of the adaptive oscillator as in [14], [15] makes it possible to synchronize an arbitrary trajectory to an arbitrary periodic signal congruent with the desired task. Since the phase estimation and learning of the desired trajectory are done simultaneously, all system delays are automatically included [27]. The basic equations for dynamic movement primitives are summarized from [26], [28], [29]

$$\dot{b} = \Omega(\alpha_z(\beta_z - a - b) + f), \quad (16)$$

$$\dot{a} = \Omega b, \quad (17)$$

where α_z and β_z are the positive constants, which guarantee, that the system monotonically converges to the desired trajectory, and f is the nonlinear part that determines the shape of the trajectory given by

$$f = \frac{\sum_{i=1}^N \psi_i w_i r}{\sum_{i=1}^N \psi_i}, \quad (18)$$

where r is the parameter for amplitude modulation and ψ are Gaussian like kernel functions given by

$$\psi_i = e^{h(\cos(\phi - c_i) - 1)}, \quad (19)$$

Here, h is the width and c_i is the distribution on one period. If not stated otherwise, we use $c_i, i=1, \dots, 25$ and they are equally spread between 0 and 2π .

By applying locally weighted regression, the system learns the shape of the trajectory on-line. The equations are summarized from [26], where the target trajectory is governed by

$$f_t = \frac{\ddot{x}_G}{\Omega^2} - \alpha_z \left(\beta_z - x_G - \frac{\dot{x}_G}{\Omega} \right). \quad (20)$$

To update the amplitude of the kernel function ψ_i we use recursive least-squares method with forgetting factor λ . The

forgetting factor was set prior to $\lambda = 0.995$. With a given goal (20), the recursive algorithm is updating the weights w_i using the following rule

$$w_i(t+1) = w_i(t) + \psi_i P_i(t+1) r(t) e_r(t), \quad (21)$$

$$e_r(t) = f_t(t) - w_i(t) r(t), \quad (22)$$

$$P_i(t+1) = \frac{1}{\lambda} \left(P_i(t) - \frac{P_i(t)^2 r(t)^2}{\bar{\psi}_i + P_i(t) r(t)^2} \right), \quad (23)$$

If not stated otherwise, we use $w_i(0) = 0$ and $P_i(0) = 1$, where $i=1, \dots, 25$.

For the third layer $DMP_f(\Omega, \phi)$, equations for encoding and learning are similar as for the second layer $DMP_t(\Omega, \phi)$. The difference between $DMP_t(\Omega, \phi)$ and $DMP_f(\Omega, \phi)$ is that instead of learning the target function (20) for $DMP_t(\Omega, \phi)$, we learn the target function given as

$$f_t = \frac{\ddot{\mathbf{x}}_f}{\Omega^2} - \alpha_z \left(\beta_z - \boldsymbol{\tau}_f - \frac{\dot{\mathbf{x}}_f}{\Omega} \right), \quad (24)$$

where $\boldsymbol{\tau}_f$ is the commanded torque/force signal. By learning the control torque, which is generated as the combination of the operational space and biSLIP model, the system essentially learns the corresponding inverse dynamics $\boldsymbol{\tau}_f$ along the executed CoM trajectory \mathbf{x}_G^d .

After the learning phase, the movement trajectory \mathbf{x}_G^d and the corresponding torque/force control signal $\boldsymbol{\tau}_f$ are executed at the desired frequency Ω_d . The controller used for executing the motion is given by

$$\boldsymbol{\tau} = \mathbf{J}^{*T} (\mathbf{A}^* \mathbf{C} (DMP_t(\Omega, \phi) - \mathbf{x}_G) + \boldsymbol{\mu}^* + \mathbf{p}^*) + DMP_f(\Omega, \phi) + \mathbf{N}^* \mathbf{J}_t^* \dot{\mathbf{x}}_t, \quad (25)$$

where \mathbf{C} is the feedback gain and $DMP_f(\Omega, \phi)$ represents the feedforward torques that correspond to the biSLIP behaviour model.

5 SIMULATIONS

In this section we applied the proposed control approach to a five linked four DOF planar robot. The robot parameters are given in a Table 1. We used the same parameters as in [17].

Table 1 - Parameters for the multi-body planar robot.

	Shank	Thigh	Trunk
Mass – [kg]	5	5	60
Length – [m]	0.48	0.48	0.48
Inertia – [kg m ²]	0.01	0.01	1

To evaluate the performance of the proposed system we used the same stability analysis, i.e. the Poincaré map, as in [17]. Here we also assume that if the angle of attack and the walking velocities are constant then the energy is kept constant as well. To keep the walking velocity constant, while using the proposed approach, we modulated the amplitude of the $DMP_f(\Omega, \phi)$ by using the amplitude parameter r . The amplitude is updated based on the following control law

$$\dot{r} = C_v (\dot{x}_{G,y}^d - \dot{x}_{G,y}), \quad (26)$$

where C_v was experimentally set to 100 and the initial condition for r_0 was set to 1. The simulated results are shown in Figure 7. As we can see the control system can easily track the desired forward walking velocity $\dot{x}_{G,y}^d$. We can also see that it goes beyond the limits of the biSLIP model. The limits are indicated with the dotted red lines. For the exact limits see Figure 2.

In Figure 8 we show the behaviour of the proposed control system on the Poincaré map. The map corresponds to the same simulation as in Figure 7. Therefore, there is no unique fixed point because the desired forward velocity $\dot{x}_{G,y}^d$ was changing. However, we can see that there are some groups of fixed points to which the controller converges while it is in certain state, i.e. when the velocity was kept constant. For example, red dots indicate the last state from 50 s to 60 s. The simulation sequence of the robot model with the proposed control system is shown in Figure 9.

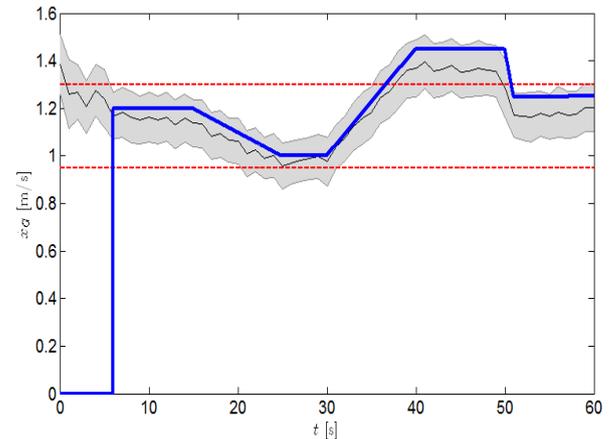


Figure 7 Walking with the desired forward velocity using the proposed approach. In the first 6 seconds a model based control as in [17] was used for learning the task trajectories $DMP_t(\Omega, \phi)$ and corresponding control signals $DMP_f(\Omega, \phi)$. After 6 s the model based control was abruptly switched to the DMP based controller. The blue line shows the desired forward walking velocity and the grey line shows the average forward walking velocity. The light grey area is the standard deviation. The dotted blue lines show the walking velocity limits of the biSLIP model.

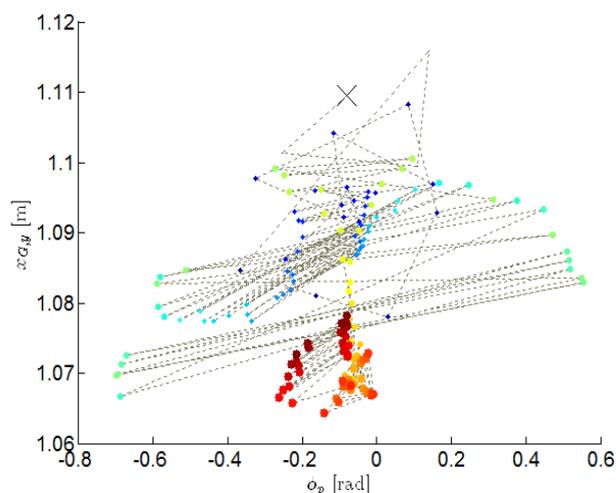


Figure 8 Poincaré section of the multi-body model controlled with the proposed approach.

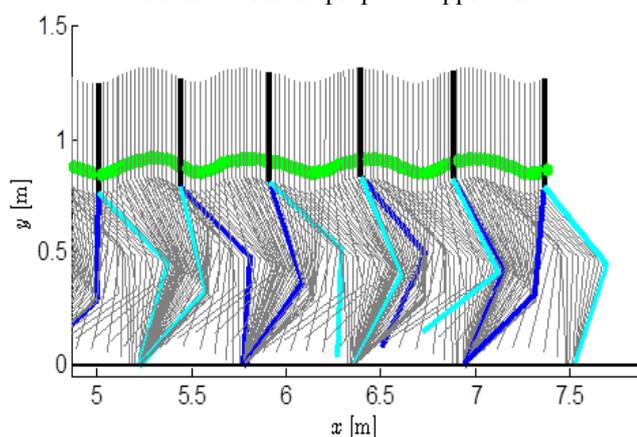


Figure 9 A sequence of photos shows a performance of the proposed approach in simulation. The green line shows the movement of the CoM.

6 CONCLUSION

We proposed a new control framework for learning the task trajectory and the corresponding control signal. The proposed control was applied to a five link planar walking robot, where we show that the system can learn the enforced dynamics based on the biSLIP model template. After a short learning phase, the system was also able to maintain the stable limit-cycle behaviour of the multi-body robot. Furthermore, with the proposed approach we could control the forward walking velocity in the area which cannot be achieved by using the biSLIP model. Note that the biSLIP model can produce a stable limit-cycle only in certain regions. With the Poincaré section map we also show that the proposed control converges to a fixed point, which proves that the proposed control system produces a stable limit cycle. In future, our plans are to investigate how to further exploit the walking region concerning the forward velocity, i.e. walking with slower forward velocity.

ACKNOWLEDGMENTS

This paper was partially funded by the Slovene Human Resources Development and Scholarship Found under grant number 11012-25/2012-7.

The third and fourth authors acknowledge support from the Initiative and Networking Fund of the Helmholtz Association (VH-NG-808).

REFERENCES

- [1] J. L. Jones, "Robots at the tipping point: the road to iRobot Roomba," *IEEE Robot. Autom. Mag.*, vol. 13, no. 1, pp. 76–78, Mar. 2006.
- [2] D. Gouaillier, V. Hugel, P. Blazevic, C. Kilner, J. Monceaux, P. Lafourcade, B. Marnier, J. Serre, and B. Maisonnier, "Mechatronic design of NAO humanoid," in *2009 IEEE International Conference on Robotics and Automation*, 2009, pp. 769–774.
- [3] S. Schaal and C. Atkeson, "Learning Control in Robotics," *IEEE Robot. Autom. Mag.*, vol. 17, no. 2, pp. 20–29, Jun. 2010.
- [4] a. D'Souza, S. Vijayakumar, and S. Schaal, "Learning inverse kinematics," in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*, 2001, vol. 1, no. Iros, pp. 298–303.
- [5] D. Nguyen-Tuong, M. Seeger, and J. Peters, "Model Learning with Local Gaussian Process Regression," *Adv. Robot.*, vol. 23, no. 15, pp. 2015–2034, Jan. 2009.
- [6] M. Kalakrishnan, J. Buchli, P. Pastor, and S. Schaal, "Learning locomotion over rough terrain using terrain templates," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 167–172.
- [7] B. Nemeč, R. Vuga, and A. Ude, "Exploiting previous experience to constrain robot sensorimotor learning," in *2011 11th IEEE-RAS International Conference on Humanoid Robots*, 2011, pp. 727–732.
- [8] C. G. Atkeson and J. M. Hollerbach, "Kinematic features of unrestrained vertical arm movements.," *J. Neurosci.*, vol. 5, no. 9, pp. 2318–30, Sep. 1985.
- [9] D. M. Wolpert and M. Kawato, "Multiple paired forward and inverse models for motor control.," *Neural Netw.*, vol. 11, no. 7–8, pp. 1317–29, Oct. 1998.
- [10] D. W. Franklin and D. M. Wolpert, "Computational mechanisms of sensorimotor control.," *Neuron*, vol. 72, no. 3, pp. 425–42, Nov. 2011.
- [11] S. Schaal, A. Ijspeert, and A. Billard, "Computational approaches to motor learning by imitation.," *Philos. Trans. R. Soc. Lond. B. Biol. Sci.*, vol. 358, no. 1431, pp. 537–47, Mar. 2003.
- [12] A. J. Ijspeert, "Central pattern generators for locomotion control in animals and robots: a review.," *Neural Netw.*, vol. 21, no. 4, pp. 642–53, May 2008.

- [13] L. Righetti, "Programmable central pattern generators: an application to biped locomotion control," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, 2006, no. May, pp. 1585–1590.
- [14] A. Gams, T. Petric, A. Ude, and L. Žlajpah, "Performing Periodic Tasks: On-Line Learning, Adaptation and Synchronization with External Signals," in *The Future of Humanoid Robots - Research and Applications*, R. Zaier, Ed. Intech, 2012, pp. 1–28.
- [15] T. Petric, A. Gams, A. J. Ijspeert, and L. Zlajpah, "On-line frequency adaptation and movement imitation for rhythmic robotic tasks," *Int. J. Rob. Res.*, vol. 30, no. 14, pp. 1775–1788, Sep. 2011.
- [16] M. Hutter, C. D. Remy, M. A. Hopflinger, and R. Siegwart, "SLIP running with an articulated robotic leg," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 4934–4939.
- [17] G. Garofalo, C. Ott, and A. Albu-Schaeffer, "Walking control of fully actuated robots based on the Bipedal SLIP model," in *2012 IEEE International Conference on Robotics and Automation*, 2012, pp. 1456–1463.
- [18] A. Seyfarth, H. Geyer, M. Günther, and R. Blickhan, "A movement criterion for running," *J. Biomech.*, vol. 35, no. 5, pp. 649–55, May 2002.
- [19] L. Sentis, "Compliant Control of Whole-body Multi-contact Behaviors in Humanoid Robots," in *Motion Planning for Humanoid Robots*, no. 1, H. Kensuke, Y. Eiichi, and Y. Kazuhito, Eds. Springer London, 2008, pp. 29–67.
- [20] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE J. Robot. Autom.*, vol. 3, no. 1, pp. 43–53, Feb. 1987.
- [21] M. Hutter, M. Hoepflinger, and C. Gehring, "Hybrid Operational Space Control for Compliant Legged Systems," in *Robotics Science and Systems VIII*, 2012.
- [22] D. Kushida, M. Nakamura, S. Goto, and N. Kyura, "Human direct teaching of industrial articulated robot arms based on force-free control," *Artif. Life Robot.*, vol. 5, no. 1, pp. 26–32, Mar. 2001.
- [23] P. Evrard, E. Gribovskaya, S. Calinon, A. Billard, and A. Kheddar, "Teaching physical collaborative tasks: object-lifting case study with a humanoid," in *2009 9th IEEE-RAS International Conference on Humanoid Robots*, 2009, pp. 399–404.
- [24] J. Babic, J. G. Hale, and E. Oztop, "Human sensorimotor learning for humanoid robot skill synthesis," *Adapt. Behav.*, vol. 19, no. 4, pp. 250–263, Jul. 2011.
- [25] T. Petrič, B. Curk, P. Cafuta, and L. Žlajpah, "Modelling of the robotic Powerball®: a nonholonomic, underactuated and variable structure-type system," *Math. Comput. Model. Dyn. Syst.*, vol. 16, no. 4, pp. 327–346, Nov. 2010.
- [26] A. Gams, A. J. Ijspeert, S. Schaal, and J. Lenarčič, "On-line learning and modulation of periodic movements with nonlinear dynamical systems," *Auton. Robots*, vol. 27, no. 1, pp. 3–23, May 2009.
- [27] T. Petric, A. Gams, M. Tomsic, and L. Zlajpah, "Control of rhythmic robotic movements through synchronization with human muscle activity," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 2172–2177.
- [28] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, 2002, vol. 2, no. May, pp. 1398–1403.
- [29] S. Schaal, P. Mohajerian, and A. Ijspeert, "Dynamics systems vs. optimal control--a unifying view.," *Prog. Brain Res.*, vol. 165, no. 1, pp. 425–45, Jan. 2007.

DESIGN AND SIMULATION FOR KINEMATIC CHARACTERISTICS OF A TRIPOD MECHANISM FOR BIPED ROBOTS

Mingfeng Wang Marco Ceccarelli

LARM: Laboratory of Robotics and Mechatronics
DiCEM-University of Cassino and South Latium, Cassino (Fr), Italy

ABSTRACT

In this paper, design and kinematic analysis of a novel leg with a tripod mechanism is presented for biped robots. The configuration of the proposed tripod mechanism has been determined to make the moving platform possess pure translational motion. Kinematics analysis has been formulated for a characterization and evaluation of performances. A 3-D model has been elaborated in SolidWorks® both for design and simulation purposes. Simulation results show that the proposed mechanism is able to perform movements of a human-like gait trajectory with suitable motion capability, flexibility, and operation performances

Keywords: Biped Robots, Leg Mechanisms, Tripod Mechanisms, Human-like Walking Foot Step.

1 INTRODUCTION

The development of biped robots is a hot topic that has attracted interests of many research communities in the past decade [1]-[2]. Biped robots have higher mobility and better obstacle crossing ability, with respect to conventional wheeled and crawler-type robots, especially when moving in rough or unconstructed environment [3].

Compared with serial architecture, parallel architecture is well known for having better performances in terms of dynamic performance, accuracy, ratio of payload to own weight and has been widely studied both in industry and academia [4][5]. Furthermore, in the field of leg designs for biped robots, Ota [6] and Sugahara [7] have proposed to use a Gough-Stewart parallel mechanism for leg modules. In particular, in November 2003 WL-16 (Waseda Leg-No.16) had achieved world first dynamic biped walking by carrying an adult human and the latest version of this biped robot is currently WL-16RV [8]. However, the typical Stewart-based hexapods suffer from some disadvantages, e.g., reduced workspace, difficult mechanical design, complex direct kinematics and control algorithms. Ceccarelli and Carbone

[9] have investigated the possibility of using parallel manipulator architectures with less than six degrees of freedom for leg design in legged walking robots as inspired from the leg human muscular system. In addition, parallel manipulators with 3-DOF, have been widely investigated for relevant applications [10], and they have simpler structure and kinematics, larger workspace, and more convenient control with respect to hexapods. Nevertheless, the potentiality of parallel architectures for leg designs has not been fully investigated, since no other designs have been proposed with efficient prototypes in the literature or in lab environments. Most of the existing biped robots are based on leg designs with human-like architectures by using serial chain solutions, such as ASIMO, NAO, HUBO, HRP-4 [13].

In this paper, a novel leg design has been proposed by using a tripod mechanism for biped robots. Configuration and kinematic analysis of the proposed architecture have been determined and formulated for characterization and evaluation of performances, respectively. A three-dimensional (3-D) model has been elaborated in SolidWorks® environment with low-cost easy-operation high-payload features. Based on a human-like gait trajectory, kinematic simulation of the designed model has been carried out in SolidWorks® environment to characterize and to evaluate the operation performances of the proposed tripod mechanism for biped robots.

Contact authors: Mingfeng Wang, Marco Ceccarelli.

Email: {wang, ceccarelli}@unicas.it

Address: Via Di Biasio 43, 03043, Cassino (Fr), Italy.

2 CONFIGURATION AND KINEMATICS

For further characterization and evaluation of the proposed tripod mechanism for biped robots, configuration and kinematics are determined and formulated in this section.

The proposed architecture of a tripod mechanism is a 3-UPU parallel manipulator, as shown in Figure 1. This manipulator consists of a fixed base (waist), a moving platform (foot), and three extensible limbs of identical kinematic structure. Each limb connects the fixed base to the moving platform by a prismatic joint and two universal joints (U-joint) at each end. A linear actuator has been adopted to drive each prismatic joint. Hence, there are eight links connected by six U-joints and three prismatic joints, and the degrees of freedom of the mechanism can be calculated as

$$F = 6(n - j - 1) + \sum_{i=1}^j f_i = 6(8 - 9 - 1) + (6 \times 2 + 3 \times 1) = 3 \quad (1)$$

Since the joint degrees of freedom of each limb are equal to five, each limb provides one constraint to the moving platform. As shown in Figure 1, the U-joints in each limb are arranged with two outer revolute joint axes parallel to each other and the two inner revolute joint axes also parallel to one another, so that each limb provides one rotational constraint to the moving platform. A combination of three limbs completely constrains the moving platform from any instantaneous rotation. Hence, if the constraints are independent of one another, the moving platform possesses pure translational motion, as discussed in [15].

For the purpose of kinematic analysis, static coordinate frame A: $O\text{-}xyz$ and moving coordinate frame B: $P\text{-}uvw$ are fixed on the platforms as shown in Figure 2, and a position

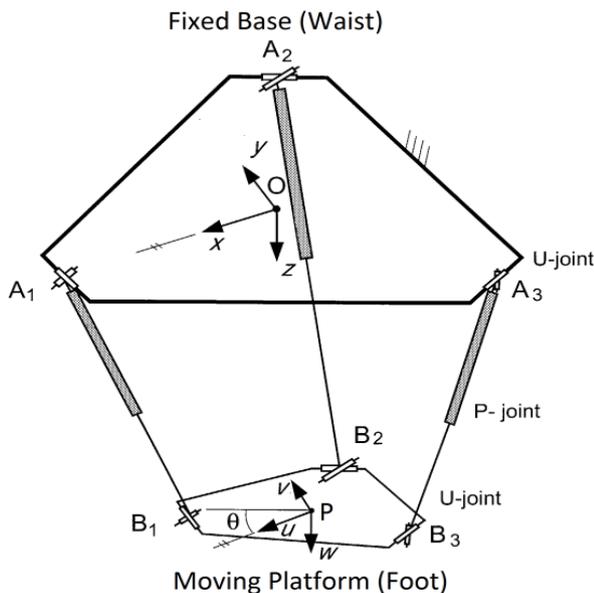


Figure 1 A kinematic scheme of a 3-UPU parallel manipulator as tripod mechanism

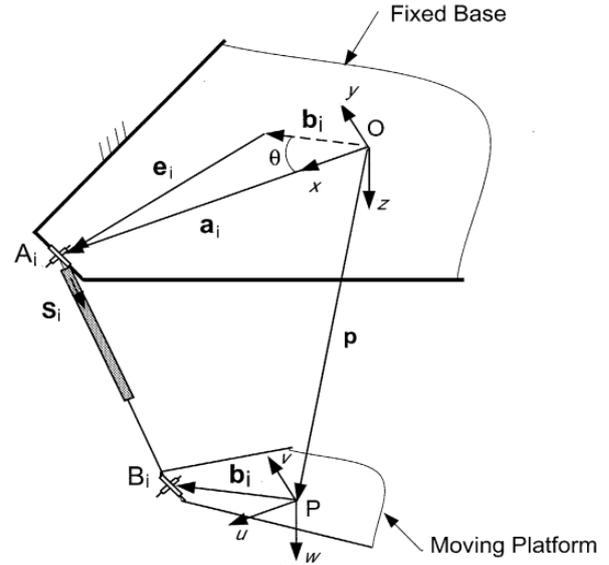


Figure 2 A kinematic model for typical limb of the 3-UPU parallel manipulator

vector $p = [p_x, p_y, p_z]^T$ of a reference point P in the moving platform is given for indexing a walking performance. The i^{th} ($i=1, \dots, 3$) actuated limb is connected to the moving platform at point B_i and to the fixed base at point A_i . Furthermore, we assume that points A_i ($i=1, \dots, 3$) lie into equilateral triangle on the $O\text{-}xy$ plane at a radial distance of r_a from point O , and B_i ($i=1, \dots, 3$) lie into equilateral triangle on the $P\text{-}uv$ plane at a radial distance of r_b from point P , respectively. Hence, the position vectors of A_i and B_i can be obtained as

$$\begin{aligned} \mathbf{a}_1 &= [r_a, 0, 0]^T & \mathbf{b}_1 &= [r_b, 0, 0]^T \\ \mathbf{a}_2 &= \left[-\frac{r_a}{2}, \frac{\sqrt{3}r_a}{2}, 0 \right]^T & \mathbf{b}_2 &= \left[-\frac{r_b}{2}, \frac{\sqrt{3}r_b}{2}, 0 \right]^T \\ \mathbf{a}_3 &= \left[-\frac{r_a}{2}, -\frac{\sqrt{3}r_a}{2}, 0 \right]^T & \mathbf{b}_3 &= \left[-\frac{r_b}{2}, -\frac{\sqrt{3}r_b}{2}, 0 \right]^T \end{aligned} \quad (2)$$

Referring to Figure 2, a vector-loop equation can be written for each limb as

$$\mathbf{l}_i = q_i \cdot \mathbf{s}_i = \mathbf{p} + \mathbf{b}_i - \mathbf{a}_i \quad (3)$$

where l_i is the length of the i^{th} limb, namely the magnitude of vector \mathbf{l}_i , \mathbf{s}_i is a unit vector pointing along the direction of the i^{th} limb, and $\mathbf{a}_i, \mathbf{b}_i$ can be obtained when r_a, r_b are given. For inverse kinematic analysis, it is to calculate the limb lengths l_i ($i=1, \dots, 3$) according to the given position vector $\mathbf{p} = [p_x, p_y, p_z]^T$ of the moving platform. The solution can be expressed by inverting (3) in the form as

$$\begin{aligned}
 l_1 &= \sqrt{(p_x - w)^2 + p_y^2 + p_z^2} \\
 l_2 &= \sqrt{(p_x + w/2)^2 + (p_y - \sqrt{3}w/2)^2 + p_z^2} \\
 l_3 &= \sqrt{(p_x + w/2)^2 + (p_y + \sqrt{3}w/2)^2 + p_z^2}
 \end{aligned} \quad (4)$$

where $w = r_a - r_b$, and the negative limb length cannot be obtained in this mechanism.

For direct kinematic analysis, it is to calculate the position vector $\mathbf{p} = [p_x, p_y, p_z]^T$ of the moving platform according to the limb lengths l_i ($i=1, \dots, 3$). The solution can be expressed as

$$\begin{aligned}
 p_x &= (-2l_1^2 + l_2^2 + l_3^2) / 6w \\
 p_y &= (l_3^2 - l_2^2) / 2\sqrt{3}w \\
 p_z &= \sqrt{l_1^2 - (p_x - w)^2 - p_y^2}
 \end{aligned} \quad (5)$$

where negative root of p_z cannot be obtained in this mechanism.

It is obvious that the kinematic equations (4) and (5) are explicit functions and easy to solve, hence, it is fairly easy to obtain inverse and direct kinematic solutions. Furthermore, the equations also show that there is no singularity and kinematic coupling in the whole workspace, which are distinct advantages in practical applications.

3 A 3-D MODEL OF THE TRIPOD MECHANISM

A 3-D model of the proposed tripod mechanism has been designed in SolidWorks® environment, as shown in Figure 3. The tripod mechanism is a parallel manipulator consisting of a relatively fixed waist plate and moving foot plate, between them are three identical linear actuators with U-joints at each end.

Furthermore, in order to improve the compactness of the proposed tripod mechanism, the male part of the lower U-

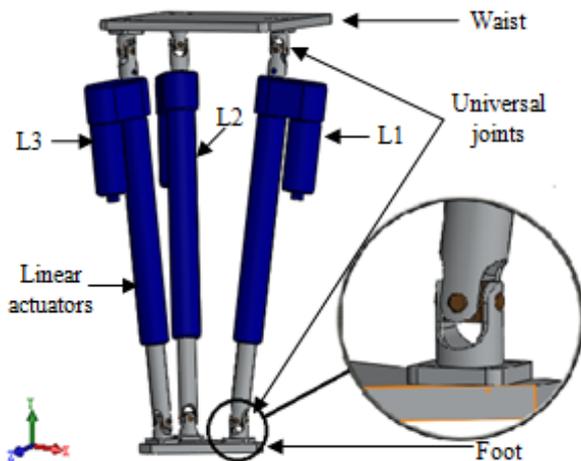
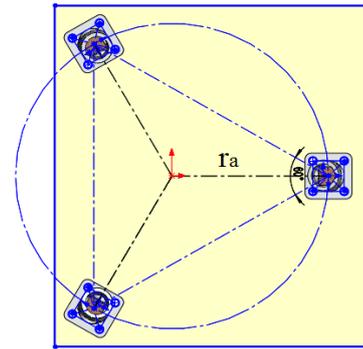
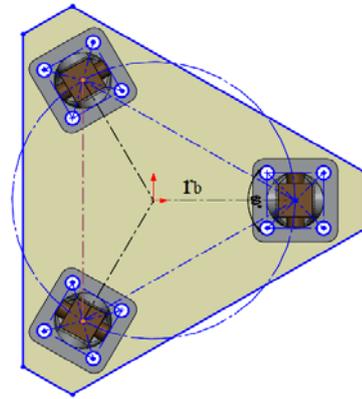


Figure 3 A 3-D model of the proposed tripod mechanism in SolidWorks® environment



a)



b)

Figure 4 A scheme of arrangement of the U-joints: a) in the upper plate (waist); b) in the lower plate (foot)

joints is designed as the bottom part of the rod of each linear actuator, and the assembly of lower three U-joints are shown in Figure 3 in the zoomed view. Additionally, the upper and lower three U-joints are installed in equilateral triangle arrangement with one ahead and the other two rear, as shown in Figure 4, where each three inner revolute axes are installed pointing to the circumcenter of the triangle and the circumradiuses are r_a and r_b , respectively. To guarantee the moving platform (foot) possessing pure translational motion, the two U-joints in each limb are arranged with the two outer revolute joint axes parallel to each other and the two inner revolute joint axes also parallel to one another, as indicated in [15].

The main specifications of the designed model are listed in Table I, and the details of dimension parameters are respectively indicated and listed in Figure 5 and Table II, where the distance between the upper U-joint and waist is equal to that between the lower U-joint and foot, noted as $d_{UW} = d_{UF}$; the distance between the two U-joints in each linear actuator, namely the length of each linear actuator, is noted as l_i , and the initial value of l_i is set as l_{i0} , which determines the initial height of the designed model; the stroke of three linear actuators is noted as l_s , hence, the displacement of each linear actuator, noted as L_i , can be expressed as

Table I - Main specifications of the 3-D model of the proposed tripod mechanism for Figure 3

Degrees of freedom	Weight (kg)	Dimension (mm)	Step size (mm)	Leg motion cycle (s/step)
3	4.5 kg	223×200×470	245×45	4

Table II – Designed dimension parameters for the model in Figure 5 (in mm)

l_W	l_F	w_W	w_F	h_W	h_F	d_{UW}	d_{UF}	H	l_{i0}
223	140	200	121	10	10	35	35	470	403

Table III – Mass values of components for the 3-D model in Figure 3 (in kg)

m_W (Waist)	m_F (Foot)	m_L (Linear actuator)	m_U (U-joints)	Total mass
1.20	0.28	0.90	0.04	4.50

$$L_i = l_i - l_{i0} \quad (6)$$

Since the proposed tripod mechanism is developed for biped robots, which will be capable of moving with flexibility and versatility in practical applications, during the mechanical design, particular attentions have been paid to the characteristics of low cost, load capacity, easy operation, lightweight and compact. The solution for that is to choose proper commercial products which have been extensively used in model design and select aluminium as the material for the plates of waist and foot for its proper stiffness, mass density, and cheap price.

Furthermore, in this model, the mass center is set in the geometry center of each element and the mass values of each component are listed in Table 3, as m_W , m_F , m_L , and m_U for the tripod mechanism. The torques of three linear actuators, which are set up at the top of the rods, are noted as T_1 , T_2 , and T_3 . All of the above parameters are designed to evaluate the proposed tripod mechanism design for a practical feasible operation and optimal design purposes.

The mobile platform is used as foot with pure translational motion that is ensured by the kinematic structure of the parallel leg mechanism. The feasibility of the mechanical

design is obtained by using light materials (like aluminium alloy) and commercial components that have been modelled properly in the proposed CAD design for simulation.

4 SIMULATION MODEL AND RESULTS

The purpose of this paper is to verify the possible application of a tripod mechanism for biped robots, whose 3-D model has been elaborated in SolidWorks® environment, as shown in Figure 3. Consequently, the kinematic analysis of human-like gait trajectory is necessary to be prescribed before the simulation. For the purpose of kinematic analysis, the reference coordinate system for the measured characterization values has been fixed on the ground in SolidWorks® environment, also as shown in Figure 3.

For human normal walking, the motion of a leg can be divided into two phases, namely a propelling phase and non-propelling phase [13]. In the prescribed human-like foot step trajectory which is an ovoid curve, as shown in Figure 6a), the straight line segment represents the supporting phase and the curve segment represents the swinging phase.

During the leg motion cycle, point C will be assumed as the starting point and the foot step trajectory will be followed the sequences of C-B-D-A-C for kinematic simulation, although a practical foot step trajectory will be performed from point A to point B passing through point C, namely the curve segment in Figure 6a). Furthermore, the walking gait can be characterized by a step length s and a step height h , which are assumed as $s=235\text{mm}$ and $h=45\text{mm}$ and the kinematic analysis of the prescribed foot step has been carried out to obtain the data of the position as function of axes in MATLAB® environment, which has been plotted in Figure 6 b).

Then, the input motions of three linear actuators for the tripod mechanism, namely the displacements of each rod, noted as L_i , can be calculated via inverse kinematics (4) and (6) by using both the position data of prescribed human-like foot step trajectory and the dimension parameters of the proposed model in the MATLAB® environment.

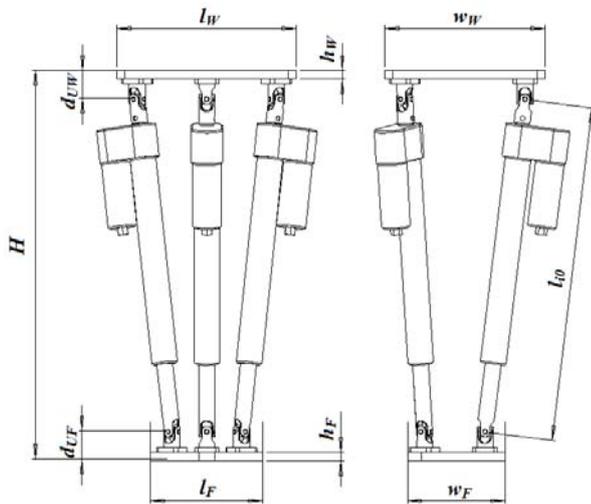


Figure 5 Dimension parameters of the 3-D model in Fig 3

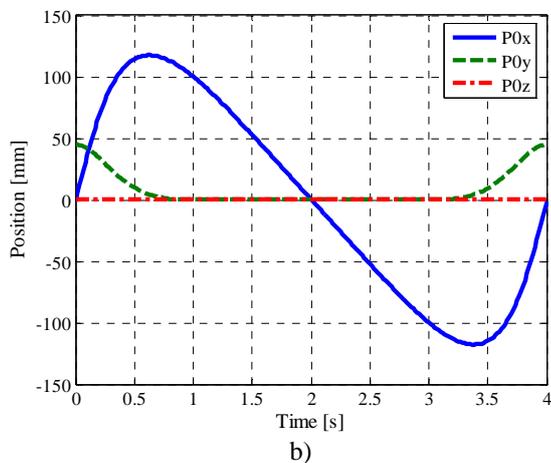
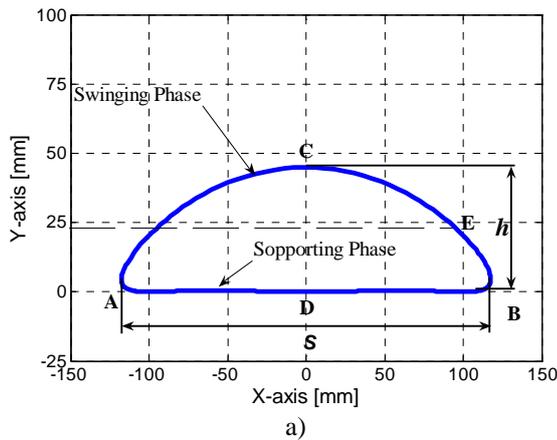


Figure 6 Human-like foot step trajectory:

- a) a prescribed foot step, where $s=235\text{mm}$ and $h=45\text{mm}$;
 b) a kinematic analysis

In the Figure 7, the input displacements of three linear actuators as function of time has been shown as synchronized with the motion of the prescribed step trajectory. Particularly, since the prescribed step trajectory moves in the $O-xy$ plane, and the rear two linear actuators are assembled symmetrically about the plane, it is obviously to obtain that the rear two linear actuators gain coincident displacements in the simulation. In addition, an initial displacement of three linear actuators has been set up as $L_0=20\text{mm}$, and the maximum of length of linear actuator is less than 100 mm, hence, the stroke of the linear actuator is determined as 100 mm.

The computed input actuation displacements of three linear actuators, as shown in Figure 7, are useful for selecting actuators with proper stroke. By using the corresponding functions that are available in SolidWorks® environment, the input motions of three linear actuators are obtained. Simulation time has been prescribed in 12 s (leg motion cycle is 4 s) to simulate the function of the tripod mechanism in three full cycles of human-like foot step, during which $100 \times 12 = 1200$ simulation steps are computed.

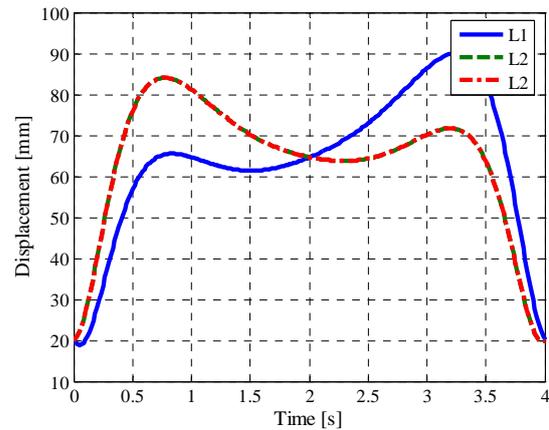


Figure 7 Input displacements of three linear actuators

The input data for numerical simulations have been assumed by referring to a human-like foot trajectory with sizes according to an average human walking. Those values have determined proper motion for the leg tripod mechanism with numerical results that are reported in Figures 8 to 11.

Figures 8 to 10 show the kinematic simulation results of the mass center of the foot (MC), which moves along X -axis and Y -axis but not along Z -axis, and that is coincident with the prescribed foot step trajectory moving in the $O-xy$ plane. In addition, Figure 11 shows the dynamic simulation results of three linear actuators, whose torques are indicated along X -axis, Y -axis and Z -axis, respectively.

In the Figure 8, the maximum position differences along X -axis and Y -axis, noted as ΔP_x and ΔP_y , have been respectively computed as 235.05 mm and 45.05 mm, which are approximately equal to the prescribed step length s and height h . The minimum value of P_y , computed as 5mm shows the height of MC with respect to the ground, which is $1/2h_f$.

In the Figure 9, the maximum velocities have been computed as 384.69 mm/s along X -axis and -94.80 mm/s along Y -axis, which occur at the point C (P_y is maximal) and the point E (P_y is medial) of the trajectory, respectively.

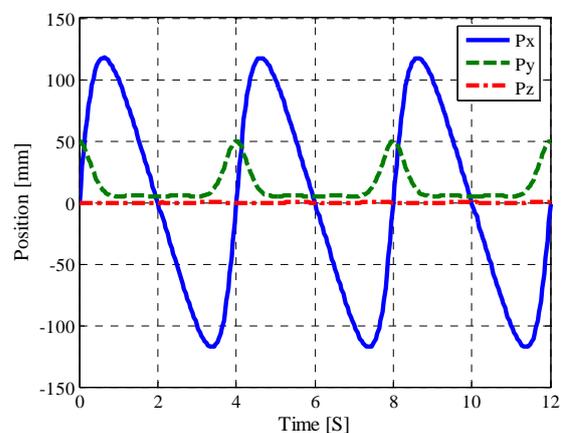


Figure 8 Computed position coordinates of MC

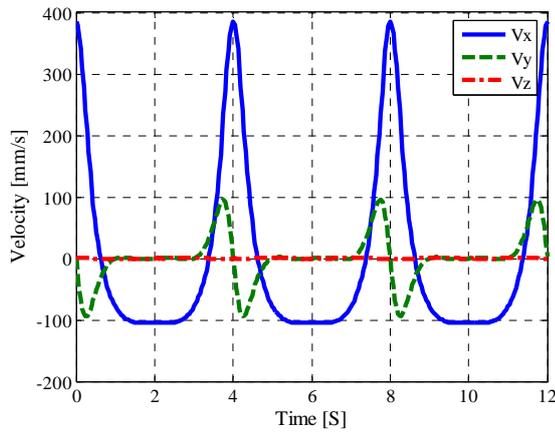


Figure 9 Velocity components of MC

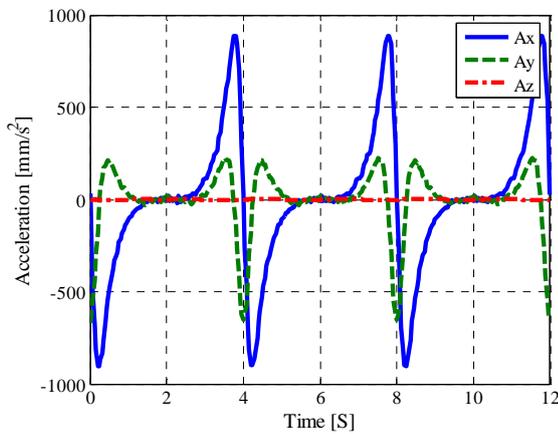


Figure 10 Acceleration components of MC

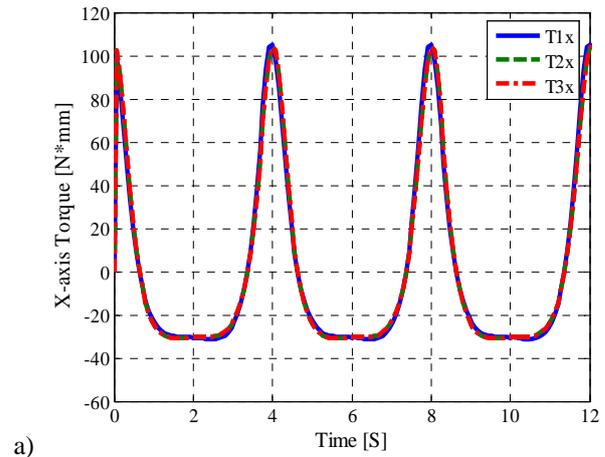
Particularly, a tiny error between maximum forward and reverse velocity along Y-axis is caused by the force of gravity.

In the Figure 10, the maximum accelerations have been computed as -902.36 mm/s^2 along X-axis, which occurs 0.24s after the beginning of each step; and 669.87 mm/s^2 along Y-axis, which occur exactly at the start of each step. Particularly, the tiny error between maximum forward and reverse acceleration along X-axis is caused by the force of inertia.

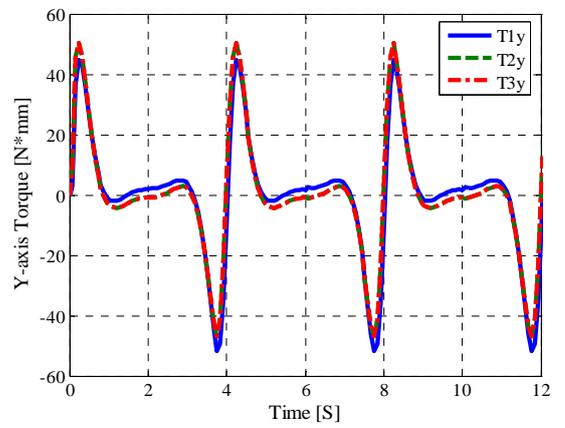
In the Figure 11, both the torque components along X-axis and Y-axis of three linear actuators are approximately equal with each other. Nevertheless, along Z-axis, T_{1z} is always equal with zero, and T_{2z} , T_{3z} are symmetrical about axis $T_z=0$, which is determined by the prescribed human-like foot step trajectory. In addition, by using (7), which is shown as

$$T_i = \sqrt{T_{ix}^2 + T_{iy}^2 + T_{iz}^2} \quad (7)$$

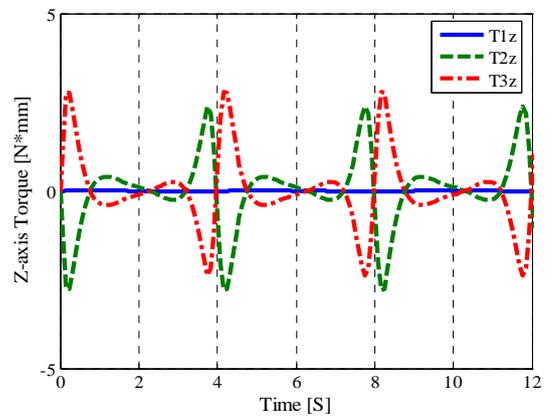
the magnitudes of T_1 , T_2 , and T_3 can be calculated, and the maximum values can be obtained as $109.76 \text{ N}\cdot\text{mm}$, 106.37



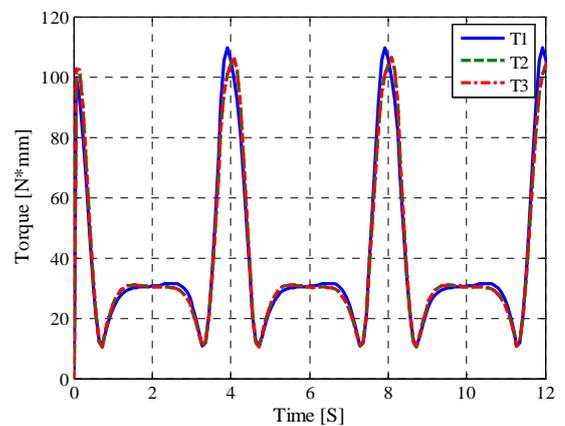
a)



b)



c)



d)

Figure 11 Torque components of three linear actuators: a) in X-axis; b) in Y-axis; c) in Z-axis; d) Magnitude

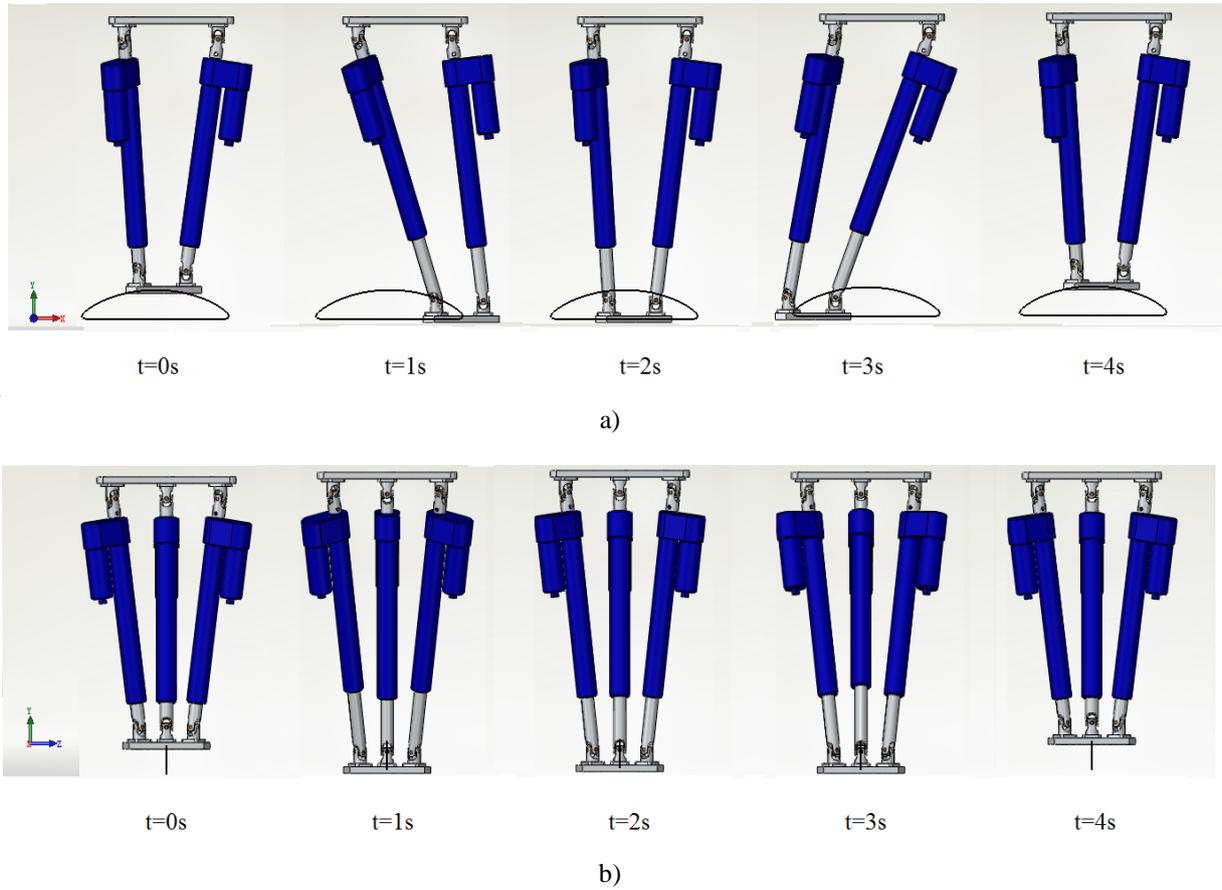


Figure 12 Snapshots of the simulated motion sequences of the tripod mechanism: a) in O - xy plane b) in O - yz plane

N·mm, and 106.37 N·mm, respectively, which are useful for choosing proper commercial products.

Figure 12 shows the snapshots of the motion sequences during the simulation of one leg motion cycle in front plane and right plane, respectively. Corresponding simulation time is indicated for both of snapshots in O - xy plane (front view) and in O - yz plane (side view). We can see that the tracked trajectory of MC in the designed model of the tripod mechanism can well imitate movements of human-like foot step trajectory in O - xy plane and no offset in Z -axis.

Each step action refers to the linear segment as for supporting phase and the curvilinear segment is used in the swinging phase, and the prescribed step trajectory includes the linear segment while a practical walking step not, therefore, simulated leg motion cycle is double of practical step cycle, namely, one step will have a practical duration of 2s in the simulated trajectory for 4s, noted as $\Delta t_{\text{step}}=4s$,

and a practical duration of 1s in the simulated trajectory for 2s, noted as $\Delta t_{\text{step}}=2s$. The characteristic parameters, which are the maximum values of displacement, velocity, acceleration of MC and torques of three linear actuators, are listed with $\Delta t_{\text{step}}=2s$ and $\Delta t_{\text{step}}=4s$ in Table IV.

The values of displacement, velocity, acceleration indicate that ΔP_x and ΔP_y are almost the same, V_x exhibits linear variation with leg motion cycle while A_x not. The ratios of T_i at $\Delta t_{\text{step}}=2s$ to T_i at $\Delta t_{\text{step}}=4s$ are respectively 3.65, 3.37 and 3.37, which are non-linear variation with leg motion cycle. The computed performance in terms of kinematic characteristics of MC, and dynamic characteristics of linear actuators in two walking conditions can be used to evaluate the proposed tripod mechanism for a practical feasible operation and to work out optimal design from the procedure.

Table IV – Characteristic parameters of the kinematic simulation

	ΔP_x (mm)	ΔP_y (mm)	V_x (mm/s)	A_x (mm/s ²)	T_1 (N·mm)	T_2 (N·mm)	T_3 (N·mm)
$\Delta t_{\text{step}}=2s$	235.22	45.06	770.03	-3.72×10^3	400.26	357.94	357.94
$\Delta t_{\text{step}}=4s$	235.05	45.05	384.69	-902.36	109.76	106.37	106.37

5 CONCLUSIONS

A mechanical design and walking simulation have been presented for a new tripod mechanism for biped robots in this paper. The configuration and kinematics analysis of the proposed 3-UPU parallel manipulator have been determined and formulated for characterization and evaluation of performances, respectively. A 3D model has been elaborated in SolidWorks® environment for the evaluation design performances and checking the operation feasibility. Simulation results show that the proposed mechanical designed model with proper input motions for the three linear actuators can well imitate movements of human-like foot step trajectory, and has practical and feasible operation performances of the tripod mechanism for biped robots.

ACKNOWLEDGMENTS

The first author would like to acknowledge Chinese Scholarship Council (CSC) for supporting his PhD study and research at the Laboratory of Robotics and Mechatronics (LARM) in the University of Cassino and South Latium, Italy, for the years 2013-2015.

REFERENCES

- [1] Pfeiffer F., Technological Aspects of Walking. *Walking: Biological and Technological Aspects*. Springer Wien New York, pp. 119-154, 2004.
- [2] Siciliano B. and Khatib O., *Handbook of robotics, Part G, legged robots*. Springer, pp.361-390, 2008.
- [3] Song S.M. and Waldron K.J., *Machines That Walk-The Adaptive Suspension Vehicle*. The MIT press, 1989.
- [4] Merlet J.P., *Parallel Robots*, Kluwer Academic Publishers, 2002.
- [5] Ceccarelli, M., *Fundamentals of Mechanics of Robotic Manipulation*, Kluwer Academic Publishers, 2004.
- [6] Ota Y., Inagaki Y., Yoneda K. and Hirose S., Research on a Six-Legged Walking Robot with Parallel Mechanism. In: *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, Victoria (Canada), Oct. 13-17, pp.241-248, 1998.
- [7] Sugahara Y., Endo T., Lim H. O. and Takanishi A., Design of a Battery-powered Multi-purpose Bipedal Locomotor with Parallel Mechanism. In: *Proc. IEEE Intl. Conf. on Intelligent Robots and Systems, Lausanne* (Switzerland), Sept. 30-Oct. 4Vol.3, pp.2658-2663, 2002.
- [8] Takanishi Laboratory. 2013. Multi-purpose Biped Locomotion WL-16RV. <http://www.takanishi.mech.waseda.ac.jp/top/research/parallel/index.htm>
- [9] Ceccarelli M. and Carbone G., A New Leg Design with Parallel Mechanism Architecture. In: *Proc. IEEE/ASME Intl. Conf. on Advanced Intelligent Mechatronics*, Singapore, Jul. 14-17, pp. 1447-1452, 2009.
- [10] Clavel R., DELTA, a fast robot with parallel geometry. In: *Proc. 18th Intl. Symp. on Industrial Robots*, Lausanne (Switzerland), Apr. 26-28, pp. 91-100, 1988.
- [11] Lee K. and Shah D. K., Kinematic analysis of a three degrees of freedom in-parallel actuated manipulator. In: *Proc. of IEEE Intl. Conf. Robotics and Automation Raleigh* (USA), Mar. 31-Apr. 1 3, Vol. 1, pp. 345-350, 1987.
- [12] Joshi S. and Tsai L.W., A comparison study of two 3-DOF parallel manipulators: one with three and the other with four supporting legs, *IEEE Trans. Robotics and Automation*, Vol. 19(2), pp. 200-209, 2003.
- [13] Carbone G. and Ceccarelli M., Legged Robotic Systems. *Cutting Edge Robotics, ARS Scientific Book*, Wien. pp.553-576, 2005.
- [14] Liu J., Tan M. and Zhao X.G., Legged robots-an overview. *Trans. of the Institute of Measurement and Control*, Vol. 29(2), pp.185-202, 2007.
- [15] Tsai L. W. and Joshi S. A., Kinematics and optimization of a spatial3-UPU parallel manipulator, *ASME J. Mech. Design*, Vol. 122, pp.439-446, 2000.

DESIGN AND OPERATION OF CASSINO HEXAPOD II

Franco Tedeschi* Daniele Cafolla* Giuseppe Carbone*

* *LARM: laboratory of Robotics and Mechatronics,
DICEM; University of Cassino and South Latium - Cassino, Italy
URL: <http://webuser.unicas.it/web/larm/larminindex.htm>*

ABSTRACT

This paper deals with the implementation of an hybrid walking machine called Cassino Hexapod II. The body and the control architecture of the robot are presented and operation features are also discussed. Dynamic simulations have been carried out by using a 3D Cad model in Solidworks and ADAMS environments. The operation capability of Cassino Hexapod II are described also by referring to controller firmware, processors and actuators. Main design parameters and operation features have been considered also for the gait simulation and implementation of Cassino Hexapod II. Experimental tests are reported in order to show feasibility and operational capability of proposed design.

Keywords: Design, Walking Machines, Low-Cost Control.

1 INTRODUCTION

Hexapod walking robots are mechanical vehicles that can walk on six legs. They have been widely studied for their significant advantages with respect to wheeled ones for walking over rough terrain. Hexapod robots have several benefits [1],[2]:

- hexapod robots are easy to maintain static stability on three or more legs;
- they have a great deal of flexibility in how they can move;
- hexapod robots are the most efficient one for statically stable walking;
- hexapod robots show robustness in case of leg faults;
- hexapods makes it possible for the robot to use one, two or three legs to work as hand and perform complex operations.

Hexapod walking robots, have attracted considerable attention in recent decades. Many studies have been implemented for hexapod walking robots but only in the recent past efficient walking machines have been conceived, designed and built with performances that are suitable for practical applications [3].

Hexapod robots have been developed in the last part of 20-th century around the world in research centers and universities as machines that can help the human being in dangerous or exhausting tasks, for example, landmine detection or rescue operation, like COMET IV [4], or forest harvesting, like the hexapod develop at Plustech Ltd (2002). Others examples of existed hexapod robots in the world are RHEX robot [5], the Adaptive Suspension Vehicle [6], Boadicea [7], SLAIR2 [8] and the ATHLETE [9].

Mobile robots can be structured of different types. First ones are based on crawlers or wheels and second ones are equipped with biologically inspired legs. This second type of walking machines can be slow and more difficult to design and operate with respect to the first ones. Nevertheless, legged robots are more suitable for rough terrain, where obstacles of any size can appear [10]. In fact, the use of wheels or crawlers limits the size of the obstacle that can be climbed to half the diameter of the wheels [11]. On the contrary, legged machines can overcome obstacles that are comparable with the size of the machine leg [12]. There is also a third type of walking machines that is called hybrid robot since it has legs and wheels at the same time. This type of walking machines may range from wheeled devices to true walking machines with a set of legs to overcome particularly difficult obstacles, or wheels to enhance the speed when moving on flat terrain.

The aim of this paper is to propose the design of a novel walking machine having low-cost and user-friendly features. This novel hybrid walking machine can be seen as

Contact authors: Giuseppe Carbone¹, Franco Tedeschi²
Daniele Cafolla³

¹ carbone@unicas.it

² franco.tedeschi@unicas.it

³ cafolla@unicas.it

A preliminary version of this paper has been presented at RAAD 2013 Conference in Portoroz (Slovenia).

an evolution of the Cassino Hexapod robot [13][14] it is composed of six legs having a modular anthropomorphic architecture with wheels as feet at its extremity.

2 THE ATTACHED PROBLEM

The proposed design has been conceived by using low-cost components with the basic requirements as:

- have a robust simple mechanical design;
- have a modular design that can be used for robots with different number of legs;
- operate with an easy flexible programming;
- have low-cost both in design and operation.

Expected field of application for this prototype is the inspection and operation in non-accessible sites, a detailed description of the proposed application can be found in [15]. Earth provides an extremely large variety of terrains, but the obstacles encountered in the environment application can be classified basically in 4 group as shown in Fig. 1:

- a step, defined basically by the height (H);
- a step down defined basically by the height (H);
- a crest, characterized by 2 parameters, width (W) and height (H);
- a ditch, characterized by its width (W).

Figure 2 shows the expected operation capability of the proposed walking machine over the above mentioned typical obstacles. The proposed design can theoretically overpass a step obstacle in step up and step down operation with $H=64$ mm; a crest obstacle with $H=64$ mm, $W=110$ mm, and a ditch with a width $W=110$ mm; expected surface inclination: $\pm 10\%$. The main obstacle size are $H \cong 40\%$ and $W \cong 65\%$ of the robot leg .

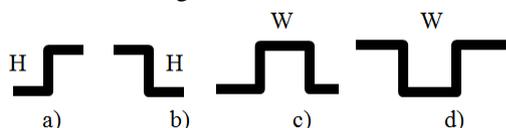


Figure 1 Possible obstacles for Cassino Hexapod II design: a) step b) step down c) crest d) ditch.

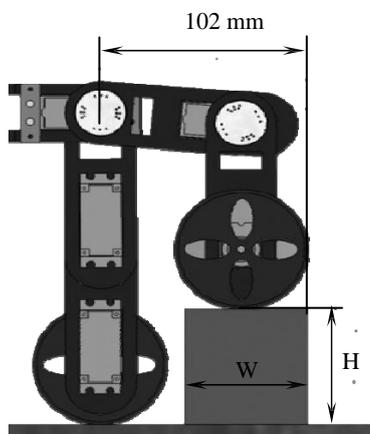


Figure 2 A detail of Cassino Hexapod II leg showing its maximum step height and width.

3 MECHANICAL DESIGN

In this section, we introduce our six-legged robot called Cassino Hexapod II. This walking machine can fit into a cube of $0.4\text{m} \times 0.3\text{m} \times 0.2\text{m}$ and it has an overall weight of 20N . It can carry on-board its own control board and battery; in this case, the robot weight is about 22N . Robot configuration is presented in Fig. 3.

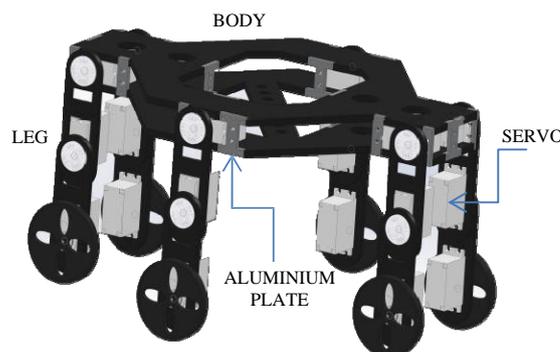


Figure 3 3D CAD model of Cassino Hexapod II structure.

The low cost design is intended to be achieved by using low-cost commercial components into a suitable design for the whole system: servomotors as actuators, thermoplastic material to make robot body, and a commercial control board as robot controller. The body of Cassino Hexapod II is composed of various modules made of POM (Polyoxymethylene), a commercial thermoplastic that has high stiffness, low friction and excellent dimensional stability. The robot body structure is composed of two plates that are connected with screws and aluminium plates. The aluminium plates allow also to fix the servomotors to the robot body. Mechanical design of the new hexapod operations have been carried out by using the 3D-CAD model of Fig.3. Simulations have consisted in investigating basic robot performances in a virtual environment in order to check the design feasibility before prototyping. SolidWorks environment has been used due to its convenient features for the structure analysis to check the feasibility of the real prototype. The model has been elaborated by introducing each component with its specific characteristics in terms of material, mass, density, shape and mechanical design. Components such as screws and nuts have been considered as made of steel alloy. Servomotor have been modelled introducing the mechanical properties reported in their respectively datasheets. Each of the legs has 3 DoFs: two of them have a movement that has a range between -90° and 90° that allow the robot to overpass obstacles, the other one allow to move the wheel in a full range rotation. This assembly solution as based on previous experience that are reported in [10],[12],[16]. The described leg has a total weight of about 1.5N . To make the assembly scheme clear an exploded view of the leg is shown in Fig. 4. The maximum dimensions of the leg and of the main components are shown in Figure 5. The leg measures 167mm in length,

35mm in width and 5mm in depth. It is worth noting that the leg is composed of two link. Each module contains two commercial servomotors; the servo output shafts are used in order to connect a module to another one. The diameter of wheels is 66mm.

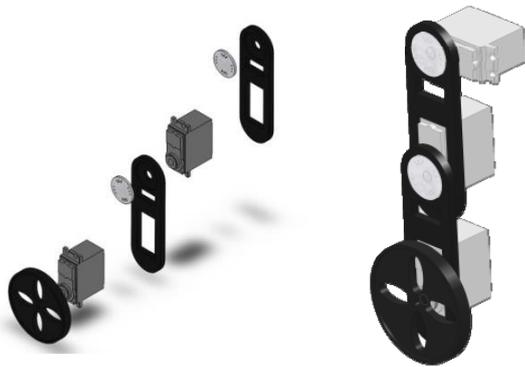


Figure 4 A exploded view of Cassino Hexapod II leg.

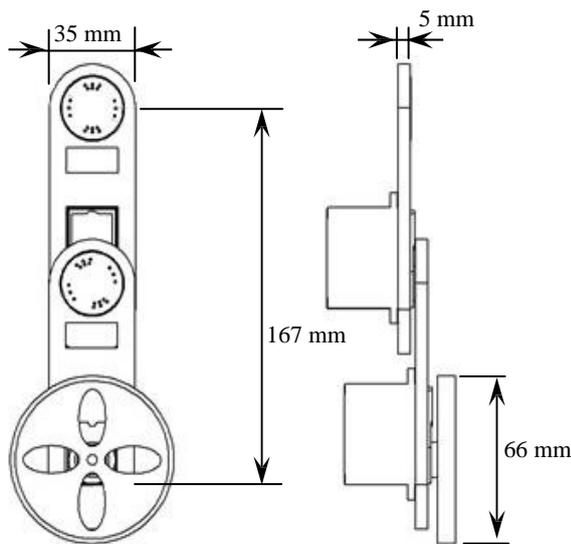


Figure 5 A CAD model of the proposed leg.

Additional components are needed for the extremity modules. In particular, the support motor requires additional fixing parts in order to connect the leg to the robot body, and the leg extremity link requires an additional wheel. The adopted modular design of one leg has been used as basic component for the new Cassino Hexapod II. Then, six legs are assembled with the main body as shown in Figure 6. In particular, six legs have been connected to a suitable plates body in order to build the hexapod robot. On the other hand the main plates as the one shown in Fig. 7 are 258mm in long, 364mm high and 6mm in thick.



Figure 6 Exploded view of Cassino Hexapod II

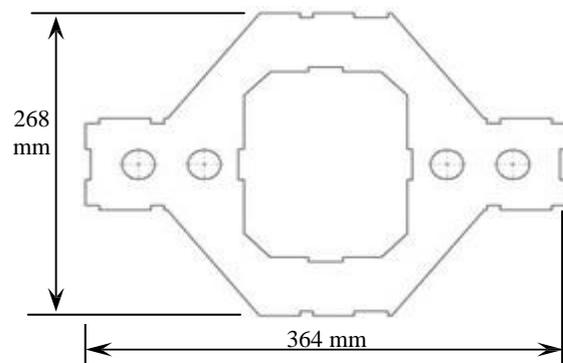


Figure 7 A CAD model of Cassino Hexapod II plate.

Making a structure analysis using Von Mises procedure with the help of CATIA Analysis environment and considering the distributed weight on each single leg, it has been noticed that the effects of the own weight is a negligible as shown in Figure 8. The most stressed part is in the shoulder connector although the its maximum stress is still within the operation range of the chosen material. So these measures were considerate suitable to realize a real functional prototype.

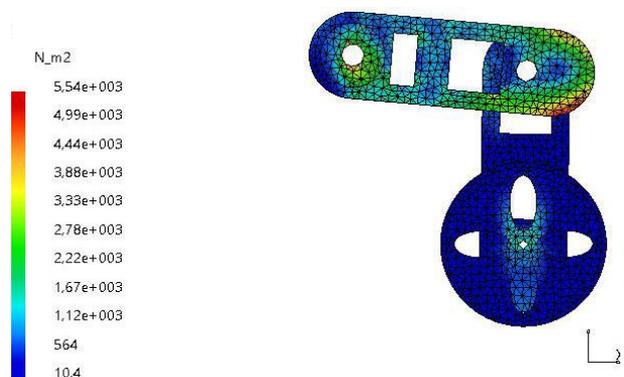


Figure 8 Example of Von Mises procedure on Cassino Hexapod II.

4 OPERATIONAL FEATURES

The control system of a hexapod robot has been simplified by reducing unnecessary factors to the minimum and to design and realize the best effective gaits and body structures of the latter. Suitable commercial low-cost hardware components have been properly selected.

4.1 THE PROPOSED CONTROL HARDWARE

Overall motors configuration of the model of a designed hexapod walking robot can be schematized as in Fig.9 It is necessary to install small and light type of motors for the joint actuator because of restriction of space and mobility. The RC servomotors are suitable for controlling mobile robots as they are small and light, they take only small room and low cost. The Cassino Hexapod II has 18 RC servomotors, 12 of which are used for giving mobility to legs and the others are used for rollover operation. A suitable commercial low-cost control board have been selected. A customized interface to servomotors and controller have been implemented.

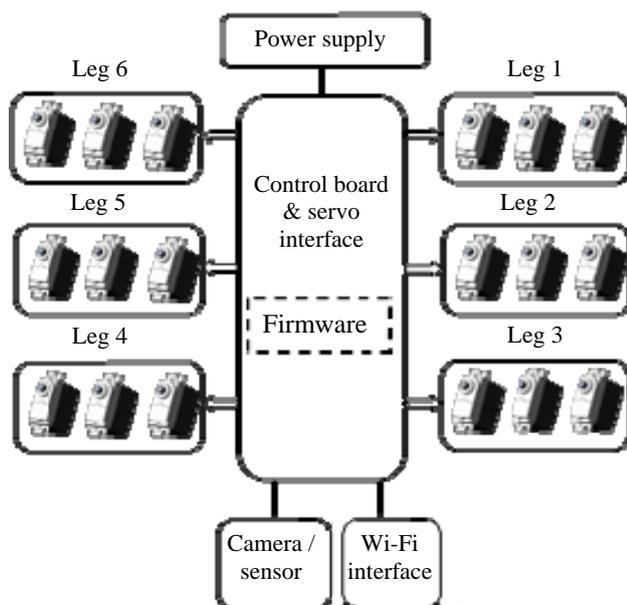


Figure 9 Overall configuration of Cassino Hexapod II.

Most of mobile robots have to work autonomously, so they must have local batteries for the controller and the motors; Cassino Hexapod II has a Li-Po battery 7.4V- 2200 mAh, and it can work more than 4 hours. A radio wireless communication link is used to exchange information between a robot and a remote controller. The high level remote control have been developed in a Java environment: it allows to select a motion and allow task planning between a Wi-Fi network using pc or smart devices.

4.2 SERVO MOTORS

A servomotor is composed of an electric motor, a reduction gearbox, a position feedback system for the axis output, and

an electronic control for close-loop positioning of the output. The control of the servomotor is achieved by means of a proper PWM modulation. The main features of the adopted servomotor (Hitec model HS-322HD) are:

- input power voltage from 4.8V to 6V;
- output torque from 0.3 to 0.37Nm (nominal input);
- output axis rotation from 0 to 180 deg.;
- operating speed from 0.15 to 0.19 sec/60 deg. at no load;
- PWM pulse signal ranging from 0.6 ms to 2.4 ms;
- idle current from 7.4 mA to 7.7 mA;
- running current from 160 mA to 180 mA at no load;
- weight 0.043 kg.

The output shaft of a standard servomotor usually can rotate from 0 to 180 deg.. This rotation range is suitable for actuation of leg joints of the proposed hexapod robot. But, the rotation of wheels requires a continuous rotation of the input axis. A modified servomotor is required for this purpose such as Parallax, continuous rotation servo. The main features of the adopted continuous rotation servo (Parallax mod 900-00008) are:

- input power voltage from 4.8V to 6V;
- output torque from 0.27 Nm (nominal input);
- output axis speed from 0 to 50 RPM.;
- PWM pulse signal ranging from 1.3 ms to 1.7 ms;
- weight 0.043 kg.

The above mentioned motor is suitable for the operation of the wheels. The angular position of the wheel can be obtained by a proper software routine.

4.3 CONTROL BOARD

A suitable commercial low-cost control board should be capable of operating at least 18 servomotors. Additionally, it should have a significant number of extra Inputs/Outputs for managing additional external sensors and motors in a modular architecture. Arduino ATmega 2560 can be a suitable choice having the above-mentioned features. In particular, Arduino ATmega 2560 has:

- 54 IO pins, whose 14 can be used for PWM;
- 16 analogic inputs;
- a flash memory of 128 kbytes;
- an EEPROM of 4 kbytes;
- a clock speed of 16 Mhz.

The Arduino ATmega 2560 board, contains all the hardware components for the operation of the embedded microcontroller. It is based on an open-source multi-platform integrated development environment that can be operated via Linux, Apple Macintosh and Windows. This feature allows an user-friendly software implementation of many different customized input/output operations. The programming of the Arduino can be achieved by means of source codes written in C/C++.

The remote wi-fi interface can be achieved by means the Arduino wi-fi shield. The wi-fi shield allow an Arduino board to connect to the internet using 802.11 wireless specification.

4.4 SERVMOTORS INTERFACE

The adopted solution provides a direct connection of each servomotors control signal to a digital output pins of the Arduino control board. It should be noted that each servomotors power has to be connected to an external power supply, since Arduino cannot provide currents totaling more than 400 mA. The proposed drive mode is the most simple and economical way to drive servomotors with Arduino control board. This solution has been tested by developing a specific board (shield) for the Arduino card. Figure 10 shows the built board for 18 servomotors. The geometrical sizes of the proposed Arduino shield have been defined to fully match with Arduino Mega 2560 control board pin-out as shown in Fig.10b). Main attention has been addressed to the cross sections of the printed circuit board that has been set up to be compatible with the needed high currents of the servomotors. Figure 10a) shows the top side of the built direct drive Arduino shield with the connectors to the 18 servomotors and the cable for the external power supply.

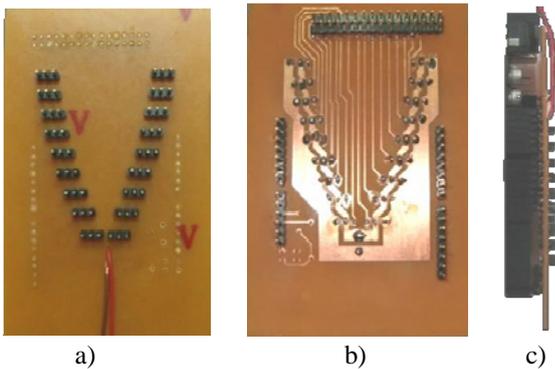


Figure 10 The Arduino servo shield built at LARM: a) top side to servomotors b) bottom side to Arduino; c) side view: shield assembled on Arduino Mega 2560.

5 THE PROPOSED OPERATION STRATEGIES

The Arduino control board is connected to a PC through a standard USB cable; programming can be achieved on the PC thanks to the Arduino IDE. Then, the program can be easily downloaded on the flash memory and the cable can be disconnected. A commercial Li-Po battery 7.4V provides the power supply for both the control board and servomotors, between customized voltage regulator. A remote Wi-fi control is used for operating the hexapod. Operation strategies can be structured of two basic types: first one is wheeled mode and second one is walking mode. The wheeled foot gives the characteristic of moving as a vehicle to the hexapod robot. In wheeled mode, it should be noted that the motors of the right side and the left side of the robot are driven with opposite rotation. In fact, due to the axial symmetry a motor rotating in clockwise direction for the left side of the hexapod produces the same motion effect of a motor that is rotating in counterclockwise direction on the right side. In

fact, forward and backward motions can be achieved when all the wheels move at same speed in the same clockwise or counterclockwise direction, respectively. Instead, the turning operation can be achieved by operating the wheels of left side and right side in opposite directions. The selection of operation type can be obtained through the Wi-Fi control by PC selecting a proper one. A hexapod consists of six legs and proper movement of the robot is possible by properly synchronizing the motion of all the legs. Thus, programming the movement of one leg is very important for the successful movement of the robot according to [17]. Various operation strategies have been analyzed and implemented on the built prototype. In particular, flow-chart analysis of a single leg operation and continuous forward walking, of the whole hexapod robot are reported in Figs.11 and 12. In Figs. 11 a descriptive flow-chart diagrams are reported for the forward and backward operations of one leg. This operation strategy for one leg is the basic operation strategy that will be used in executing other complex motion tasks.

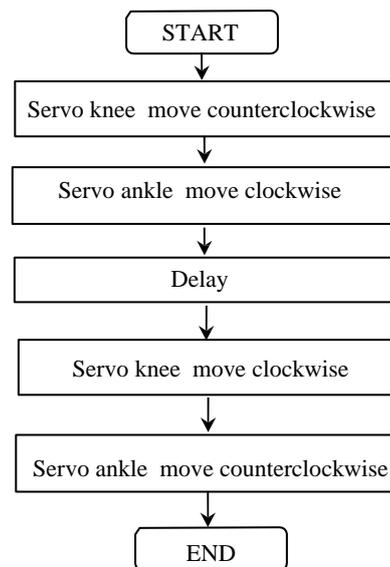


Figure 11 Flow-chart for single leg operation.

Figures 12 shows the flow chart diagram and a footfall formula representation for a forward walking operation of the hexapod robot.

In the forward walking operation strategy a tripod gait has been considered as reported in the picture of Fig.13. In fact, for maintaining the stability of the robot, at least three legs should be in contact with the ground simultaneously [18],[19]. In a tripod gait, the front and rear leg of one side and the middle leg of another side perform their swing movements at the same time. A delay is introduced between each step in order to ensure proper coordination and phase difference among the motion of legs. This delay was chosen after experimental tests and is also a function of the operating speed. The operation of each leg has the same logic that is reported in the flow chart above mentioned.

5.1 BASIC FIRMWARE DEVELOPMENT

The Arduino development environment contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions, and a series of menus.

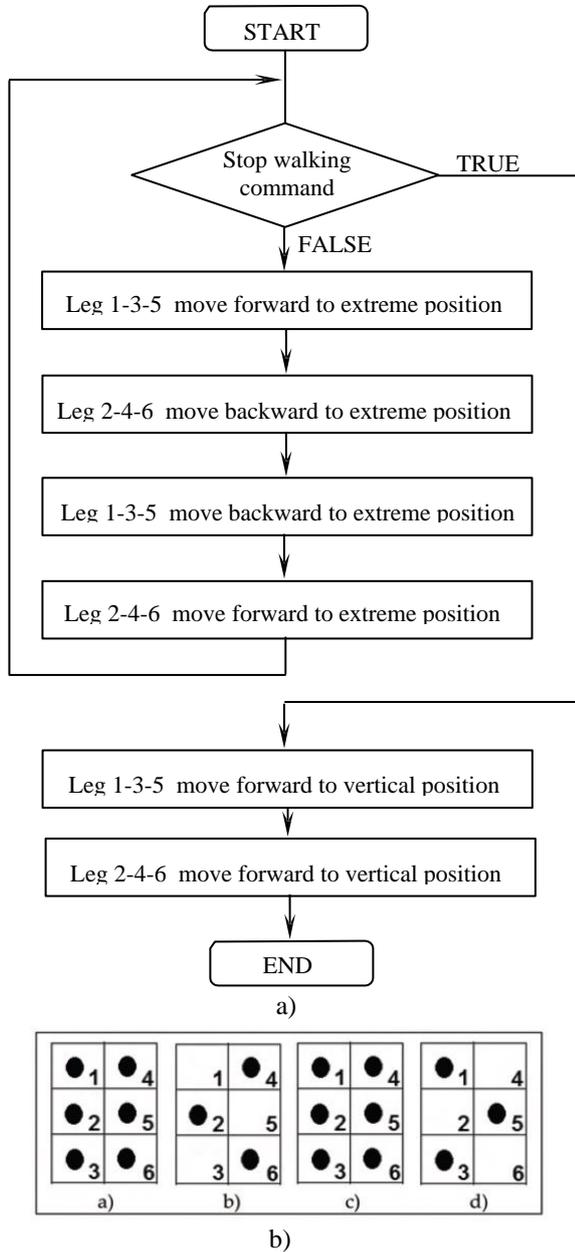


Figure 12 a) Scheme of tripod gait of Cassino Hexapod II
 b) Footfall formula representation : black circles stands for the legs in contact with the ground surface; arrow indicates the moving forward direction.

It connects to the Arduino hardware to upload programs and communicate with them. The operation of the proposed control architecture has been achieved by developing a proper firmware of Arduino control board, as based on the Arduino Servo library. This servo library provide features for a user-friendly operation of up to 48 servomotors without requiring any setting of PWM. The operation strategy for one servomotor is the basic

operation that will be used in executing other complex motion task. The first step is to include the Servo library (header file) in the proposed Arduino project. The second step is to define a Servo object. In the third step the attach function allows to connect the servo object to a specific pin of the Arduino control board. Moreover, the user can define the min parameter that is the pulse width (in microseconds) that is related with the minimum angle of the output shaft position.

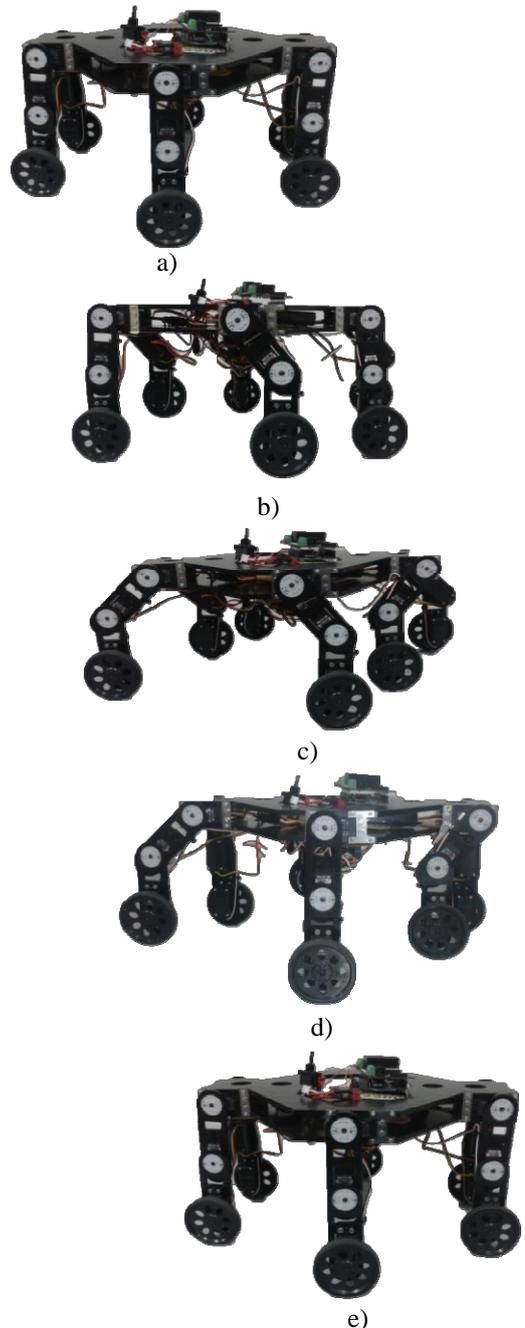


Figure 13 Tripod gait on Cassino Hexapod II a) all legs are in contact with ground; b) legs 1-3-5 are lifted and moved forward; c) all legs come in contact with ground; d) legs 2-4-6 are lifted and moved backward e) all legs are in contact with ground.

The user can also define the max parameter that is related with the maximum angle of the output shaft position (180 deg.). The last step is the write() function. This function sets the angular position of a standard servo (in degrees) so that the corresponding servomotor moves its shaft to this angular position. In the case of continuous rotation servomotors the write function will set the desired speed of the servomotor. In this case a value 0 refers to a full-speed in clockwise direction; 180 refers to a full-speed in counterclockwise direction; 90 refers to a standstill configuration.

6 CONCLUSIONS

This paper deal with the design, construction and operation of an hybrid walking machine called Cassino Hexapod II. The mechanical design of robot is based on proper low-cost commercial components and easy operation features. The proposed control architecture has required the design of a proper servomotor control shield and a user-friendly software library that allows the operation of the required 18 servomotors. The feasibility of the proposed new design and easy operation modes have been tested experimentally. Results have shown a robust system operation, with limited power consumption in wheeled operation modes an even lower power consumption is obtained as suitable gait planning. A hybrid robot Cassino Hexapod II can also use wheels for locomotion on flat ground. In this way robot can move fast without consuming a lot of energy. Results of the experimental test also demonstrate that commercial lithium batteries can provide a suitable power supply to Cassino Hexapod II for at least 4 hours of continuous operation.

REFERENCES

- [1] Preumont, A., Alexandre P., Ghuys D., Gait analysis and implementation of a six leg walking machine. In: *Proc. 91 ICAR., Fifth International Conference on Advanced Rob.*, 1991.
- [2] Preumont, A., An investigation of the kinematic control of a six-legged walking robot, *Mechatronics*, Volume 4, Issue 8, pp. 821-829, 1994.
- [3] Silva, M.F., Machado J.A.T., A historical perspective of legged robots, *Journal of Vibration and control*, Vol.13, n.9, 10 pp. 1447-1486, 2007.
- [4] Oku, M., Yang H., Paio G., Harada Y., Adachi K, Barai R., Nonami K., Development of Hydraulically Actuated Hexapod Robot COMET-IV-The 1st Report: System Design and Configuration. In: *Proc. of the 2007 JSME Conference on Robotics and Mechatronics*, 2A2-G01, 2007.
- [5] Saranlı, U., Buehler M., Koditschek D.E., RHex – A Simple and Highly Mobile Hexapod Robot. *The International Journal of Robotics Research*. Vol. 20; No. 7; pp. 616 – 631, 2001.
- [6] Song, S.-M., Waldron K. J., *Machines That Walk: The Adaptive Suspension Vehicle*. The MIT Press, 1989.
- [7] Binnard M., Design of a Small Pneumatic Walking Robot, *M. Sc. Thesis*. Massachusetts In. of Technology Department of Mechanical Engineering, 1995.
- [8] Dzhantimirov J., Palis A., Schmucker B., Telesh M., Zavgorodniy B., HIL/SIL by Development of a Six Legged Robot SLAIR2. In: *Proc. International Conference on Climbing and Walking Robots CLAWAR*, Singapore, pp. 652-661, 2007.
- [9] Heverly, M. and Matthews J., A Wheel-on limb Rover for Lunar Operation. In: *Proc. Symposium on AI, Robotics and Automation in Space*, Los Angeles, pp. 340-344, 2008.
- [10] Carbone G. and Ceccarelli M., A Mechanical Design of a Low-Cost Easy-Operation Anthropomorphic Wheeled Leg for Walking Machines, *International Journal Robotica Manager*, 9(2), pp. 3 – 8, 2004.
- [11] Chakraborty N. and Ghosal A., Kinematics of Wheeled Mobile Robots on Uneven Terrain, *Mechanism and Machine Theory*, 39, pp. 1273– 1287, 2004.
- [12] Carbone G. and Ceccarelli M., Legged Robotic Systems, *Cutting Edge Robotics ARS Scientific Book*, Wien, pp.553-576, 2005.
- [13] Carbone G., Shrot A., Ceccarelli M., Operation Strategy for a Low-Cost Easy-Operation Cassino Hexapod, *Applied Bionics and Biomechanics*, Vol.4, N.4, pp.149–156, 2007.
- [14] LARM homepage. 2012. <http://webuser.unicas.it/weblarm/larmiex.htm>
- [15] Cigola M., Pelliccio A., Salotto O., Carbone G., Ottaviano E., Ceccarelli M., Application of Robots for inspection and restoration of Historical sites. In: *Proc. International Symposium on Automation and Robotics in Construction*, Ferrara, paper 37, 2005.
- [16] Nava Rodríguez N.E., Carbone G., Ceccarelli M., Moreno Lorente L.E., Design Evolution of Cassino Hexapod Robot. In: *Proc. 10th Biennial ASME Conference on Engineering Systems Design and Analysis ESDA2010*, Istanbul, paper n. ESDA2010-24020, 2010.
- [17] Garca-Lopez M.C., Gorrostieta-Hurtado E., Vargas-Soto E., Ramos-Arregun J.M., Sotomayor-Olmedo A. and Moya Morales J.C., Kinematic analysis for trajectory generation in one leg of a hexapod robot, *Procedia Technology*, Volume 3, pp. 342-350, 2012.
- [18] Wang,Z., X. Ding, A. Rovetta, Giusti A., Mobility analysis of the typical gait of a radialsymmetrical six-legged robot, *Mechatronics* Vol.21, pp. 1133-1146, 2011.
- [19] Zielinska T., Heng J., Development of a walking machine: mechanical design and control problems, *Mechatronics*, Vol. 12, n. 5, pp. 737-754, 2002.

DEVELOPMENT OF A HYBRID PATH/FORCE CONTROL ARCHITECTURE FOR AN ASSISTING ROBOT

Benjamin Großmann Thomas Schlegl

Regensburg Robotics Research Unit
Faculty of Mechanical Engineering
Ostbayerische Technische Hochschule Regensburg

ABSTRACT

For human machine interaction a control architecture for the force controlled guidance mode of a five axis robot is developed. This enables a human operator to move the robot easily by applying forces at the end-effector. The position and the path can then be tracked and stored for a later reuse. The control architecture is also used in a hybrid mode where it is possible to control chosen workspace coordinates with a path controller and others with a force controller. The hybrid mode is especially used to cut tissue material with a scalpel which is mounted at the robots effector.

Keywords: Robotics, force control, hybrid control, assisting robot

1 INTRODUCTION

The Regensburg Robotics Research Unit develops an assisting robot named CARo5X (Cleanroom Application Robot with Five Moving Axes, as seen in Figure 2) inside the research project ManuCyte. The aim of the project is the development of a plant for the automated cell and tissue cultivation in an industrial scale [1] [2]. The research in human-robot interaction has become special interest over the last two decades. A wide range of applications can be improved by combining human operators with assisting robots. For example, rescue robots can be used to search for buried people [3] or household robots can assist handicapped or old people [4]. The assisting robot is working together with the human operator inside the so called hybrid workplace (see Figure 1). The aim of the collaboration is to benefit from the combination of an automation system and the skills of the human operator as described from Heiligensetzer [5]. The tasks of the robot are on the one hand to supply the human operator with bioreactors from the plant and vice versa and on the other hand to cut tissue material. Two of the requested work packages for the assisting robot are realized with a force control scheme. The mentioned work packages are the force controlled guidance and the cutting of tissue material.

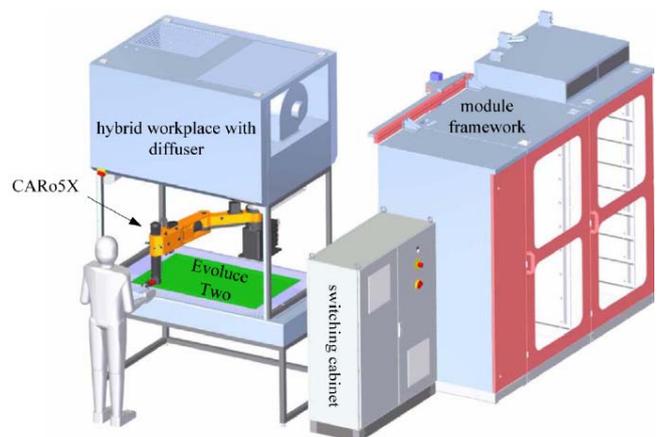


Figure 1 CARo5X inside the hybrid workplace.

With the force controlled guidance it is possible to move the robot by applying forces at the end effector. The position and the path can be tracked and then repeated by the robot [6]. The cutting of tissue material is realized by hybrid force/path control architecture. Selected workspace coordinates are controlled by a path control architecture and other workspace coordinates are controlled by a force control architecture simultaneously [7]. Further applications of the hybrid control architecture outside this project are table wiping or automatic piston insertion into cylinder blocks [8]. Within this paper the force control architecture is presented and validated via simulation and experimental results.

Contact author: Benjamin Großmann¹, Thomas Schlegl²

¹benjamin2.grossmann@oth-regensburg.de

²thomas.schlegl@oth-regensburg.de

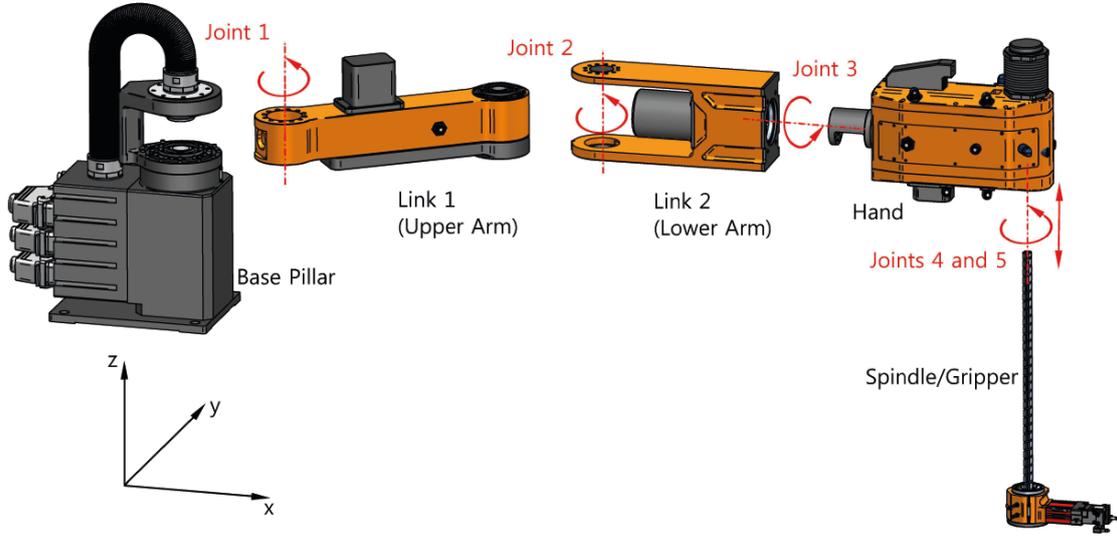


Figure 2 Cleanroom assisting robot with five moving axes (CARo5X).

2 CONTROL ARCHITECTURE

This chapter describes the control architecture of the assisting robot. The control architecture consists of a feed-forward computed torque controller and the outer path/force control loop.

2.1 COMPUTED-TORQUE-CONTROL

The dynamic model of the CARo5X is derived via the Lagrange formalism as shown in [9]. One obtains the differential equation of the robot

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) + \mathbf{r}(\mathbf{q}) = \boldsymbol{\tau}_n. \quad (1)$$

Here $\mathbf{M} \in \mathbb{R}^{n \times n}$ is the inertia matrix, $\mathbf{c} \in \mathbb{R}^{n \times 1}$ are gyroscopic forces, $\mathbf{g} \in \mathbb{R}^{n \times 1}$ is the vector of gravitational influences, $\mathbf{r} \in \mathbb{R}^{n \times 1}$ is vector of friction torques, $\boldsymbol{\tau} \in \mathbb{R}^{n \times 1}$ are external forces and torques and $\mathbf{q} \in \mathbb{R}^{n \times 1}$ are actual joint angles.

With Eq. (1) one can calculate the necessary torques for each drive. Feed-forwarding these torques and combining the feed-forward loop with the plant model one obtains a linear system

$$\mathbf{q}(t) = \iint (\mathbf{u}(t) + \mathbf{z}(t)) dt \quad (2)$$

as new plant with the controller's output $\mathbf{u} \in \mathbb{R}^{n \times 1}$ as input variable. The model inaccuracies which especially occur in the term of friction are taken into account via the disturbance term $\mathbf{z} \in \mathbb{R}^{n \times 1}$ as additional input to the plant [10].

2.2 PATH-CONTROL ARCHITECTURE

The path control architecture consists of a simple and well known PID-controller and a path-planning with polynomials of order three. The differences in position and in velocity

are inputs for the PID-Controller. The desired acceleration is added to the controllers output. With the control deviation

$$\mathbf{e} = \mathbf{q}_d - \mathbf{q} \quad (3)$$

the path-controller output is calculated by

$$\mathbf{u}_p = \ddot{\mathbf{q}}_{d,P} + \mathbf{K}_I \int \mathbf{e} dt + \mathbf{K}_P \mathbf{e} + \mathbf{K}_D \dot{\mathbf{e}} \quad (4)$$

With Eq. (2) and Eq. (3) one obtains the differential equation

$$\ddot{\mathbf{e}} + \mathbf{K}_D \dot{\mathbf{e}} + \mathbf{K}_P \mathbf{e} + \mathbf{K}_I \mathbf{e} = -\dot{\mathbf{z}} \quad (5)$$

for the deviation of the path-control loop. Eq. (5) shows that for constant disturbances, no remaining deviations exist due to the fact that the disturbance \mathbf{z} only occurs with its first time derivation $\dot{\mathbf{z}}$

2.3 TRANSFORMATION OF FORCES AND TORQUES

Since the force-/torque sensor delivers a vector with six elements

$$\mathbf{f}_s = (f_x \ f_y \ f_z \ T_x \ T_y \ T_z)^T \quad (6)$$

defined in the sensors frame (see Figure 3) and the workspace of the CARo5X only consists of five workspace coordinates

$$\mathbf{w} = (x_0 \ y_0 \ z_0 \ \alpha \ \beta)^T \quad (7)$$

$$\alpha = \angle(z_0, z_E) = q_3 \quad (8)$$

$$\beta = \angle(z_2, x_E) = q_4 \quad (9)$$

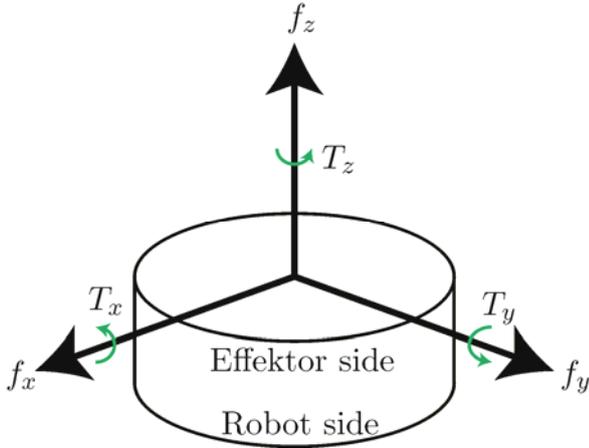


Figure 3 Force/torque sensor with sensor frame.

it is necessary to reduce the number of elements in f_s and to transform them into the base frame. The forces in f_s can be transformed simply by using the rotation matrix R_S^0

$$\begin{pmatrix} f_x^0 \\ f_y^0 \\ f_z^0 \end{pmatrix} = R_S^0 \begin{pmatrix} f_x^S \\ f_y^S \\ f_z^S \end{pmatrix} \quad (10)$$

The torque about the sensors z-axis T_β is transformed to a torque in the direction of β , which is per definition equal to q_4

$$T_\beta = T_z. \quad (11)$$

The achieved behaviour is then, that the robots spindle rotates if a torque about the spindle is applied. The torques T_x and T_y are used to influence the work space coordinate α . As one can see in Figure 4 the influence of T_x and T_y on α depends on q_4

$$T_\alpha = T_x \cos(q_4) + T_y \sin(q_4). \quad (12)$$

Combining Eq. (11) and Eq. (12) one gets the matrix

$$\begin{pmatrix} T_\alpha \\ T_\beta \end{pmatrix} = \underbrace{\begin{pmatrix} \cos(q_4) & \sin(q_4) & 0 \\ 0 & 0 & 1 \end{pmatrix}}_P \begin{pmatrix} T_x \\ T_y \\ T_z \end{pmatrix}. \quad (13)$$

Again, combining Eq. (10) and Eq. (13) yields to the transformation matrix $X \in \mathbb{R}^{5 \times 6}$

$$f_0 = \underbrace{\begin{pmatrix} R_S^0 & 0^{3 \times 3} \\ 0^{2 \times 3} & P \end{pmatrix}}_X \begin{pmatrix} T_x \\ T_y \\ T_z \end{pmatrix}. \quad (14)$$

The desired values in f_d are also defined in the base system with five components so that no transformation is needed here.

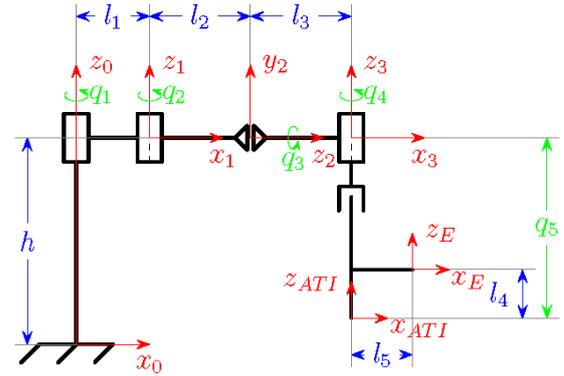


Figure 4 Kinematic chain of the assisting robot CARo5X.

2.4 FORCE-CONTROL ARCHITECTURE

The force control architecture gives desired velocities in workspace coordinates which are proportional to the applied forces. Constant disturbances must not lead to a permanent control deviation. The first requirement is achieved by using a damping model with the tuneable damping matrix $D \in \mathbb{R}^{5 \times 5}$ to calculate the velocities in workspace coordinates

$$\dot{w}_d = D^{-1} \Delta f \quad (15)$$

and transformation of these velocities via the Jacobian matrix into joint space

$$\dot{q}_d = J^{-1}(q) \dot{w}_d = J^{-1}(q) D^{-1} \Delta f. \quad (16)$$

The controller is designed as PI-controller with an additional damping. To increase the damping of axes the angle velocity is factorized with K_D and added to the PI-controllers output. The differential equation of the so designed control loop

$$\ddot{q} + K_D \dot{q} + K_P q = z + K_P \int J^{-1} D^{-1} \Delta f + K_I \int \int J^{-1} D^{-1} \Delta f dt \quad (17)$$

gives a nonlinear system of third order. The Jacobian depends on the joint angles q . The differential equation shows that the Hurwitz criterion is not fulfilled if K_D is not present ($K_D = 0$) so that the system would not be stable. Also, if the integral part is not present ($K_I = 0$)

$$\ddot{q} + K_D \dot{q} + K_P q = z + K_P \int J^{-1} D^{-1} \Delta f dt + K_I \iint J^{-1} D^{-1} \Delta f dt \quad (18)$$

constant disturbances z would yield to remaining control deviations. With the control deviation

$$\dot{e} = \dot{q}_d - \dot{q} = J^{-1} D^{-1} \Delta f - \dot{q} \quad (19)$$

the force-controllers output is calculated by

$$u_f = K_I \int e dt + K_P e + K_D \dot{e} \quad (20)$$

which also has PID-structure.

2.5 STATIC BEHAVIOR

To examine the static behaviour we assume that the applied forces are constant, the disturbances are constant and the velocities are small. Removing all terms in Eq. (17) which occur time derivative yields to

$$\mathbf{K}_I \mathbf{q} = \mathbf{K}_P \mathbf{J}^{-1} \mathbf{D}^{-1} \Delta \mathbf{f} + \mathbf{K}_I \int \mathbf{J}^{-1} \mathbf{D}^{-1} \Delta \mathbf{f} dt. \quad (21)$$

Considering the constant forces

$$\frac{d \Delta \mathbf{f}}{dt} = \mathbf{0} \quad (22)$$

Eq. (21) can be written as

$$\mathbf{K}_I \dot{\mathbf{q}} = \mathbf{K}_I \mathbf{J}^{-1} \mathbf{D}^{-1} \Delta \mathbf{f} \quad (23)$$

$$\mathbf{J} \dot{\mathbf{q}} = \mathbf{D}^{-1} \Delta \mathbf{f} \quad (24a)$$

$$\dot{\mathbf{w}} = \mathbf{D}^{-1} \Delta \mathbf{f}. \quad (24b)$$

It becomes apparent that the workspace velocity is proportional to the applied forces as requested. Furthermore constant disturbances do not lead to a permanent control deviation.

2.6 PARAMETERIZATION

Hence the axes of the robot are with the computed torque control not coupled, the control parameters can be calculated for a SISO system. The parameterization of the control parameters K_P , K_I and K_D is done by pole placement. With the real and negative poles $p_i = \{-10, -12, -14\}$ one obtains an asymptotically stable system with the control parameters $K_P = 428$, $K_I = 1680$ and $K_D = 36$. These parameters are used for simulation and are tuned during the experiments.

2.7 PATH/FORCE CONTROL ARCHITECTURE

Both, the force control architecture and the path control architecture result in a controller output $\mathbf{u}_f \in \mathbb{R}^{5 \times 1}$ and $\mathbf{u}_p \in \mathbb{R}^{5 \times 1}$ for each joint. For the hybrid path/force control architecture it is necessary to combine these two outputs. Therefore the selection matrices

$$\mathbf{S}_f = \text{diag}(0 \ 0 \ 0 \ 0 \ 1), \quad (25)$$

$$\mathbf{S}_p = \mathbf{I} - \mathbf{S}_f = \text{diag}(1 \ 1 \ 1 \ 1 \ 0) \quad (26)$$

are introduced and the combined controller output is calculated by

$$\mathbf{u} = \mathbf{S}_f \mathbf{u}_f + \mathbf{S}_p \mathbf{u}_p. \quad (27)$$

Use of Eq. (25) and Eq. (26) makes the fifth joint, which is the translational motion of the spindle, force controlled and joints one to four path controlled. Since cutting of tissue material is mostly done in a SCARA configuration of the robot, the fifth joint variable is equivalent to the z-coordinate in workspace.

2.8 AVOIDING SINGULARITIES

Since we use the inverse Jacobian matrix in Eq. (17), kinematic singularities need to be properly avoided. The singularities are listed in Tab.1.

Table 1: Singular configurations of CARo5X robot.

A	$q_2 = 0 \cap q_3 = 0$
B	$q_3 = \pm \pi/2$
C	$q_2 = 0 \cap q_5 = 0$

The only relevant singular configuration is configuration A due to the angular limitations of q_3 and q_5

$$-\frac{\pi}{3} \leq q_3 \leq \frac{\pi}{3}, \quad (28)$$

$$q_5 \leq -0.1 \text{ m}. \quad (29)$$

On account of the setup of the hybrid workplace the robot only has to operate with

$$q_2 \geq 0 \quad (30)$$

so that we can avoid the singularity by ensuring that Eq. (30) is strictly fulfilled. The avoidance of singularity is implemented as shown in Figure 5. A step-down function is implemented, that saturates the desired velocity when

$$|q_2| < 2 \text{PRM_AngQSingFrc}. \quad (31)$$

And finally stops if

$$|q_2| < \text{PRM_AngQSingFrc}. \quad (32)$$

The parameter `PRM_AngQSingFrc` is set to $0.05 \text{ rad} \cong 2.86^\circ$. It is always possible to move the robot away from the singular configuration since Eq. (31) and Eq. (32) are coupled to the desired velocity. If the desired velocity has another sign than the joint angle a motion is possible.

3 SIMULATION

The simulation is done inside the "RRRU simulation environment". The simulation environment has been developed as software structure by the Regensburg Robotics Research Unit using Matlab/Simulink. This allows the developer also to use the simulation environment to generate code for a real-time system. For the simulation we use the dynamic model of the CARo5X which is derived via the Lagrange formalism. The dynamic model is used to simulate the robots behaviour and used inside the computed torque control to calculate the torque for the feed-forward loop. The kinematic model is also implemented inside the Simulink model. It consists of the forward kinematics, the inverse kinematics and the differential kinematics. The differential kinematic model provides the Jacobian matrices. The control architecture is implemented inside the simulation environment as shown in Figure 6.

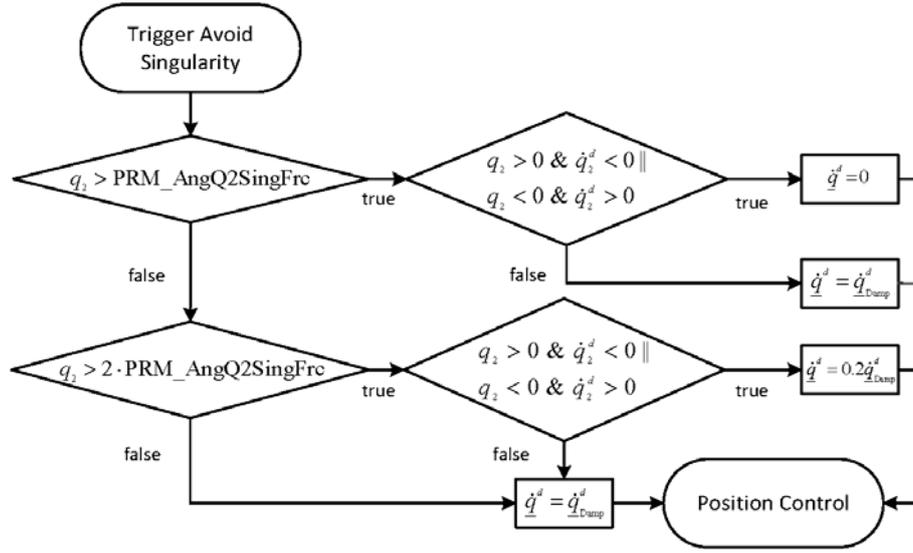


Figure 5 Scheme of avoiding singularities.

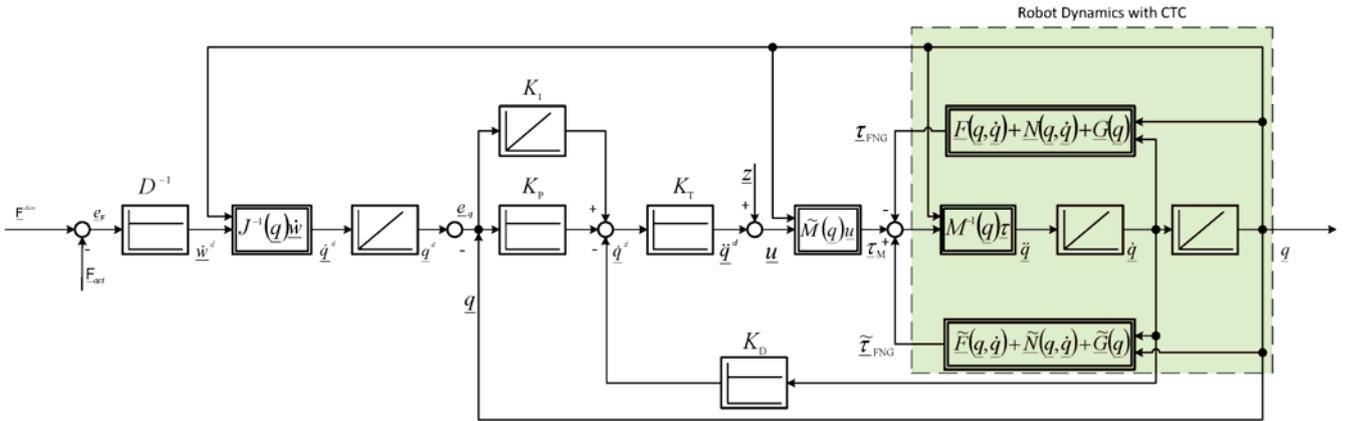


Figure 6 Control Architecture.

3.1 FORCE CONTROLLED GUIDANCE

For the simulation of the force controlled guidance the damping matrix is set to

$$D = \text{diag}(d_f \quad d_f \quad d_f \quad d_m \quad d_m) \quad (33)$$

with

$$d_f = 1000 \text{ Nsm}^{-1} \quad (34)$$

$$d_m = 100 \text{ Nsm} . \quad (35)$$

The desired force has to be

$$f_d = (0 \quad 0 \quad 0 \quad 0 \quad 0)^T . \quad (36)$$

Forces of 10N and -10N are applied one by one in x, y and z direction through step-functions inside the model of the force/torque-sensor. As one can see in Figure 7 the robot moves in the expected direction and shows no sudden

change of position. The robot also moves smoothly although there is additive noise in the simulated force sensor signal. The low pass filter for the force sensor signal which was implemented in the control scheme results in a delay between the moment of applying force and the start of the motion.

3.2 HYBRID MODE

For the simulation of the hybrid mode the work plane of the hybrid workplace is integrated into simulation. Here, the contact force between the robot and the work surface is recognized via the force/torque-sensor. The work surface is modelled via a spring and damper system

$$f_z = \begin{cases} 0 & \text{for } q_5 > h \\ -c(q_5 - h) - d\dot{q}_5 & \text{for } q_5 \leq h \end{cases} \quad (37)$$

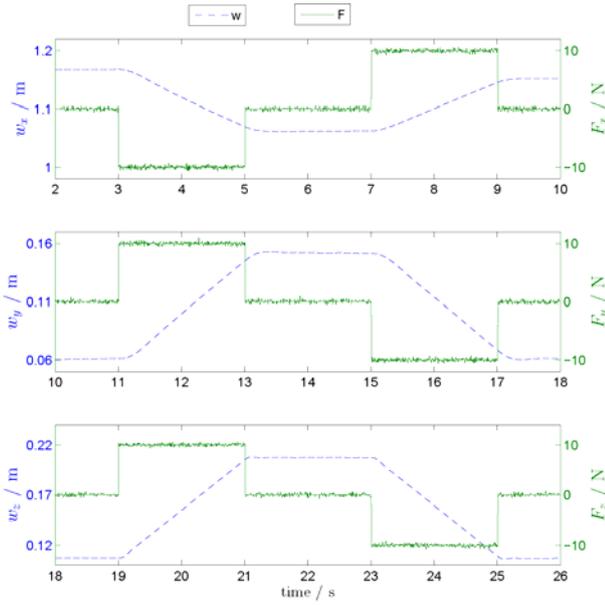


Figure 7 Simulation results for force controlled guidance.

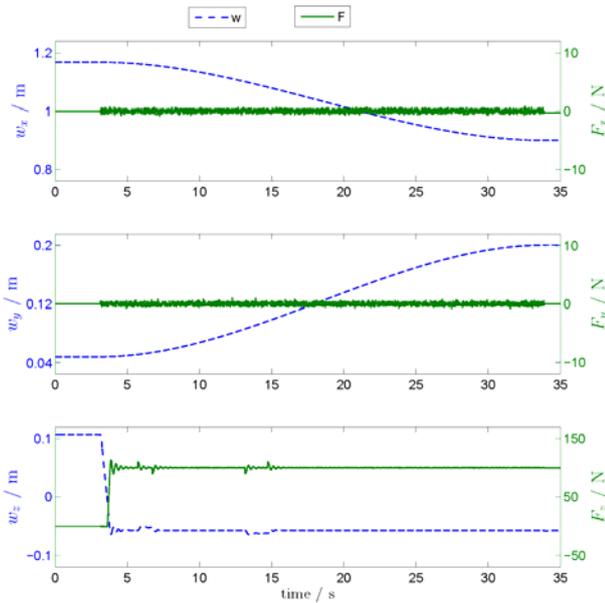


Figure 8 Simulation results for hybrid mode.

with stiffness $c=2000\text{ Nm}^{-1}$, damping coefficient $d=5\text{ Nsm}^{-1}$ and the height $h=-0.3\text{ m}$. Contact is given for $q_5 \leq h$. The desired values for the path-planning algorithm are

$$\begin{pmatrix} w_x \\ w_y \end{pmatrix} = \begin{pmatrix} 0.9\text{ m} \\ 0.2\text{ m} \end{pmatrix}. \quad (38)$$

After three seconds the path-planning starts and the desired force of

$$f_d = (0 \ 0 \ 100\text{ N} \ 0 \ 0)^T. \quad (39)$$

is applied. In hybrid mode there are also disturbances modelled as step functions which occur from $t=6\text{ s}$ to $t=7\text{ s}$ and from $t=13\text{ s}$ to $t=15\text{ s}$.

As one can see in Figure 8 the desired force is built up quickly and there is no remaining deviation. In fact, there is a significant overshoot which is caused on the one hand by the low stiffness of the modelled work surface and on the other hand by the long distance without contact force between the start position and the position of the work surface. Also it is apparent that the robot moves slightly until it reaches the desired values for w_x and w_y . There is no noise in the sensor signal for the first three seconds due to the fact that the software container in which the sensor is simulated is not triggered ab initio.

4 TESTS WITH THE CARO5X

As mentioned before, the Simulink model can also be used to generate code for the real time system through Matlabs Coder toolbox. Therefore there are only slightly changes - e.g. activation of safety systems, routing signals from the data acquisition system to the controller instead of data from the hardware simulation system - in the software model which was used for simulation necessary.

4.1 PARAMETERIZATION

Initially the control parameters are tuned during the experiment. Finally the parameters

$$\mathbf{K}_P = \text{diag}(120 \ 120 \ 300 \ 100 \ 500) \quad (40)$$

$$\mathbf{K}_I = \text{diag}(25 \ 25 \ 25 \ 2 \ 5) \quad (41)$$

$$\mathbf{K}_D = \text{diag}(20 \ 20 \ 20 \ 1.7 \ 20) \quad (42)$$

were used. The values of axis four and five are notably different from the other. That is caused by the computed torque model. The computed torque control is implemented for axis one, two and three due to the fact that they have the significant inertias and the significant effects on gyroscopic forces. Also the damping matrix is adapted during the experiment and were finally parameterized to

$$\mathbf{D} = \text{diag}(25 \ 25 \ 25 \ 2 \ 5). \quad (43)$$

With these parameters a fast and intuitive guidance by force is possible.

4.2 FORCE CONTROLLED GUIDANCE

To gather experimental data the operator moves the robot by applying forces as shown in Figure 10. During the motion the measured workspace coordinates, the desired velocities in workspace coordinates and the applied forces and torques are tracked. Figure 9 shows the translational workspace coordinates $(w_x \ w_y \ w_z)^T$ and the measured forces $(f_x \ f_y \ f_z)^T$, Figure 11 the rotational workspace coordinates $(\alpha \ \beta)^T$ and the applied torques $(M_\alpha \ M_\beta)^T$. It is obvious that the robot moves in the direction in which

a force is applied. Also there are no harsh motions present although the force signal changes rapidly in some cases.

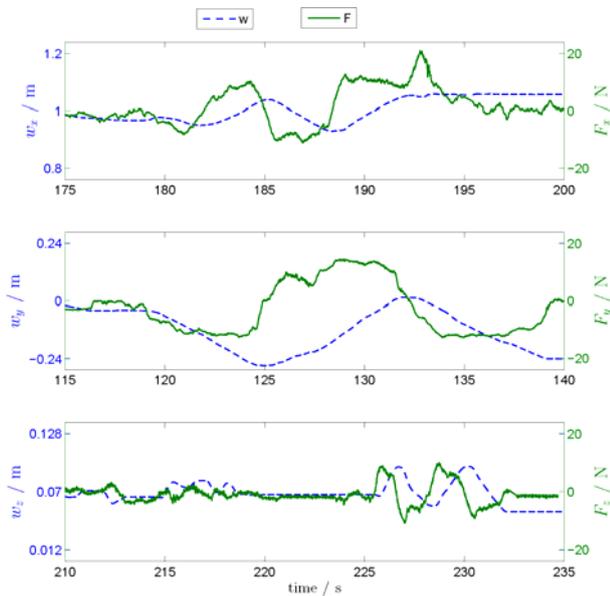


Figure 9 Translational workspace coordinates and applied forces.

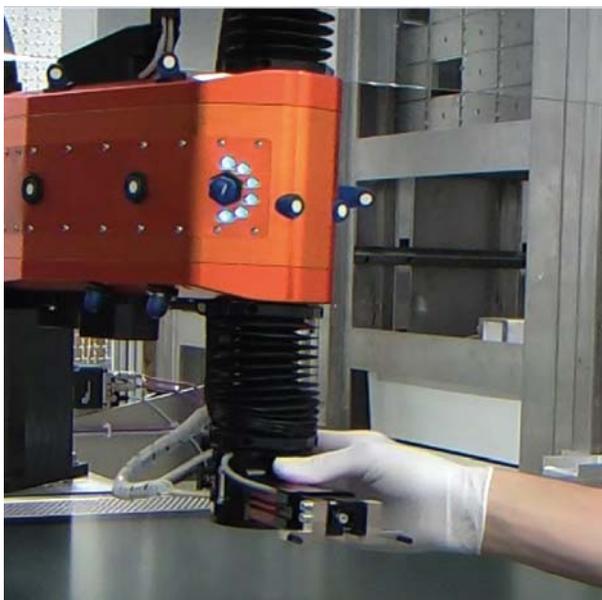


Figure 10 CARo5X in collaborative mode.

The delay between the moment of applying forces and the start of the motion is short enough to not being recognized by the operator. The actual velocity in workspace coordinates can be seen in Figure 12 and Figure 13. The left and right y-axis is scaled with the damping matrix. The actual velocity has a lot of noise due to the here used incremental encoders. However, it is nevertheless identifiably that the actual velocities follow the applied forces. In the z-axis there occur motions although there are nearly no forces applied.

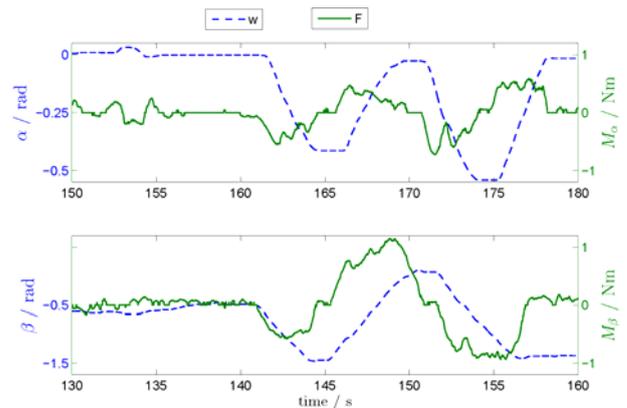


Figure 11 Rotational workspace coordinates and applied forces.

This is based on the effect that the z-coordinate is also influenced by axis four if axis three is not zero. Also one can see that the rotational workspace coordinates are extremely noisy. Here the profit of the used control technique is distinct.

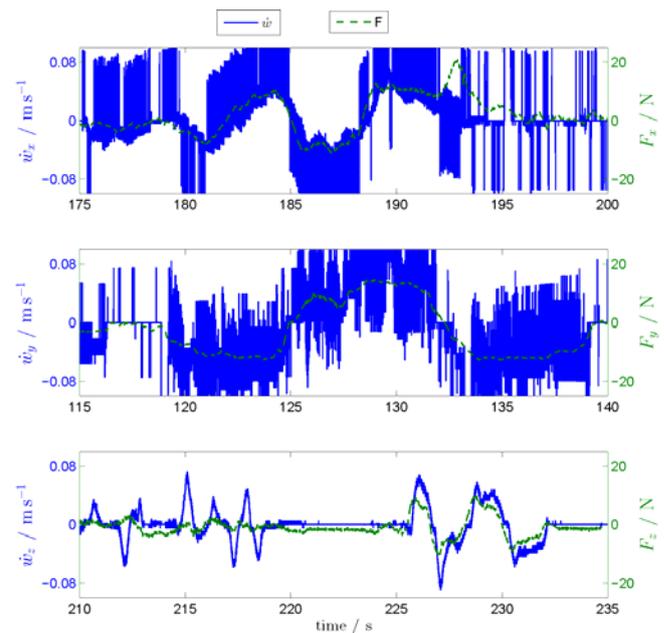


Figure 12 Actual translational workspace coordinate velocity and applied forces.

4.3 HYBRID MODE

The experiment is done by setting a desired value for f_z and desired values for the path planning. The robot then moves along the path and has contact with the hand of the operator. The operator also moves his hand away from the robot during the experiment to see whether the robot follows the hand. Figure 13 shows the value of joint five and the measured force in z-direction. At the beginning the operator moves his hand and the robot follows until the desired force of $f_z = 10\text{N}$ is reached. At the end the operator

holds his hand and the robot applies the desired force against the operators' hand.

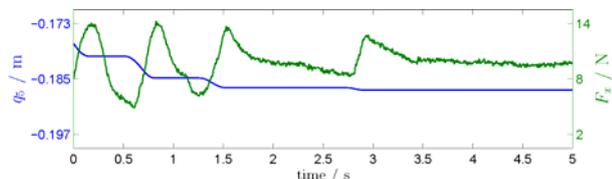


Figure 13 Joint angle q_5 and applied force in z-direction.

5 CONCLUSIONS AND OUTLOOK

An architecture for the force controlled guidance and path/force controlled hybrid mode has been developed and experimentally validated. The architecture has been tested through simulation and an experiment. In simulation the behaviour of the robot to applied forces in force controlled guidance mode and to desired forces in hybrid mode have been evaluated. The experimental results have shown that the proposed algorithm works adequate although there is a lot of noise in both the force sensor signal and the velocity signal from the incremental encoders. Improvements can be done in the dynamic model by including axis four and five in the dynamic model. Also the noise in the velocity signal can be reduced by replacing the incremental encoders with absolute multiturn encoders or by inserting an observer based velocity estimation. The PID-control architecture will be compared to PPI-control architecture to possibly increase the performance of the control loop.

ACKNOWLEDGMENTS

The authors would like to thank the European Commission which funded the ManuCyte Project and made it possible to develop the robot and its control architecture.

REFERENCES

- [1] Hoecherl J. and Schlegl T., An Image Based Algorithm to Safely Locate Human Extremities for Human-Robot Collaboration, *Lecture Notes in Computer Sciences*, Vol. 7508: *Intelligent Robotics and Applications*, Su C.-Y., Rakheja S. and Liu H., pp. 164-175, Springer, 2012.
- [2] Arbesmeier M., Goß A., Schlegl T., Entwicklung eines Assistenzroboters für eine Anlage zur Kultivierung menschlicher Zellen in industriellem Maßstab, *Tagungsband zur 8. Fachkonferenz: Angewandte Automatisierungstechnik in Lehre und Entwicklung*, Göttingen/Germany, Vol. 8, pp. 347-349, 2011.
- [3] Murphy R. R., Human-Robot Interaction in Rescue Robotics, *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 34, No. 2, pp. 138-143, May 2004.
- [4] Cappelli A. and Giovannetti E., Human-Robot Interaction, *Proceedings of the first RoboCare Workshop*, Rome/Italy, pp. 15-20, October 2003.
- [5] Heiligensetzer P., Aktuelle Entwicklungen bei Industrierobotern im Bereich der Mensch-Roboter Kooperation, *Tag der Arbeitssicherheit in Fellbach*, Fellbach, 2009.
- [6] Kang H. S. and Park J. O., Robotic Polishing with Direct Teaching Method Using Force/Torque Sensor, *Proceedings of the 1st International Conference on Engineering Design and Automation*, Bangkok/Thailand, Vol. 1, March 1997.
- [7] Anderson R. and Spong M., Hybrid Impedance Control of Robotic Manipulators, *IEEE Journal of Robotics and Automation*, Vol. 4, No. 5, pp. 549-556, October 1988.
- [8] Albu-Schäffer A., Ott A. and Hirzinger G., A Unified Passivity Based Control Framework for Position, Torque and Impedance Control of Flexible Joint Robots, *The International Journal of Robotics Research*, Vol. 26, No 1, pp. 23-39, January 2007.
- [9] Becke M. and Schlegl T., Derivation of Dynamic Equations of Serial Robot Manipulators with Coupled Ideal Joint Motion, *Lecture Notes in Computer Sciences*, Vol. 7508: *Intelligent Robotics and Applications*, Su C.-Y., Rakheja S. and Liu H., pp. 236-247, Springer, 2012.
- [10] Salinas S., Larrarte E. and Alban A., Operational Space Control for a Puma Robot, *Proceedings of the International Conference on Mechatronics and Automation*, Harbin/China, pp. 2183-2188, August 2007.

EXTENDING CONTINUOUS PATH TRAJECTORIES TO POINT-TO-POINT TRAJECTORIES BY VARYING INTERMEDIATE POINTS

Hubert Gatringer* Matthias Oberherber* Klemens Springer*

* Johannes Kepler University Linz

ABSTRACT

This paper shows a method for the calculation of fast point-to-point movements for industrial robots subject to various physical constraints. The main idea of the proposed algorithm is to solve first the problem of time optimisation of continuous path trajectories. A predefined path in world coordinates as well as the dynamical equations of motion are projected onto a one dimensional path parameter, the time behaviour of which is optimised with the dynamic programming approach (Bellman optimisation). The optimal path for the point-to-point problem is found by varying additional intermediate points between start- and endpoint of the predefined path in an optimal manner using a nonlinear solver. Each path with varied intermediate points is optimised with the dynamic programming approach. Simulation results as well as experimental results for an industrial robot are shown.

Keywords: Point-to-Point Motion, Optimisation, Industrial Robot, Robots Kinematic and Dynamics

1 INTRODUCTION

Fast robot movements in highly automated production lines have become more and more important in the last few years to sustain production locations in high wage countries. This can be achieved by path optimisation. In a first step, the optimisation of a continuous path (CP) trajectory (predefined path) subject to mechanical constraints is performed. The problem is divided in the geometric path planning and its optimisation using a scalar path parameter. The works of [12], [2] and [14] provide the basis for effective algorithms. [4], [8] and [10] extended this basic algorithms by additionally taking end-effector friction constraints into account. The aim is to transport an object that is loosely put onto the end-effector in shortest possible time on a predefined path (waiter-motion-problem). [8] use quintic B-splines to define the geometric path. The time optimal solution for the path parameter is obtained by finding optimal switching points for acceleration and deceleration in the phase plane (forward and backward integration).

In contrast to that, we use Bernstein polynomials for the definition of the geometric path via splines. Motor torque constraints can be effectively included in the phase plane. The limiting curves are evaluated with a reformulation of the Projection Equation, see [3] and [6]. Moreover we calculate the time optimal solution using the dynamic programming approach (Bellman optimisation) to provide faster solutions and to be able to include additional terms in the cost functional, see also [11]. This offers the possibility to calculate time/energy optimal solutions. A recursive log-barrier method for time-optimal robot path tracking is presented in [15]. As an extension to the CP trajectories, the point-to-point (PtP) optimisation is not limited to geometrically defined paths. It can, for instance, be approached by formulating the task as two-point boundary value problem and solved with numerical software packages, see e.g. [7]. However, in this work, a solution is computed by varying the position and orientation of intermediate points of the CP trajectory using a numerical solver. The paper is organized as follows: In section 2 the optimisation of the CP trajectories is performed. Constraints like motor torques, velocities and accelerations are included. This is the basis for the optimisation of the PtP trajectories in section 3. Experimental results and comparisons to other optimisation approaches are part of section 4. The experiment is realized with a Stäubli RX130L six axes industrial robot.

Contact author: Hubert Gatringer¹

¹Institute for Robotics, Johannes Kepler University Linz,
Altenbergerstr. 69, 4040 Linz, Austria
Email: hubert.gatringer@jku.at

2 CONTINUOUS PATH TRAJECTORIES

2.1 GEOMETRIC PATH PLANNING

There are several methods to define a geometric path, e.g. straight lines, curves, clothoids or splines. The latter ones are used in this work to define the path for the end-effector

$$\mathbf{z}_E = (\mathbf{r}_E^T \ \boldsymbol{\varphi}_E^T)^T, \quad (1)$$

see Fig. 1.

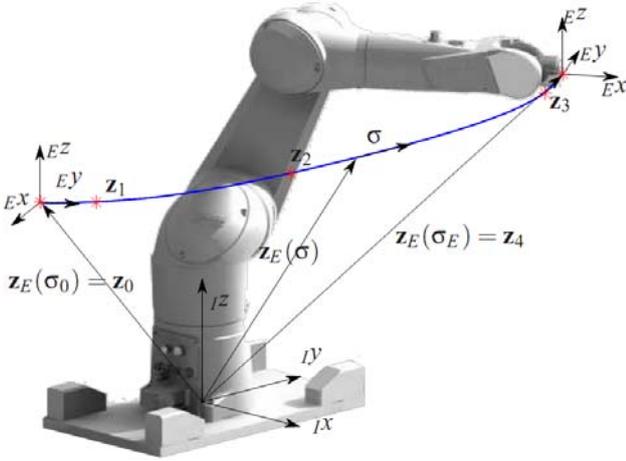


Figure 1 Path planning by interpolation of points.

Vector \mathbf{r}_E is the position of the end-effector while $\boldsymbol{\varphi}_E$ is the orientation in terms of Cardan angles, respectively. The orientation matrix $\mathbf{A}_{EI} = \mathbf{A}_\gamma \mathbf{A}_\beta \mathbf{A}_\alpha$ for this transformation is defined by subsequent standard rotations about the x - (angle α), y - (angle β) and z - (angle γ) direction. A set of j pre-defined points $\mathbf{z}_{E,j}$ is interpolated to the whole path that can be parameterised by a (scalar) path parameter σ that is determined in the range $\sigma_0 = 0$ (start point) and $\sigma_E = 1$ (end point). The spline within this range is defined by Bernstein polynomials

$$R_k^d(\sigma) = \binom{d}{k} \frac{(\sigma - \sigma_0)^k (\sigma_E - \sigma)^{d-k}}{(\sigma_E - \sigma_0)^d} \text{ for } k = 0, \dots, d \quad (2)$$

resulting from a specific choice of number and location of the B-spline nodes, see [13] for details. This kind of global basis functions provides the possibility to calculate derivatives with respect to the path parameter analytically in a simple way. Furthermore the non-recursive definition has a positive effect on the computation time. Parameter d in Eq. (2) is the degree of the Bernstein polynomial. Figure 2 shows them for degree $d=4$, that is chosen in the present work.

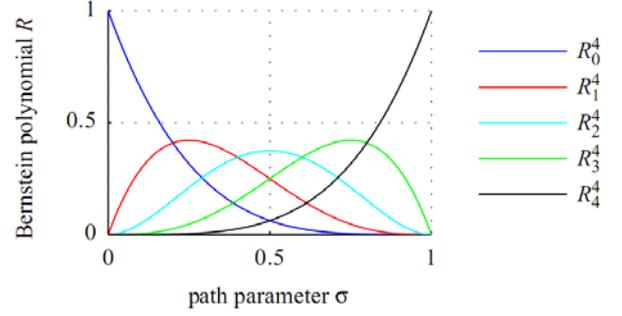


Figure 2 Bernstein polynomial of degree $d=4$.

To define a path through interpolation points $\mathbf{Z} = (\mathbf{z}_0, \mathbf{z}_1 \dots \mathbf{z}_n)$, a chordal discretisation for σ is performed. This means, that $n+1$ points are distributed over the range of the path parameter $\sigma_0 \dots \sigma_E$ with respect to the distance of the defined points in space. With the length of the polynomial-line

$$L = \sum_{j=0}^{n-1} \|\mathbf{z}_{E,j+1} - \mathbf{z}_{E,j}\|, \quad (3)$$

the discretisation points are defined by

$$\sigma_k = \sigma_0 + \frac{\sigma_E - \sigma_0}{L} \sum_{j=0}^{k-1} \|\mathbf{z}_{E,j+1} - \mathbf{z}_{E,j}\|. \quad (4)$$

With this chordal parameterisation for σ , the end-effector path can be written as

$$\mathbf{z}_E(\sigma) = \sum_{j=0}^n \mathbf{d}_j R_j^d(\sigma). \quad (5)$$

Note, the more interpolation points are chosen, the more basis functions, see Fig. 2, enter calculations. The control points

$$\mathbf{D} = (\mathbf{d}_0, \mathbf{d}_1 \dots \mathbf{d}_n) \quad (6)$$

are the result of the linear equation

$$\mathbf{R}_B \mathbf{D} = \mathbf{Z}, \quad (7)$$

where \mathbf{R}_B is the matrix of Bernstein polynomials

$$\mathbf{R}_B = \begin{bmatrix} R_0^d(\sigma_0) & R_1^d(\sigma_0) & \dots & R_n^d(\sigma_0) \\ R_0^d(\sigma_1) & R_1^d(\sigma_1) & \dots & R_n^d(\sigma_1) \\ \vdots & \vdots & \ddots & \vdots \\ R_0^d(\sigma_n) & R_1^d(\sigma_n) & \dots & R_n^d(\sigma_n) \end{bmatrix} \quad (8)$$

and $\mathbf{Z} = (\mathbf{z}_0, \mathbf{z}_1 \dots \mathbf{z}_n)$ are the pre-defined points in space.

With the geometric definition at hand, the time behaviour of $\sigma(t)$ can be optimised to calculate the trajectory. More details on spline parameterisations can be found in [5].

2.2 PHYSICAL CONSTRAINTS

For trajectory planning, several physical constraints have to be considered, namely motor torque-, velocity- and acceleration constraints

$$\begin{aligned} \dot{\mathbf{q}}_{min} &\leq \dot{\mathbf{q}} \leq \dot{\mathbf{q}}_{max} \\ \ddot{\mathbf{q}}_{min} &\leq \ddot{\mathbf{q}} \leq \ddot{\mathbf{q}}_{max} \\ \mathbf{Q}_{min} &\leq \mathbf{Q} \leq \mathbf{Q}_{max} \end{aligned} \quad (9)$$

Since the trajectory is defined for the end-effector in world coordinates, a transformation into joint co-ordinates has to be performed by inverse kinematic calculations

$$\mathbf{q} = \text{invkin}(\mathbf{z}_E(\sigma)) = \mathbf{q}(\sigma). \quad (10)$$

A projection of velocities and accelerations onto the path coordinate yields

$$\dot{\mathbf{q}} = \frac{d\mathbf{q}}{d\sigma} \dot{\sigma} = \mathbf{q}' \dot{\sigma} \quad (11)$$

$$\ddot{\mathbf{q}} = \frac{d}{d\sigma}(\mathbf{q}' \dot{\sigma}) \dot{\sigma} = \mathbf{q}'' \dot{\sigma}^2 + \frac{1}{2} \mathbf{q}' (\dot{\sigma}^2)', \quad (12)$$

when using the identity $\ddot{\sigma} = (d\dot{\sigma}/d\sigma)(d\sigma/dt) = (\dot{\sigma}^2)'/2$. However, the evaluation of the derivations with respect to σ are difficult to realize even for the case, when the inverse kinematics Eq. (10) is solved numerically. Fortunately, the identity

$$\mathbf{z}'_E = \frac{d\mathbf{z}_E}{d\mathbf{q}} \mathbf{q}' \quad (13)$$

$$\mathbf{z}''_E = \frac{d\mathbf{z}_E}{d\mathbf{q}} \mathbf{q}'' + \frac{d\mathbf{z}'_E}{d\mathbf{q}} \mathbf{q}', \quad (14)$$

and therefore

$$\mathbf{q}' = \left(\frac{d\mathbf{z}_E}{d\mathbf{q}} \right)^+ \mathbf{z}'_E \quad (15)$$

$$\mathbf{q}'' = \left(\frac{d\mathbf{z}_E}{d\mathbf{q}} \right)^+ \left[\mathbf{z}''_E - \frac{d\mathbf{z}'_E}{d\mathbf{q}} \mathbf{q}' \right] \quad (16)$$

holds, which is much less computational complex. The generalized matrix inverse reads

$$\left(\frac{d\mathbf{z}_E}{d\mathbf{q}} \right)^+ = \left[\left(\frac{d\mathbf{z}_E}{d\mathbf{q}} \right)^T \left(\frac{d\mathbf{z}_E}{d\mathbf{q}} \right) \right]^{-1} \left(\frac{d\mathbf{z}_E}{d\mathbf{q}} \right)^T. \quad (17)$$

2.2.1 Torque Constraints

The basis for taking motor torque constraints into account is an efficient dynamical modelling. We use the Projection Equation (see [3])

$$\sum_{i=1}^N \left(\begin{pmatrix} \frac{\partial_R \mathbf{v}_c}{\partial \dot{\mathbf{q}}} \\ \frac{\partial_R \boldsymbol{\omega}_c}{\partial \dot{\mathbf{q}}} \end{pmatrix}_i \right)^T \left(\begin{matrix} {}_R \dot{\mathbf{p}} + {}_R \tilde{\boldsymbol{\omega}}_{IR} {}_R \mathbf{p} - {}_R \mathbf{f}^e \\ {}_R \dot{\mathbf{L}} + {}_R \tilde{\boldsymbol{\omega}}_{IR} {}_R \mathbf{L} - {}_R \mathbf{M}^e \end{matrix} \right)_i = (18)$$

for this task. Linear momenta $\mathbf{p} = m \mathbf{v}_c$ and angular momenta $\mathbf{L} = \mathbf{J}^c \boldsymbol{\omega}_c$ are projected into the minimal space (minimal velocities $\dot{\mathbf{q}}$) via the appropriate Jacobian matrices. All the values like the translational velocity \mathbf{v}_c or the rotational velocity $\boldsymbol{\omega}_c$ of the centre of gravity can be inserted in arbitrary coordinate systems R . In contrast to $\boldsymbol{\omega}_c$, $\boldsymbol{\omega}_{IR}$ is the velocity of the used reference system. Since we use body fixed reference systems, m, \mathbf{J}^c holds in this case. Parameters are the mass and the inertia tensor, respectively, while $\tilde{\boldsymbol{\omega}} \mathbf{p}$ characterizes the vector product $\boldsymbol{\omega} \times \mathbf{p}$. \mathbf{f}_i^e and \mathbf{M}_i^e are imposed forces and moments acting on the i th body. Evaluating the Projection Equation for $N = 12$ bodies (the industrial robot consists of six motors and six links) leads to the equations of motion

$$\mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{g}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{Q}, \quad (19)$$

where $\mathbf{M}(\mathbf{q})$ is the position dependent, positive definite, symmetric mass matrix, \mathbf{g} contains all nonlinear terms like gravitational, Coriolis, centrifugal and friction forces, while \mathbf{Q} are the motor torques. Of course, the dynamical behaviour of the robot can be simulated with Eq. (19). However in our case, it is needed to take motor torque constraints into account. Therefore, Eq. (18) can be rewritten to

$$\sum_{i=1}^N \mathbf{F}_i^T \left\{ \underbrace{\begin{bmatrix} m\mathbf{E} & 0 \\ 0 & \mathbf{J}^c \end{bmatrix}}_{\mathbf{M}} \begin{pmatrix} \dot{\mathbf{v}}_c \\ \dot{\boldsymbol{\omega}}_c \end{pmatrix} + \underbrace{\begin{bmatrix} \tilde{\boldsymbol{\omega}}_c m & 0 \\ 0 & \tilde{\boldsymbol{\omega}}_c \mathbf{J}^c \end{bmatrix}}_{\mathbf{G}} \begin{pmatrix} \mathbf{v}_c \\ \boldsymbol{\omega}_c \end{pmatrix} \right\}_i = \mathbf{Q}^c \quad (20)$$

by substituting the abbreviations of the momenta and \mathbf{E} for the identity matrix. The velocities and accelerations depending on the path parameter are

$$\begin{pmatrix} \mathbf{v}_c \\ \boldsymbol{\omega}_c \end{pmatrix}_i = \underbrace{\begin{bmatrix} \frac{\partial \mathbf{v}_c}{\partial \dot{\mathbf{q}}} \\ \frac{\partial \boldsymbol{\omega}_c}{\partial \dot{\mathbf{q}}} \end{bmatrix}}_{\mathbf{F}_i} \dot{\mathbf{q}} = (\mathbf{F}_i \mathbf{q}') \dot{\sigma}, \quad (21)$$

$$\begin{pmatrix} \dot{\mathbf{v}}_c \\ \dot{\boldsymbol{\omega}}_c \end{pmatrix}_i = \frac{1}{2} (\mathbf{F}_i \mathbf{q}') (\dot{\sigma}^2)' + (\mathbf{F}_i \mathbf{q}')' (\dot{\sigma}^2). \quad (22)$$

The expression $\overline{\mathbf{G}}_i (\mathbf{v}_c^T \ \boldsymbol{\omega}_c^T)_i^T$ can further be evaluated as

$$\overline{\mathbf{G}} \begin{pmatrix} \mathbf{v}_c \\ \boldsymbol{\omega}_c \end{pmatrix} = \underbrace{\begin{bmatrix} \left[\frac{\partial \boldsymbol{\omega}_c}{\partial \dot{\mathbf{q}}} \mathbf{q}' \right]^{\sim} m & 0 \\ 0 & \left[\frac{\partial \boldsymbol{\omega}_c}{\partial \dot{\mathbf{q}}} \mathbf{q}' \right]^{\sim} \mathbf{J}^c \end{bmatrix}}_{\mathbf{G}^*} \mathbf{F} \mathbf{q}' \dot{\sigma}^2, \quad (23)$$

(index i suppressed) and Eq. (20) equals

$$\sum_{i=1}^N \left\{ \mathbf{F}^T \left[\overline{\mathbf{M}} \left(\frac{1}{2} \mathbf{F} \mathbf{q}' (\dot{\sigma}^2)' + (\mathbf{F}' \mathbf{q}')' (\dot{\sigma}^2) \right) + \mathbf{G}^* \mathbf{F} \mathbf{q}' (\dot{\sigma}^2) - \begin{pmatrix} \mathbf{f}^e \\ \mathbf{M}_{frict}^e \end{pmatrix} \right] \right\}_i = \sum_{i=1}^N \mathbf{F}_i^T \begin{pmatrix} 0 \\ \mathbf{M}_{mot}^e \end{pmatrix}. \quad (24)$$

The impressed torques can be divided into the friction torques $\mathbf{M}_{frict}^e = -\mathbf{r}_c \text{sign}(\mathbf{q}'(\sigma)) - \mathbf{r}_v \mathbf{q}'(\sigma) \dot{\sigma}$ and the motor torques \mathbf{M}_{mot}^e that are written on the right side of Eq. (24). Collecting appropriate terms in Eq. (24) yields

$$\mathbf{A}(\sigma)(\dot{\sigma}^2)' + \mathbf{B}(\sigma)(\dot{\sigma}^2) + \mathbf{C}(\sigma) + \mathbf{D}(\sigma)\dot{\sigma} = \mathbf{Q} \quad (25)$$

with

$$\mathbf{A} = \sum_{i=1}^N \left(\frac{1}{2} \mathbf{F}^T \overline{\mathbf{M}} \mathbf{F} \mathbf{q}' \right)_i, \quad (26)$$

$$\mathbf{B} = \sum_{i=1}^N \left(\mathbf{F}^T \overline{\mathbf{M}} (\mathbf{F} \mathbf{q}')' + \mathbf{F}^T \mathbf{G}^* \mathbf{F} \mathbf{q}' \right)_i, \quad (27)$$

$$\mathbf{C} = -\sum_{i=1}^N \mathbf{F}_i^T \begin{pmatrix} \mathbf{f}^e \\ 0 \end{pmatrix}_i + \mathbf{r}_c \text{sign}(\mathbf{q}'(\sigma)), \quad (28)$$

$$\mathbf{D} = \mathbf{r}_v \mathbf{q}'(\sigma). \quad (29)$$

When using the abbreviation $z = \dot{\sigma}^2$, the k th motor torque ($M_{mot,k}$) equation yields

$$a_k z' + b_k z + c_k + d_k \sqrt{z} = M_{mot,k} \cdot i_k = u_k \quad (30)$$

(i_k : gear ratio of k th joint). The input u_k is constrained

between $u_{k,max}$ and $u_{k,min}$ and therefore Eq. (30) can be written as

$$-u_{k,min} \leq u_k \leq u_{k,max}. \quad (31)$$

This torque constraints offer a geometrical interpretation in the phase plane $[z', z]$. For each joint, we get an upper and lower limiting curve in this plane, shown in Fig. 3. Only the feasible region is valid for optimal path planning. Note, if viscous friction is neglected ($\mathbf{r}_v = 0$), Eq. (30) degenerates to straight limiting lines. However, this simplification should be avoided for real systems.

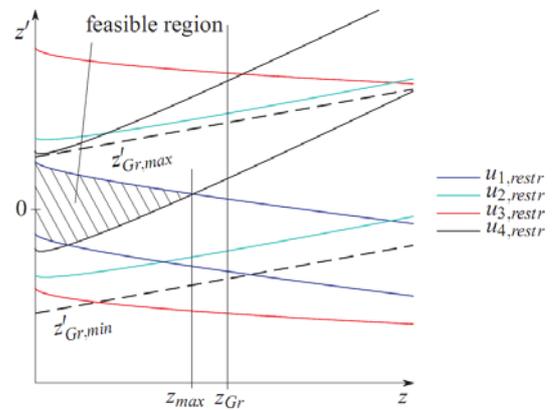


Figure 3 Limiting curves and feasible region.

2.2.2 Velocity/Acceleration Constraints

Beside torque constraints, velocity and acceleration constraints have to be considered. Joint angular as well as path velocities

$$-\dot{q}_{k,max} \leq \dot{q}'_k \dot{\sigma} \leq \dot{q}_{k,max} \quad (32)$$

$$-v_{max} \leq |\mathbf{r}'_E| \dot{\sigma} \leq v_{max} \quad (33)$$

can easily be transformed to (using $\dot{\sigma}^2 = z$)

$$z_{Gr} = \min \left(\frac{\dot{q}_{k,max}^2}{\dot{q}'_k{}^2}, \frac{v_{max}^2}{|\mathbf{r}'_E|^2} \right) \quad (34)$$

leading to vertical limiting curves on the position of z_{Gr} in the phase plane, see again Fig. 3. An evaluation of joint acceleration constraints

$$-\ddot{q}_{k,max} \leq q''_k z + \frac{1}{2} q'_k z' \leq \ddot{q}_{k,max} \quad (35)$$

delivers additional straight lines in the phase plane indicated by dashed lines in Fig. 3.

2.3 OPTIMISATION

The global goal of the CP optimisation is to find a minimum of the cost function

$$W = \int_0^{t_e} \left[k_1 + k_2 \sum_i (a_i z' + b_i z + c_i + d_i \sqrt{z})^2 \right] dt. \quad (36)$$

The second part in this equation corresponds to the minimization of the motor torques. Since we are only interested in time-optimal solutions, the weighting factors $k_1 = 1, k_2 = 0$ are used, and the equations of motion do not enter the cost functional. The unknown cycle time t_e is the sought solution of the optimisation, so a transformation of

$$\dot{\sigma} = \frac{d\sigma}{dt} \quad (37)$$

and therefore

$$dt = \frac{1}{\dot{\sigma}} d\sigma = \frac{1}{\sqrt{z}} d\sigma \quad (38)$$

leads to a cost function depending on z of

$$W = \int_0^{\sigma_E} \frac{1}{\sqrt{z}} d\sigma. \quad (39)$$

The calculation of the optimal trend of the velocity profile z is done with the help of the Bellman optimality principle [1]

”An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.”

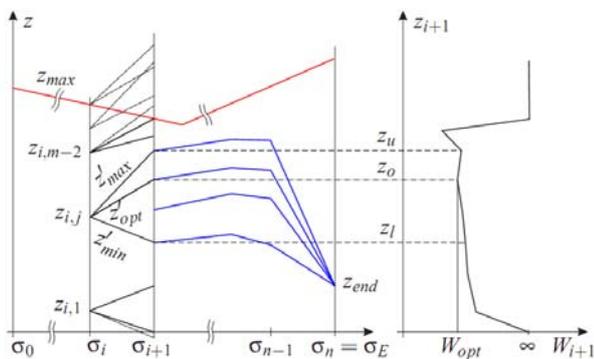


Figure 4 Bellman optimisation.

Therefore the calculation of the optimal trend z_{opt} starts at the last point σ_E by evaluating the maximum and minimum allowed values of z' (z'_{max}, z'_{min}) with the help of the

feasible region in Fig. 5. At first we discretise the velocity at each path point σ_i into $(m + 1)$ values $z_j(\sigma_i) = z_{i,j}$ with $j \in (0..m)$. The discretised cost function from Eq. (39) equals as a consequence of the Bellman optimality principle

$$W_{i,j} = \Delta\sigma \frac{1}{\sqrt{z_{i,j}}} + W_{i+1,j}, \quad (40)$$

where the discretisation step size of σ is $\Delta\sigma = \sigma_{i+1} - \sigma_i$.

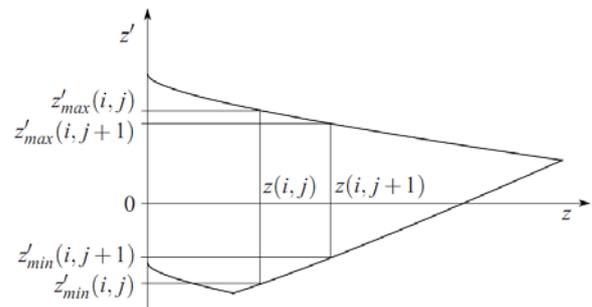


Figure 5 Feasible region with limits.

The optimal values of z' can now be evaluated by calculating the highest reachable point $z_u = z_{i,j} + z'_{max} \Delta\sigma$ and the lowest one $z_l = z_{i,j} + z'_{min} \Delta\sigma$, wherein the minimum of the cost function has to be sought, as shown in Fig. 4. This minimum search is done with the method of the golden ratio. With the found location of the minimum z_o , the optimal value of z' can be calculated with

$$z'_{opt}(i,j) = \frac{z_o - z_{i,j}}{\Delta\sigma}. \quad (41)$$

As $z'_{opt}(i,j)$ is calculated for each discretisation point (i,j) $i = 0..n, j = 0..m$, z_{opt} can be evaluated by solving

$$z_{i,opt} = z_{i-1,opt} + z'_{i-1,opt} \Delta\sigma \quad i = 1..n \quad (42)$$

with a forward iteration, starting at $z_0 = z_{start}$, see Fig. 6 for a graphical interpretation.

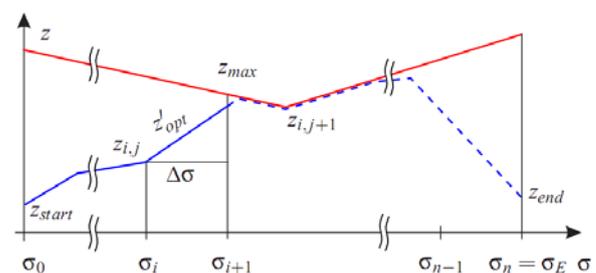


Figure 6 Computing the solution.

As soon as the optimal trend $z_{opt}(\sigma)$ is evaluated, the time behaviour of σ can be calculated by integrating Eq. (38)

$$t(\sigma) = \int_0^{\sigma} \frac{1}{\sqrt{z_{opt}}} d\sigma \quad (43)$$

and calculating the inverse function to get $\sigma(t)$. Therefore the overall trajectory as well as the motor torques Eq. (25), that can be used as model-based feed-forward torques, are defined.

3 POINT-TO-POINT TRAJECTORIES

The strategy to get optimal PtP trajectories is to use the results from the previous section and release intermediate points as optimisation variables. However, the convergence (time) to the optimal solution can drastically be increased if not all 6 coordinates of the intermediate points are used as optimisation variables, see also [8], where a similar approach is used to constrain maximal accelerations of a roller coaster system. Also lower (lb) and upper (ub) bounds for the optimisation variables decrease convergence time. For our time optimal PtP problem, positions should only vary in E^X (value n -normal) and E^Z (value b -binormal) direction of the end-effector coordinate system, while the orientations should only vary for the Cardan angles α (rotation around E^X) and β (rotation around E^Y), see Fig. 7. This restriction increases the convergence and does not influence the result significantly.

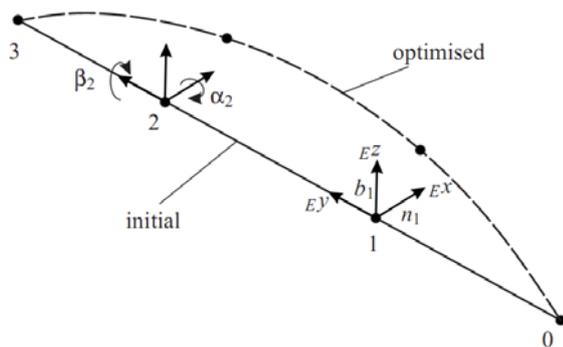


Figure 7 Optimisation variables.

The whole set of optimisation variables is

$$\xi = [n_1, b_1, \dots, n_l, b_l, \alpha_1, \beta_1, \dots, \alpha_k, \beta_k], \quad (44)$$

where l and k are the number of intermediate points for position and orientation, respectively. Geometrical constraints like barriers in the workspace can be defined as additional nonlinear constraints. For instance, the distance between the end-effector and a disturbing object $d = \|\mathbf{r}_E - \mathbf{r}_O\|$ should be greater than a safety distance

d_0 . Summing up the above mentioned assumptions, the optimisation problem reads as

$$\text{Minimize}_{\xi} \int_0^{\sigma_E} \frac{1}{\sqrt{z}} d\sigma, \quad (45)$$

subject to

$$lb \leq \xi \leq ub \quad (46)$$

$$0 \leq d - d_0. \quad (47)$$

This can be solved e.g. with the *BOBYQA* algorithm of the NLOPT optimisation package, see [16] and [17] for details. Note, in the cost functional Eq. (45), the time optimal CP trajectories have to be computed.

The overall optimisation strategy for the PtP trajectories is summarised in the flowchart Fig. 8.

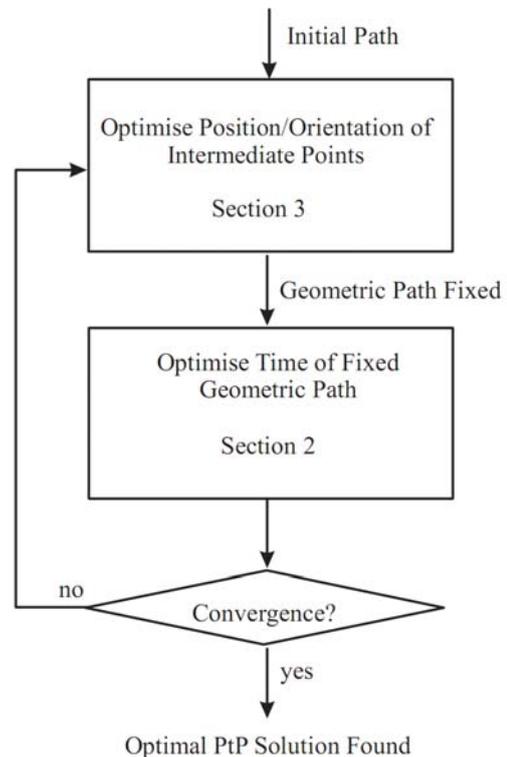


Figure 8 Flowchart of PtP optimisation approach.

4 RESULTS

4.1 PRESENT APPROACH

The proposed method is applied for a specific PtP trajectory planning problem. The test case is chosen in such a way, such that the initial trajectory passes near the wrist singularity of the robot. For a PtP solution this is not time-optimal due to the velocity constraint being active near the singular position. It is shown that the PtP time-optimal trajectory diverges from this singularity in an optimal

manner such that process time decreases. So the start- and endpoint are defined as

$$\mathbf{r}_{E,0} = \begin{pmatrix} 1.36 \\ -0.5 \\ 0.23 \end{pmatrix}, \quad \boldsymbol{\varphi}_{E,0} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad (48)$$

$$\mathbf{r}_{E,E} = \begin{pmatrix} 0.36 \\ 0.6 \\ 0.73 \end{pmatrix}, \quad \boldsymbol{\varphi}_{E,E} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad (49)$$

see Fig. 9. For the optimisation, two equidistant

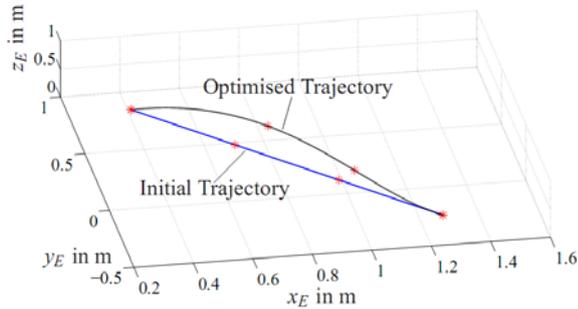


Figure 9 Initial and optimised trajectory in Cartesian coordinates.

intermediate points $\mathbf{r}_{E,1}, \mathbf{r}_{E,2}$ between start and endpoint, that are located on a straight line, are chosen, yielding

$$\mathbf{r}_{E,1} = \begin{pmatrix} 1 \\ -0.13 \\ 0.4 \end{pmatrix}, \quad \mathbf{r}_{E,2} = \begin{pmatrix} 0.7 \\ 0.2 \\ 0.6 \end{pmatrix}. \quad (50)$$

A similar procedure is chosen for the orientation description. For this case three intermediate Cardan angles are chosen. Applying the optimisation algorithm from section 3 yields the optimal intermediate points. The upper and lower bounds for the variations of positions and orientations are given in Tab. I.

Table I - Upper and lower bounds for optimisation variables

	n_1, n_2	b_1, b_2	$\alpha_1, \beta_1, \alpha_3, \beta_3$	α_2, β_2
<i>lb</i>	-0.2m	-0.5m	$-\pi/10$ rad	$-\pi/5$ rad
<i>ub</i>	0.4m	0.2m	$\pi/10$ rad	$\pi/5$ rad

The optimised result for the positions is

$$\mathbf{r}_{1,opt} = \begin{pmatrix} 1.1 \\ 0.0 \\ 0.2 \end{pmatrix}, \quad \mathbf{r}_{2,opt} = \begin{pmatrix} 0.8 \\ 0.44 \\ 0.35 \end{pmatrix} \quad (51)$$

while the optimal intermediate Cardan angles are

$$\boldsymbol{\varphi}_{1,opt} = \begin{pmatrix} 0 \\ -0.19 \\ 0 \end{pmatrix}, \quad \boldsymbol{\varphi}_{2,opt} = \begin{pmatrix} 0 \\ -0.14 \\ 0 \end{pmatrix} \quad (52)$$

$$\boldsymbol{\varphi}_{3,opt} = \begin{pmatrix} 0 \\ -0.22 \\ 0 \end{pmatrix}.$$

These points are interpolated by the spline curves, described in section 2. Figure 9 presents the optimised trajectory in world coordinates, while Fig. 10 shows the time behaviour of the Cardan angles α and β of the end-effector.

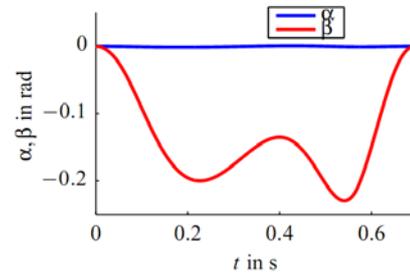


Figure 10 Optimised Cardan angles for end-effector.

For the particular chosen start- and endpoint, the optimised angle α is nearly zero. The resulting joint angles, normalized joint velocities and normalized motor torques are presented in Figs. 11, 12 and 13.

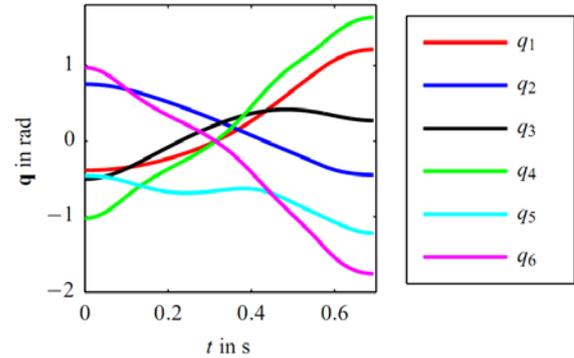


Figure 11 Joint angles: present approach.

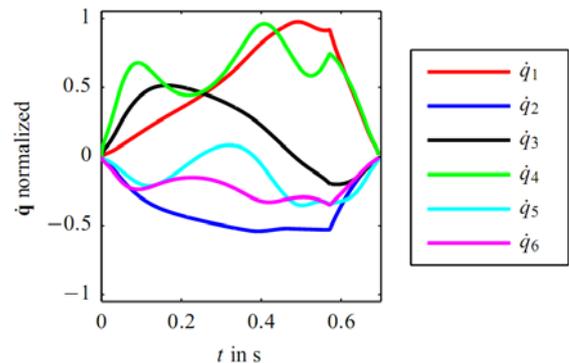


Figure 12 Normalised joint velocities: present approach.

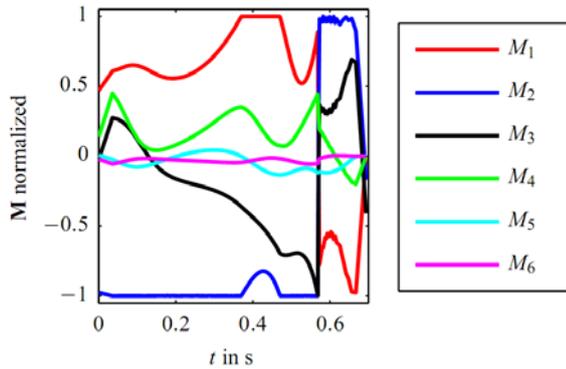


Figure 13 Normalised torques: present approach.

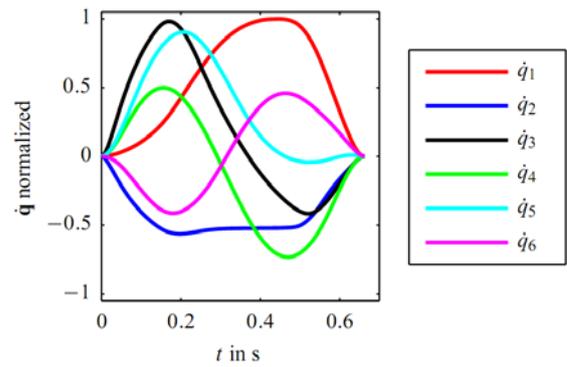


Figure 15 Normalised joint velocities: MUSCOD-II solution.

4.2 MUSCOD-II SOLUTION

An alternative solution for the same PtP problem is solved with the numerical solver MUSCOD-II, see [9] for details. The two point boundary value problem

$$\text{Minimize } \int_0^{t_e} (k_1 + k_2 \mathbf{Q}^T \mathbf{Q}) dt \quad (53)$$

subject to

$$|\dot{\mathbf{q}}| \leq \dot{\mathbf{q}}_{max} \quad (54)$$

$$|\mathbf{Q}| \leq \mathbf{Q}_{max} \quad (55)$$

$$\frac{d}{dt} \begin{pmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{pmatrix} = \begin{pmatrix} \dot{\mathbf{q}} \\ \mathbf{M}^{-1}(\mathbf{q})(\mathbf{Q} - \mathbf{g}(\mathbf{q}, \dot{\mathbf{q}})) \end{pmatrix} \quad (56)$$

$$\mathbf{q}(0) = \mathbf{q}_0 \quad (57)$$

$$\dot{\mathbf{q}}(0) = \dot{\mathbf{q}}_0 \quad (58)$$

$$\mathbf{q}(t_e) = \mathbf{q}_e \quad (59)$$

$$\dot{\mathbf{q}}(t_e) = \dot{\mathbf{q}}_e \quad (60)$$

is solved with the well known multiple shooting method. In contrast to the dynamic programming approach, presented in section 2.3 here additionally the motor torques are included to the cost functional, to achieve good convergence. However the weighting $k_1 = 1$, $k_2 = 1e-5$ is chosen to get (nearly) time optimal solutions. The results for the MUSCOD-II optimisation are presented in Figs. 14, 15 and 16.

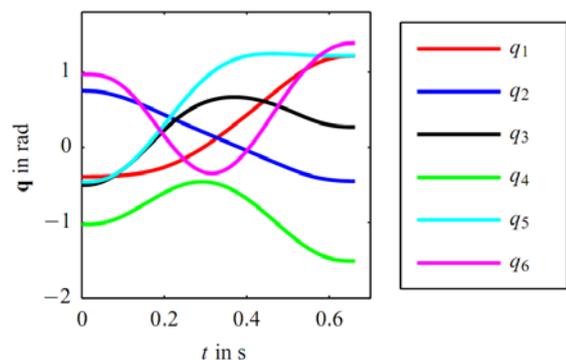


Figure 14 Joint angles: MUSCOD-II solution.

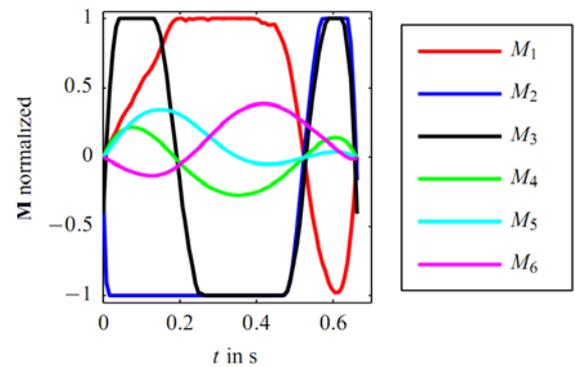


Figure 16 Normalised torques: MUSCOD-II solution.

4.3 COMPARISON

A comparison between both methods shows, that joint positions start at the same values, while the end position of the joints angles is different, cp. Figs. 11 and 14. Obviously, this effects the velocities. Regarding the motor torques (Figs. 13, 16), one can mention that in both cases, M_2 is most of the time either in positive or negative boundary. In the MUSCOD-II solution, the maximum torques are more often at their boundaries, while the joint velocities are in a quite similar range. An interesting and important comparison point is the total energy consumption

$$E = \int_0^{t_e} \mathbf{Q}^T \dot{\mathbf{q}} dt \quad (61)$$

for this specific task. Table II shows the summed up comparison for both algorithms.

Table II - Comparison: present approach, MUSCOD-II solution

	Present Approach	MUSCOD-II Solution
t_{cpu}	27 sec	150 sec
t_e	0.69 sec	0.66 sec
Energy	2270 Ws	2840 Ws

For the present approach, the task requires 20% lesser energy. Of course, this is a specific result for the chosen PtP problem and has to be evaluated for appropriate test scenarios to make a general statement. The task time t_e is for both methods nearly the same. The time t_{cpu} indicates the computational time for solving the optimisation problem on a standard PC. This time is about a factor of 5 lower for the present approach. There is also a code optimisation potential, since at the moment only non-optimised C- code is used.

5 CONCLUSION

This paper shows a method to get time optimal PtP trajectories for industrial robots. The main idea is to calculate optimal CP trajectories and vary in a second step intermediate points with a nonlinear solver. For the CP trajectories, we suggest a dynamic programming approach, since it delivers fast solutions. However, it is not important, how the time-optimal CP trajectories are computed. So the approach may also be interesting for robot manufacturers that have a solution for the CP problem and want to extend it to the PtP problem. A comparison with results evaluated with the multiple shooting method shows the effectivity of the approach and nearly the same end-time t_e for the trajectory. However, what has to be done in future is a proof of optimality which is not part of this paper. Including disturbing objects for specific cases will also be done in future works.

ACKNOWLEDGMENTS

This work has been supported by the Austrian COMET-K2 program of the Linz Center of Mechatronics (LCM), and was funded by the Austrian federal government and the federal state of Upper Austria.

REFERENCES

- [1] Bellman R. E. and Dreyfus S. E., *Applied Dynamic Programming*, Princeton Univ. Press, 1962.
- [2] Bobrow J. E., Dubowsky S. and Gibson J. S., Time-Optimal Control of Robotic Manipulators Along Specified Paths, *International Journal Robotics Research*, 4:3–17, 1985.
- [3] Bremer H., *Elastic Multibody Dynamics: A Direct Ritz Approach*, Springer Verlag, 2008.
- [4] Constantinescu D. and Croft E. A., Smooth and Time-optimal Trajectory Planning for Industrial Manipulators Along Specified Paths, *Journal of Robotic Systems*, 17(5):233–249, 2000.
- [5] De Boor C., *A Practical Guide to Splines*, Springer Verlag, 1978.
- [6] Gatringer H., *Starr-elastische Robotersysteme: Theorie und Anwendungen*, Springer Verlag, 2011.
- [7] Gatringer H., Riepl R. and Neubauer M., Optimizing Industrial Robots for Accurate High-Speed Applications, *Journal of Industrial Engineering*, 2013:1–12, 2013.
- [8] Geu Flores F. and Kecskemethy A., Time-Optimal Path Planning Along Specified Trajectories, In Gatringer H. and Gerstmayr J., eds, *Multibody System Dynamics, Robotics and Control*, pages 1–16, Springer Verlag, 2012.
- [9] Leineweber D., Bauer I., Schäfer A., Bock H. and Schlöder J., An Efficient Multiple Shooting Based Reduced SQP Strategy for Large-Scale Dynamic Process Optimization (Parts I and II), *Computers and Chemical Engineering*, 27:157–174, 2003.
- [10] Oberherber M., Gatringer H. and Springer K., A Time Optimal Solution for the Waiter Motion Problem with an Industrial Robot, *Proceedings of Austrian Robotics Workshop 13*, Vienna, pp. 55–60, 2013.
- [11] Pfeiffer F., *Mechanical System Dynamics*, Springer Verlag, 2008.
- [12] Pfeiffer F. and Johanni R., A Concept for Manipulator Trajectory Planning, *IEEE Journal of Robotics and Automation*, 3(2):115–123, 1987.
- [13] Piegl L. A. and Tiller W., *The NURBS Book*, Springer Verlag, 1995.
- [14] Shin K. and McKay N., Minimum-Time Control of Robotic Manipulators with Geometric Path Constraints, *IEEE Transactions on Automatic Control*, 30(6):531–541, 1985.
- [15] Verscheure D., Diehl M., De Schutter J. and Swevers J., Recursive Log-barrier Method for On-line Time-optimal Robot Path Tracking, *Proceedings of 2009 American Control Conference*, pp. 4134–4140, 2009.
- [16] Johnson St. G., *The nlopt Nonlinear-Optimization Package*, URL <http://ab-initio.mit.edu/nlopt>.
- [17] Powell M. J. D., *The BOBYQA Algorithm for Bound Constrained Optimization without Derivatives*, Technical Report NA2009/06, 2009.

IMPLEMENTATION OF A 3-DOF PARALLEL ROBOT ADAPTIVE MOTION CONTROLLER

J. Cazalilla*, M. Vallés*, V. Mata**, M. Díaz-Rodríguez***, A. Valera*

*Instituto de Automática e Informàtica Industrial, Universitat Politècnica de València

**Centro de Investigación de Tecnología de Vehículos, Universitat Politècnica de València

***Departamento de Tecnología y Diseño, Facultad de Ingeniería, Universidad de los Andes

ABSTRACT

For fast and accurate motion of a Parallel Manipulator, model-based control needs to be implemented. However, the dynamic model has uncertainties not only because of the unmodeled dynamics but also when, for instance, unknown inertial parameters can appear. To relax the requirement of exact knowledge, an adaptive controller has been developed in this paper. The controller is implemented in a modular way using Orocos, a real-time middleware and is compared with a fixed model passivity-based controller in simulation and experimental. Results show that the adaptive controller improves the trajectory tracking precision, compared to the fixed controller, when a payload is handled by the robot.

Keywords: Parallel manipulators, model-based control, adaptive robot control, control applications

1 INTRODUCTION

A Parallel Manipulator (PM) consists of a moving platform connected to a fixed base by at least of two kinematics chains. The end-effector in a PM is attached to the moving platform, so the load is shared by the kinematics chains connecting the moving platform to the fixed base. This fact gives to the PMs high stiffness, high load-carrying capacity and high accuracy. PMs are nowadays an active research field where several prototypes have been developed, for instances: motion simulators, tire-testing machines, flight simulators and medical applications [1-7]. In addition, PMs with very high accelerations, such as 200m/s^2 for the PAR4 manipulator [8] or 50m/s^2 for the Urane SX machine tool [9] have been proposed.

In spite of the advantages that PMs have over serial robots, the implementation of PMs in real applications is difficult. One of the difficulties lies on the controller design of PMs. Due to the inherent closed-loop constraints, the joints of PMs are tightly coupled and the dynamic characteristics are highly nonlinear [10]. The highly coupled dynamics makes it difficult to move a PM along a trajectory accurately and quickly. Moreover, the controller design can be a challenging work, which has aroused the interest many researchers in recent years [10 - 15].

In this paper, the dynamic controller design problem of a PM is addressed. The controller is implemented on a low-cost three degree-of-freedom (DOF) spatial PM. The robot was developed at Universitat Politècnica de València; its end effector is able to perform two angular rotations (rolling and pitching) and a linear motion (heave). The robot is equipped with an open control unit based-on industrial PC.

For the implementation of the model-based controller, exact knowledge of the system dynamics is required. The dynamic model of the PM implemented into the model-based control is based on a reduced model formulated in the joint space. The model is obtained by Gibbs-Appell equation and then the model is the reduced to a subset of identified relevant parameters [15]. The relevant parameters

Contact author: J. Cazalilla¹, M. Vallés¹, A. Valera¹, V. Mata², M. Díaz-Rodríguez³

¹ {jcazalilla, giuprog, mvalles} @ai2.upv.es

² vmata@mcm.upv.es

³ dmiguel@ula.ve

considered only those dynamic parameters with have significant influences on the robot dynamics; in addition the identified parameters are physically feasible. Due to this fact, the dynamic model has uncertainties. Moreover, an unknown inertial parameters can appears when a payload mass is grasped by the robot. Particularly, this kind of uncertainty limits the applicability of model-based controllers. To relax the requirement of exact knowledge, an adaptive controller for the PM has been adopted. The chosen control strategy is an adaptive passivity-based controller. One of the useful properties of these passivity-based tracking controllers is that the controller can easily be modified to account for parametric uncertainty of the robot dynamics. The controller takes advantages of the real-time middleware Orocos, allowing the control implementation in a modular way. To verify and validate the proper operation of the adaptive controller, a fixed passivity-based tracking controller has also developed and a comparison of these controllers is presented.

2 THE LOW-COST 3-DOF PARALLEL ROBOT

2.1 PHYSICAL DESCRIPTION OF THE LOW-COST PM

As mentioned before, a 3-DOF spatial PM was used for addressing the controller design problem. The robot consists of three kinematics chains; each chain has a PRS configuration (P, R, and S standing for prismatic, revolute, and spherical joint, respectively), The underlying format (P) stand for the actuated joint. The choice of the PRS configuration was guided by the need of developing a low-cost robot with 2 DOF of angular rotation in two axes (rolling and pitching) and 1-DOF translation motion (heave). In [16] a completed description of the mechatronic development process of the PM is presented.

The physical system consists of three legs connecting the moving platform to the base. Each leg consists of a motor driving a ball screw (prismatic joints) and a link with is lower part connected by a revolute joint to the ball screw. The upper part is connected to the moving platform through a spherical joint. The lower part of the ball screws are perpendicularly attached to the base platform. The positions of the ball screws at the base are in equilateral triangle configuration. The ball screw transforms the rotational movement of the motor into linear motion.

The motors in each leg are brushless DC servomotor equipped with power amplifiers. The actuators are Aerotech BMS465 AH brushless servomotors. The motors are operated by Aerotech BA10 power amplifiers.

In order to implement the control architecture for the parallel robot, an industrial PC has been used. It is based on a high performance 4U Rackmount industrial system with 7 PCI slots and 7 ISA slots. It has a 3,06GHz Intel® Pentium® 4 processor and two GB DDR 400 SDRAM. The industrial PC is equipped with 2 Advantech™ data acquisition cards: a PCI-1720 and a PCL-833.

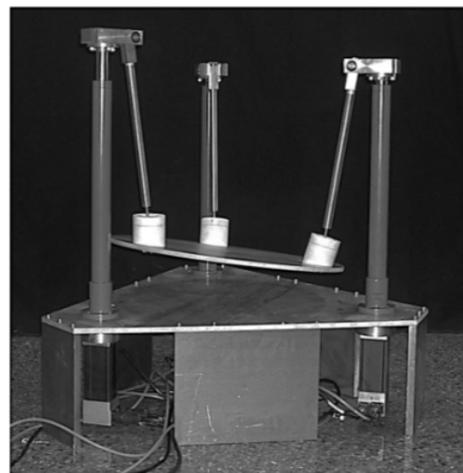


Figure 1 3-PRS parallel robot implemented

The PCI-1720 card has been used for supplying the control actions for each parallel robot actuator. It provides four 12-bit isolated digital-to-analog outputs for the Universal PCI 2.2 bus. It has multiple output ranges (0~5V, 0~10V, $\pm 5V$, $\pm 10V$), programmable software and an isolation protection of 2500 VDC between the outputs and the PCI bus. The PCL-833 card is a 4-axis quadrature encoder and counter add-on card for an ISA bus. The card includes four 32-bit quadruple AB phase encoder counters, an onboard 8-bit timer with a wide range time-based selector and it is optically isolated up to 2500V. Fig. 2 shows the control architecture based on an industrial PC developed for this study.

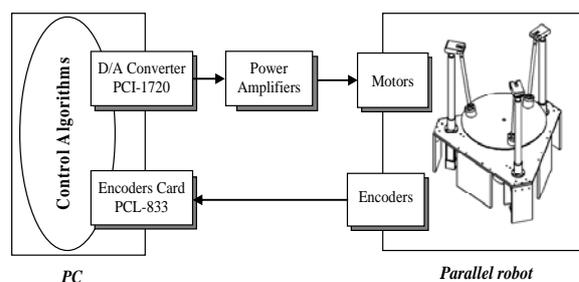


Figure 2 Robot control architecture.

The PC-based control system has two main advantages: First, it is a totally open and it gives a powerful platform for programming high level tasks based-on Ubuntu 12.04 operating system. Thus, any controller and/or control technique can be programmed and implemented, such as automatic trajectory generation, control based on external sensing using a force sensor or artificial vision, etc. The second main advantage is the low-cost of the robot. Because it is based on a PC running free and open-source software, the total cost of the control architecture doesn't exceed \$2000.

2.2 KINEMATIC MODEL

For control purpose both direct and inverse kinematics problem has to be solved. Given the actuators' linear motions, the direct kinematics of a PM consists of finding the roll (γ) and pitch (β) angles and the heave (z). The kinematic model is established by means of Denavit-Hartenbert (D-H), thus, 9 generalized coordinates are defined for modelling robot kinematics. The location of the coordinate systems is shown in Fig. 3.

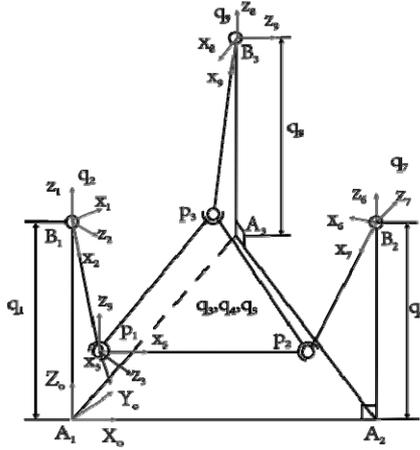


Figure 3 Location of the coordinate systems

From the figure it can be seen that the length between p_i and p_j is constant and equal to l_m . Thus, applying the geometric approach the kinematics model can be established as follows,

$$f_1(q_1, q_2, q_6, q_7) = \left\| (\vec{r}_{A_1B_1} + \vec{r}_{B_1P_1}) - (\vec{r}_{A_1A_2} + \vec{r}_{A_2B_2} + \vec{r}_{B_2P_2}) \right\| - l_m = 0 \quad (1)$$

$$f_2(q_1, q_2, q_8, q_9) = \left\| (\vec{r}_{A_1B_1} + \vec{r}_{B_1P_1}) - (\vec{r}_{A_1A_3} + \vec{r}_{A_3B_3} + \vec{r}_{B_3P_3}) \right\| - l_m = 0 \quad (2)$$

$$f_3(q_6, q_7, q_8, q_9) = \left\| (\vec{r}_{A_1A_3} + \vec{r}_{A_3B_3} + \vec{r}_{B_3P_3}) - (\vec{r}_{A_1A_2} + \vec{r}_{A_2B_2} + \vec{r}_{B_2P_2}) \right\| - l_m = 0 \quad (3)$$

In the forward kinematics the position of the actuators is known, thus the system of equations (1)-(3) is a nonlinear system with q_2 , q_7 and q_9 as unknown. The Newton-Raphson (N-R) numerical method is chosen to solve the nonlinear system. The method converges rather quickly (quadratic convergence) when the initial guess is close to the desired solution [17].

The location of the moving platform is defined using a local coordinate system attached to it. The coordinates of the spherical joints of the moving platform are obtained after having found the generalized coordinates of each leg of the robot. These three joints share the plane of the platform, so a local axis X_p is defined as a unit vector \vec{u} with the direction given by $p_1 p_2$. The axis Z_p is defined by a unit vector \vec{v} perpendicular to the plane defined by points p_1, p_2 and p_3 . Finally, the axis Y_p (axis \vec{w}) is determined by the

cross product $\vec{u} \times \vec{v}$. The rotation matrix of the moving platform is given by,

$${}^0R_p = \begin{bmatrix} \vec{u}^T & \vec{v}^T & \vec{z}^T \end{bmatrix} \quad (4)$$

The remaining generalized coordinates (q_3, q_4 and q_5) are found from the rotation matrix.

On the other hand, the inverse kinematics consists of finding the actuators' linear motion given the roll (γ) and pitch (β) angle and the heave (z). Using an X-Y-Z fixed-angle system, the rotational matrix can be defined as,

$${}^0R_p = \begin{bmatrix} c_\alpha c_\beta & c_\alpha s_\beta s_\gamma - s_\alpha c_\gamma & c_\alpha s_\beta c_\gamma - s_\alpha s_\gamma \\ s_\alpha c_\beta & s_\alpha s_\beta s_\gamma - c_\alpha c_\gamma & s_\alpha s_\beta c_\gamma - c_\alpha s_\gamma \\ -s_\beta & c_\beta s_\gamma & c_\beta c_\gamma \end{bmatrix} \quad (5)$$

In the above equation, c_* and s_* stand for $\cos(*)$ and $\sin(*)$, respectively. Given γ and β the yaw angle (α) can be found as follows,

$$\alpha = \text{atan2}(s_\beta s_\gamma, (c_\gamma + c_\beta)) \quad (6)$$

Having found the angle α , the remaining terms of the rotational matrix can be found. The actuator positions can be found by the following expressions [3],

$$q_1 = p_z - \frac{1}{2}v_z h - \frac{1}{2}u_z h \sqrt{3} - \frac{1}{2} \left(-2u_y h^2 \sqrt{3} v_y - v_y^2 h^2 - 4p_x^2 + 4p_x u_x h \sqrt{3} u_y - 4u_y^2 h^2 - 4p_y^2 + 4p_y u_y h \sqrt{3} + 4p_y v_y h + 4l_r^2 \right)^{1/2} \quad (7)$$

$$q_6 = p_z + \frac{1}{2}u_z h \sqrt{3} - \frac{1}{2}v_z h - \frac{1}{2} \left(-4p_x h \sqrt{3} u_x + 2u_x h^2 \sqrt{3} u_y - 4p_y u_y h \sqrt{3} + 2u_y^2 h^2 \sqrt{3} + 8p_x \sqrt{3} g + 12u_x h g + 4p_x u_y h - 4p_x^2 - 4u_y^2 h^2 - 4p_y^2 - v_y^2 h^2 - 3u_x^2 h^2 + 4l_r^2 \right)^{1/2} \quad (8)$$

$$q_8 = p_z + v_z h - (-3v_y h g - p_x^2 - 2p_x u_y h + g \sqrt{3} p_x - u_y^2 h^2 + h g \sqrt{3} u_y - 3g^2 - p_y^2 - 2p_y v_y h + 3p_y g - v_y^2 h^2 + l_r^2)^{1/2} \quad (9)$$

where $h = l_m / \sqrt{3}$, $g = l_b / \sqrt{3}$, $p_x = -hu_y$, $p_y = -h(u_x - v_y)$, $p_z = z$ and l_b are the lengths between $A_i A_j$.

2.3 DYNAMIC MODEL

One of the goals of this paper is to develop an open control architecture allowing the implementation and testing of dynamic control schemes. This kind of dynamic controllers requires describing the equation of motion as follows,

$$M(\vec{q}, \vec{\Phi}) \cdot \ddot{\vec{q}} + \vec{C}(\vec{q}, \dot{\vec{q}}, \vec{\Phi}) \cdot \dot{\vec{q}} + \vec{G}(\vec{q}, \vec{\Phi}) = \vec{\tau} \quad (10)$$

From equation (10) it can be seen that the system mass matrix M , the vectors corresponding to the centrifugal and Coriolis forces C , and the gravitational forces G depend on the dynamic parameters $\bar{\Phi}$ and the external generalized forces $\bar{\tau}$.

In order to identify the dynamic parameters, the model in linear parameter form has to be build first as follows as in [15],

$$K(\bar{q}, \dot{\bar{q}}, \ddot{\bar{q}}) \cdot \bar{\Phi} = \bar{\tau} \quad (11)$$

In equation (11), $K(\bar{q}, \dot{\bar{q}}, \ddot{\bar{q}})$ is the observation matrix corresponding to the set of generalized coordinates, velocities and accelerations. For this parallel robot, a complete and reduced model can be obtained [18]. The complete model contains all the rigid body dynamic parameters affecting the dynamic behavior of the robot has been obtained. This model consists of the Coulomb and viscous friction parameters, the rotor and screw dynamics of the robot actuators and the rigid body base parameters.

However, not even those parameters could always be properly identified in this base parameters model. Thus, the reduced model contains only the relevant parameters obtained through a process which considers the robot's leg symmetries, the statistical significance of the identified parameters, and the physical feasibility of the parameters. The equations of robot motion have several fundamental properties that can be exploited to facilitate dynamic controllers design. One of the useful properties is that there is a reparametrization of all unknown parameters into a parameter vector $\bar{\Phi} \in R^p$ that enters linearly in the system dynamics (11). Therefore, the following holds,

$$M(\bar{q}, \bar{\Phi}) \cdot \ddot{\bar{q}} + \bar{C}(\bar{q}, \dot{\bar{q}}, \bar{\Phi}) \cdot \dot{\bar{q}} + \bar{G}(\bar{q}, \bar{\Phi}) = M_0(\bar{q}) \cdot \ddot{\bar{q}} + \bar{C}_0(\bar{q}, \dot{\bar{q}}) \cdot \dot{\bar{q}} + \bar{G}_0(\bar{q}) + Y(\bar{q}, \dot{\bar{q}}, \ddot{\bar{q}}) \bar{\Phi} \quad (12)$$

where $M_0(\cdot)$, $\bar{C}_0(\cdot)$, $\bar{G}_0(\cdot)$ represent the know part of system dynamics, and $Y(u, v, w, x)$ is a regressor matrix of dimension $[n \times p]$ that contains nonlinear but known functions.

As a consequence of this property, the left hand side of (10) can be written as,

$$M_0(\bar{q}) \cdot \ddot{\bar{q}} + \bar{C}_0(\bar{q}, \dot{\bar{q}}) \cdot \dot{\bar{q}} + \bar{G}_0(\bar{q}) + Y(\bar{q}, \dot{\bar{q}}, \ddot{\bar{q}}) \bar{\Phi} = \bar{\tau} \quad (13)$$

Because the actual 3-PRS parallel robot reduced dynamic model has 12 parameters (3 of rigid body base parameters, 3 of the actuator dynamics and 6 of Coulomb and viscous friction), it can be expressed as,

$$M(\bar{q}) \cdot \ddot{\bar{q}} + \bar{C}(\bar{q}, \dot{\bar{q}}) \cdot \dot{\bar{q}} + \bar{G}(\bar{q}) = \bar{\tau} \quad (14)$$

where

$$M(\bar{q}) \cdot \ddot{\bar{q}} = \begin{bmatrix} J_1 & 0 & 0 \\ 0 & J_2 & 0 \\ 0 & 0 & J_3 \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix} + \begin{bmatrix} M_{11}(q) & M_{12}(q) & M_{13}(q) \\ M_{21}(q) & M_{22}(q) & M_{23}(q) \\ M_{31}(q) & M_{32}(q) & M_{33}(q) \end{bmatrix} \begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \end{bmatrix} \quad (15)$$

$$C(\bar{q}, \dot{\bar{q}}) \cdot \dot{\bar{q}} = \begin{bmatrix} F_{v_1} \dot{q}_1 + F_{c_1} \text{sign}(\dot{q}_1) \\ F_{v_2} \dot{q}_2 + F_{c_2} \text{sign}(\dot{q}_2) \\ F_{v_3} \dot{q}_3 + F_{c_3} \text{sign}(\dot{q}_3) \end{bmatrix} + \quad (16)$$

$$\begin{bmatrix} C_{11}(\bar{q}, \dot{\bar{q}}) & C_{12}(\bar{q}, \dot{\bar{q}}) & C_{13}(\bar{q}, \dot{\bar{q}}) \\ C_{21}(\bar{q}, \dot{\bar{q}}) & C_{22}(\bar{q}, \dot{\bar{q}}) & C_{23}(\bar{q}, \dot{\bar{q}}) \\ C_{31}(\bar{q}, \dot{\bar{q}}) & C_{32}(\bar{q}, \dot{\bar{q}}) & C_{33}(\bar{q}, \dot{\bar{q}}) \end{bmatrix} \begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \end{bmatrix}$$

$$G(\bar{q}) = g \begin{bmatrix} G_{11}(\bar{q}) & G_{12}(\bar{q}) & G_{13}(\bar{q}) \\ G_{21}(\bar{q}) & G_{22}(\bar{q}) & G_{23}(\bar{q}) \\ G_{31}(\bar{q}) & G_{32}(\bar{q}) & G_{33}(\bar{q}) \end{bmatrix} \begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \end{bmatrix} \quad (17)$$

Therefore, different combinations can be considered according with the unknown robot parameters. For example, if the rigid body parameters constituting the reduced model are assumed to be unknown, then (13) can be written as,

$$\bar{\tau} = \begin{bmatrix} J_1 & 0 & 0 \\ 0 & J_2 & 0 \\ 0 & 0 & J_3 \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix} + \begin{bmatrix} F_{v_1} \dot{q}_1 + F_{c_1} \text{sign}(\dot{q}_1) \\ F_{v_2} \dot{q}_2 + F_{c_2} \text{sign}(\dot{q}_2) \\ F_{v_3} \dot{q}_3 + F_{c_3} \text{sign}(\dot{q}_3) \end{bmatrix} + Y_1(\bar{q}, \dot{\bar{q}}, \ddot{\bar{q}}) \bar{\theta}_1 \quad (18)$$

Where

$$Y_1(\bar{q}, \dot{\bar{q}}, \ddot{\bar{q}}) = \begin{bmatrix} M_{11}(q) & M_{12}(q) & M_{13}(q) \\ M_{21}(q) & M_{22}(q) & M_{23}(q) \\ M_{31}(q) & M_{32}(q) & M_{33}(q) \end{bmatrix} + \begin{bmatrix} C_{11}(\bar{q}, \dot{\bar{q}}) & C_{12}(\bar{q}, \dot{\bar{q}}) & C_{13}(\bar{q}, \dot{\bar{q}}) \\ C_{21}(\bar{q}, \dot{\bar{q}}) & C_{22}(\bar{q}, \dot{\bar{q}}) & C_{23}(\bar{q}, \dot{\bar{q}}) \\ C_{31}(\bar{q}, \dot{\bar{q}}) & C_{32}(\bar{q}, \dot{\bar{q}}) & C_{33}(\bar{q}, \dot{\bar{q}}) \end{bmatrix} + \quad (19)$$

$$g \begin{bmatrix} G_{11}(\bar{q}) & G_{12}(\bar{q}) & G_{13}(\bar{q}) \\ G_{21}(\bar{q}) & G_{22}(\bar{q}) & G_{23}(\bar{q}) \\ G_{31}(\bar{q}) & G_{32}(\bar{q}) & G_{33}(\bar{q}) \end{bmatrix}$$

$$\bar{\theta}_1 = [\Omega_1 \quad \Omega_2 \quad \Omega_3]^T \quad (20)$$

3 ADAPTIVE MODEL-BASED PM CONTROL

3.1 ADAPTIVE CONTROL SCHEME

It is possible to find in the literature different adaptive control schemes that do not suffer from the parameter drift problem. For example, Bayard and Wen have developed in [19] a class of adaptive robot motion controllers, but in this

work the following one has been developed for the parallel robot:

$$\tau_c = M_0(\bar{q})\ddot{\bar{q}}_d + \bar{C}_0(\bar{q}, \dot{\bar{q}}_d)\dot{\bar{q}}_d + \bar{G}_0(\bar{q}) + Y(\bar{q}, \dot{\bar{q}}_d, \ddot{\bar{q}}_d)\bar{\theta} - K_d\bar{e} - K_p\bar{e} \quad (21)$$

$$\frac{d}{dt} \{\hat{\theta}(t)\} = -\Gamma_0 Y^T(\bar{q}, \dot{\bar{q}}_d, \ddot{\bar{q}}_d) \bar{s}_1 \quad (22)$$

where $\bar{s}_1 = \bar{e} + \Lambda_1 \dot{\bar{e}}$, with $\Lambda_1 = \lambda_1 I$ and $\lambda_1 > 0$.

The close-loop system (14)-(21)-(22) is convergent, that is the tracking error asymptotically converge to zero and all internal signals remain bounded, under a suitable conditions on the controller gains K_p and K_d .

3.2 CONTROL SIMULATIONS

To validate the correct operation of the adaptive control algorithm, several Matlab/Simulink schemes for the parallel robot simulation has been developed. Fig. 4 shows the scheme implemented for the adaptive controller. Simulink block $Y(q, dq, ddq)$ implements the regressor matrix of equation (19). *Inertial Terms M0* and *Coriolis Terms C0* blocks implement the know part of the robot dynamics (equations (13) or (18)). Finally, *PD* block implements the proportional-derivative term.

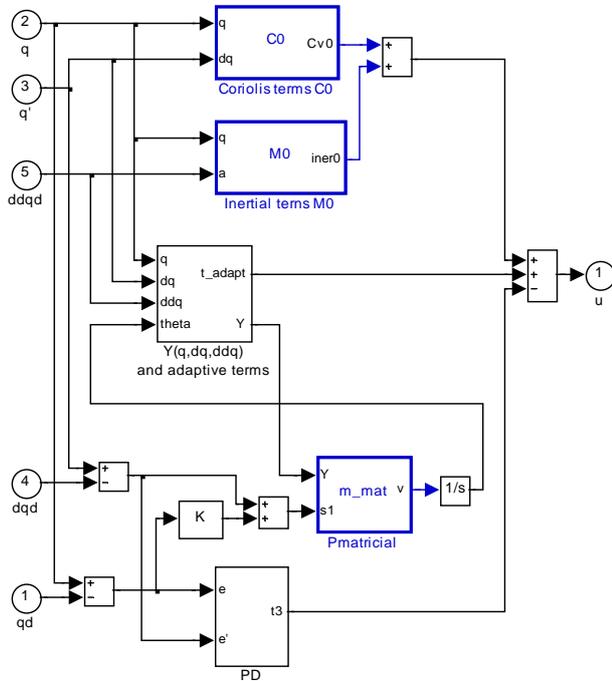


Figure 4 Adaptive controller simulation scheme

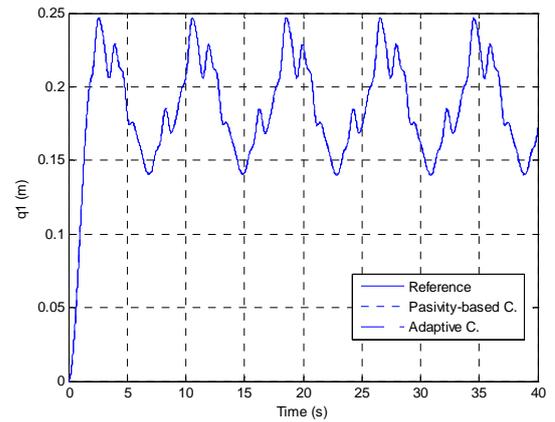
On the other hand, in order to verify the adaptive controller features, a passivity-based trajectory tracking controller has been also implemented. The control law considered is [20]:

$$\tau_c = M(\bar{q})\ddot{\bar{q}}_d + C(\bar{q}, \dot{\bar{q}})\dot{\bar{q}}_d + G(\bar{q}) - K_d\bar{e} - K_p\bar{e} \quad (23)$$

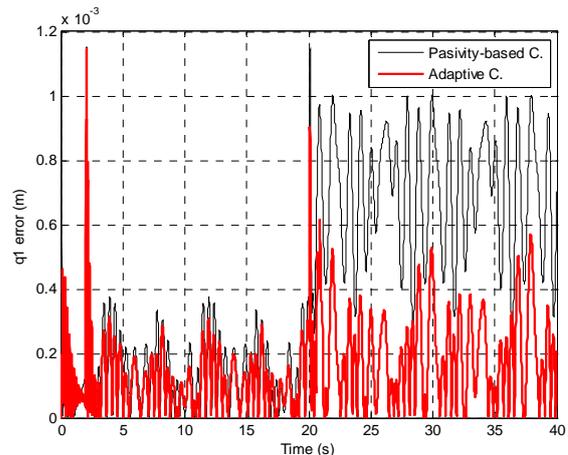
This passivity-based controller has been chosen because it has very good robust properties and because its expression is similar to the adaptive controller developed for this work, so it is easy to compare and analyze their characteristics.

As mentioned before, because the reduced robot model has 12 parameters, the adaptive scheme can be developed for different cases, depending on which parameters are considered unknown. In this work, the adaptive controller developed considers only rigid body parameters, so the robot model is expressed using equation (14)-(17).

In this way the following figures show the references and the positions obtained with an adaptive controller and a passivity-based dynamic controller, and the absolute position error. In the simulation it has been considered that at $t=20$ sec a mass of 30 kg was placed in the mobile platform m_3 , so it changed from 12kg to 42kg.



(a)



(b)

Figure 5 Position (a) and absolute error (b) of the first actuated joint.

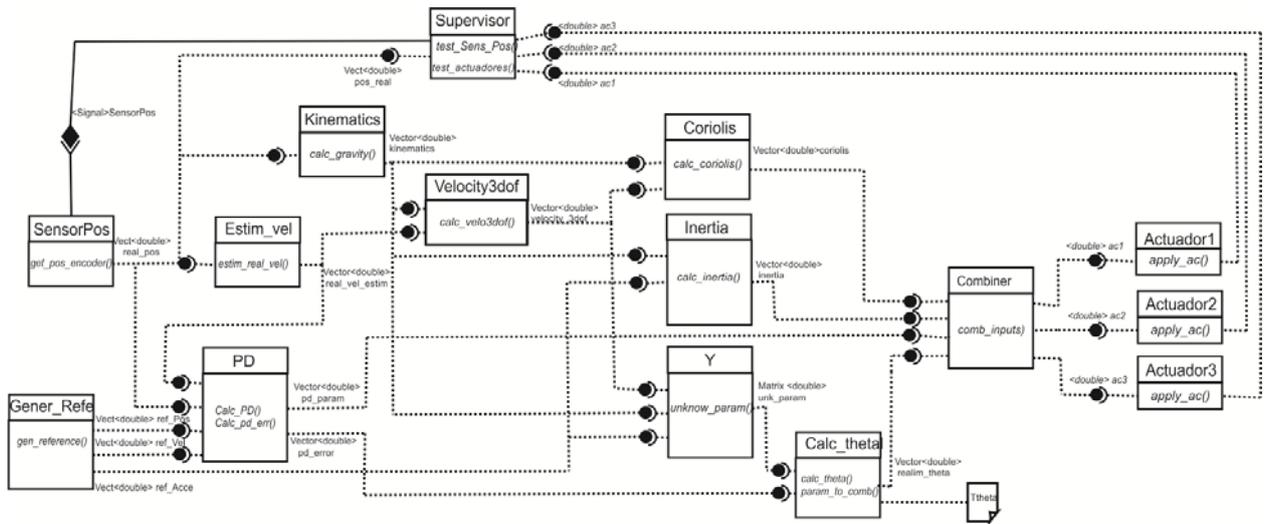


Figure 6 Adaptive control flowchart implemented in Orocos

As it can readily be appreciated in the Fig. 5, the error that is discussed with both controllers before modifying m_3 is very similar. However, after modifying this mass, the adaptive controller response is much better since the passivity-based controller uses wrong values of some dynamic parameters. Note that very similar results has been obtained for second and third actuated joint.

3.3 EXPERIMENTS WITH THE CURRENT LOW-COST ROBOT

In addition to the simulations schemes, in this work, the passivity-based and the adaptive controllers described before have been developed in a modular way using a real-time middleware.

In particular, the middleware used is Orocos (Open Robot COntrol Software), which provides the main features of a component-based middleware: creation of an abstraction layer between the operating system and the application layer and communication infrastructure component-based model. The Orocos project consists of a series of libraries and tools, being the most important the *OrocosToolchain*. This tool includes the two main libraries (RTT and OCL) to create the components and control schemes. Using this component-based middleware, we have developed a modular structure for the Paden-Panja control (Eq. 23) and adaptive control (Eq. 21).

Furthermore, component-based software development (with Orocos) has a number of advantages such as:

- easy flow tracking execution.
- distributed execution, with each component in a different thread, reducing the execution time.
- code-reusability. Note that the two controllers implemented (Eq. 21 and 23) have very similar structure. Thus, a single component has to be implemented only once and can be used on any other scheme many times.

Using the parallel robot and its open hardware and software control system, different control algorithms have been developed and tested. It's remarkable that all the schemes has been made in a modular way (using Orocos), with a cascade control and a frequency of 100Hz ($t_{\text{sample}}=10$ ms). For example, in Fig. 6 can be seen the adaptive control implemented in a modular way using Orocos. As commented above, the execution is in cascade, being the *SensorPos* component in charge of waking up the other components. Using this technique, a distributed execution is performed, decreasing execution time.

The real executions have shown that the robot response is very good and, if a payload is added to the moving platform, see Fig. 7, a direct change occurs in the estimation of the rigid body parameters, which is not the case for the viscous friction.

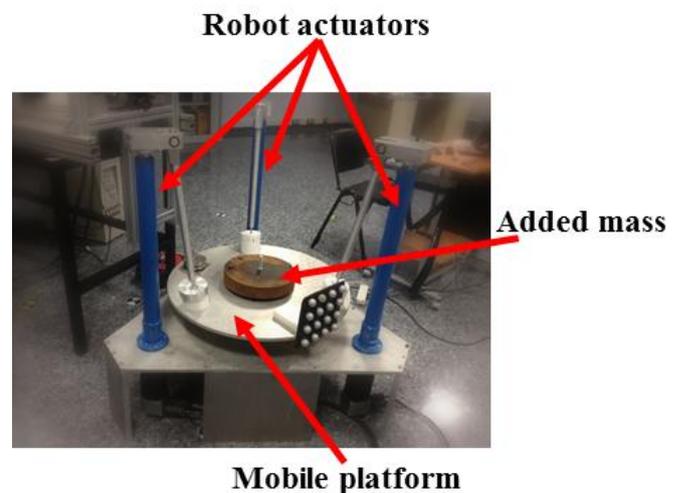


Figure 7 Actual parallel robot with the 30kg load placed on the mobile platform

The following figures show the response obtained from the actual robot: Fig.8 shows the reference and the robot q_1 positions for the adaptive and the passivity-based controllers. Fig. 9 shows the absolute error values of passivity-based and adaptive controllers. Fig. 10 shows the control action (in volts.) provided by the adaptive controller. The motion references are very similar as the references used in simulation. The only difference is that in the middle of execution, the robot remains in the same position for 8 seconds (between $t = 85$ and $t = 93$ sec). This time allows us to place a load of 30kg on the robot platform.

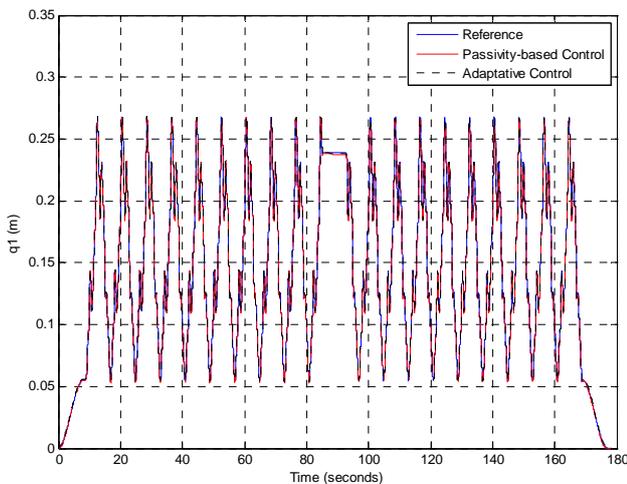


Figure 8 Reference and robot positions (passivity-based and adaptive controllers) for the first actuated joint.

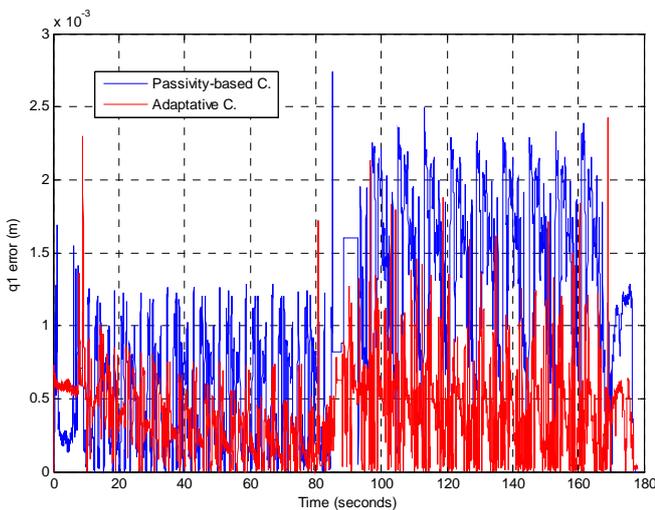


Figure 9 Absolute error position for the first actuated joint

The results obtained with the actual robot agree with those obtained in the simulation: because of the estimation the on-line dynamic parameters, the change of the load means that the robot response using an adaptive controller is significantly better than that obtained with the passivity-based controller.

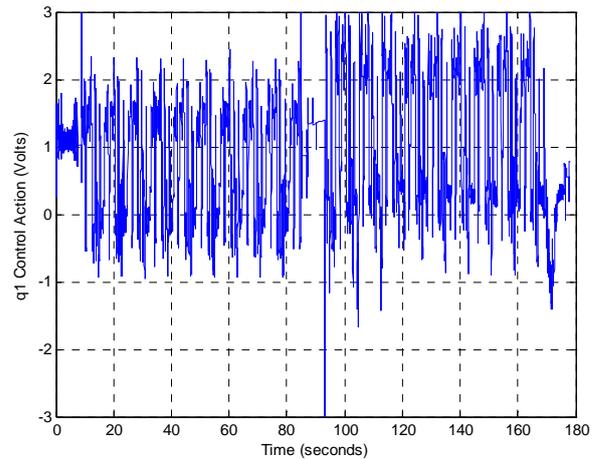


Figure 10 Control action (adaptive controller) for the first actuated joint

The mean squared error of both controllers can be seen in Tab. I. There, one can observe that during the first 90 seconds (without payload) the error with an adaptive and a passivity-based control is very similar. However, after placing a load of 30 kg on the robot platform (the next 90 seconds), the adaptive control works much better than the other one. This is because the adaptive control calculates the new dynamic parameters on-line (after placing the charge). However, since the passivity-based control doesn't calculate the dynamic parameters online, after putting the weight, the error increases significantly.

Table I - Mean squared error (MSE) of adaptive and passivity-based controllers.

	$\sqrt{\frac{\sum_{j=1}^n \sum_{i=1}^{DOF} (e_{i,j})^2}{n \cdot DOF}}$	
	Adaptive	Passivity-Based
Without payload	$4.3 \cdot 10^{-4}$	$6.2 \cdot 10^{-4}$
With payload	$5.6 \cdot 10^{-4}$	$1.4 \cdot 10^{-4}$

4 CONCLUSIONS

In this paper, the adaptive control of a 3-DOF parallel manipulator was considered. The adaptive controller is based on a reduced robot dynamic model. This model contains only a set of relevant parameters obtained through a process which considers the robot's leg symmetries, the statistical significance of the identified parameters, and the physical feasibility of the parameters. The reduced dynamic model has uncertainties not only because of the unmodeled dynamics but also when unknown inertial parameters can appear, for instances, when a payload mass is grasped by the robot. This kind of uncertainty limits the applicability of model-based controllers. To relax the requirement of exact knowledge, an adaptive controller for the PM was implemented. The adaptive scheme can be rewritten

depending on the robot parameters that are assumed to be unknown.

In order to analyze and validate the control algorithms, they have been tested on a virtual and an actual prototype of a parallel robot. The simulations of the virtual robot were developed in Matlab/Simulink. The actual prototype is a low-cost, parallel robot developed at Universitat Politècnica de Valencia.

The control of the actual robot has been implemented in Orocos middleware. Because it is a component-based middleware, Orocos provides several advantages like modular design and structure, reusable code, modules reconfiguration in real-time.

Using Orocos, an adaptive and a passivity-based controller have been developed. The results indicate that the adaptive controller perform better than the passivity-based controller if there are differences between the supposed and the real parameters used in the robot dynamic model.

ACKNOWLEDGEMENTS

The authors want to express their gratitude to the Plan Nacional de I+D, Comisión Interministerial de Ciencia y Tecnología (FEDER-CICYT) for the partial financing of this research under the projects DPI2010-20814-C02-(01, 02) and DPI2011-28507-C02-01. This research was also partially funded by the CDCHT-ULA Grant I-1286-11-02-B

REFERENCES

- [1] Stewart, D.A. 1965. A platform with 6 degree of freedom. Proceedings of the Institution of mechanical engineers Part 1, Vol. 15, pp.371-386.
- [2] Merlet, J.P. 2000. Parallel Robots. Kluwer, London, U.K.
- [3] Tsai, L.W. 1999. Robot Analysis: The Mechanics of Serial and Parallel Manipulator. Wiley Interscience, Canada.
- [4] Li Y. and Q. Xu. 2007. Design and Development of a Medical Parallel Robot for Cardiopulmonary Resuscitation. IEEE/ASME Transaction on Mechatronics, Vol. 12, pp.265-273.
- [5] Chablat D. and P. Wenger. 2003. Architecture optimization of a 3-DOF translational parallel mechanism for machining applications, the Orthoglide. IEEE Transactions on Robotics and Automation, Vol. 19, pp.402-410
- [6] Carretero J.A., R.O. Podhorodeski and M.A. Nahon. 2000. Kinematic analysis and optimization of a new three degree-of-freedom spatial parallel manipulator. Journal of Mechanical Design, Vol. 122, pp.17-24
- [7] Merlet, J.P. 2002. Optimal design for the micro parallel robot MIPS. In Proceedings of IEEE Int. Conf. on Robotics and Automation, pp.1149-1154.
- [8] Nabat V. and M. de la O Rodriguez, O. Company, S. Krut and F. Pierrot. 2005. PAR4: very high speed parallel robot for pick-and-place. In Proceedings of Int. Conf. on Intelligent Robots and Systems (IROS'05), Edmonton, Canada, pp.1202-1207.
- [9] Company O. and F. Pierrot. 2002. Modelling and design issues of a 3-axis parallel machine-tool. Mechanism and Machine Theory, Vol. 37, pp.1325-1345.
- [10] Zhan Y.X., S. Cong, W.W. Shang, Z.X. Li and S.L. Jiang. 2007. Modeling, Identification and Control of a Redundant Planar 2-DOF Parallel Manipulator. International Journal of Control, Automation and Systems, Vol. 5, pp.559-569.
- [11] Fu K. and J.K. Mills. 2007. Robust control design for a planar parallel robot. International Journal of Robotics & Automation, Vol. 22, pp.139-147.
- [12] Stan S.D., R.Balan, V. Maties and C. Rad. 2009. Kinematics and fuzzy control of ISOGLIDE3 medical parallel robot. Mechanika, Vol. 75, pp.62-66.
- [13] Gou H.B., Y.G. Liu, F.R. Liu and H.R. Li. 2009. Cascade control of a hydraulically driven 6-dof parallel robot manipulator based on a sliding mode. Control Engineering Practice, Vol. 16, pp.105-1068.
- [14] Abdellatif, H. and B. Heimann. 2010. Advanced Model-Based Control of a 6-DOF Hexapod Robot: A Case Study. IEEE/ASME Transactions on Mechatronics, Vol. 15, pp.269-279.
- [15] Díaz-Rodríguez M., V. Mata, A. Valera and A. Page. 2010. A methodology for dynamic parameters identification of 3-DOF parallel robots in terms of relevant parameters. Mechanism and Machine Theory, Vol. 45, pp.1337-1356.
- [16] Valles M., M. Díaz-Rodríguez, A. Valera, V. Mata and A. Page. 2012. Mechatronic Development and Dynamic Control of a 3-DOF Parallel Manipulator. Mechanics Based Design of Structures and Machines, Vol. 40, pp.434-452.
- [17] García de Jalón J. and E. Bayo. 1994. Kinematic and Dynamic Simulation of Multibody Systems: The Real-Time challenge. Springer-Verlag, New-York
- [18] Diaz-Rodríguez M., A. Valera, V. Mata V and M. Valles. 2012. Model-Based Control of a 3-DOF Parallel Robot Based on Identified Relevant Parameters. IEEE/ASME Trans on Mechatronics. DOI: 10.1109/TMECH.2012.2212716
- [19] Bayard, D. S. and J.T. Wen. 1988. New class of control laws for robotic manipulators Part 2. Adaptive case. International journal of control, Vol. 47 (5), pp.1387-1406.
- [20] Paden, B. and R. Panja. 1988. Globally Asymptotically Stable 'PD+' Controller for Robot Manipulators. International Journal on Control, Vol. 47, pp.1697-1712.

WEARABLE SENSORY SYSTEM FOR ROBOTIC PROSTHESIS

Luka Ambrožič*, Maja Gorišič*, Sebastjan Šlajpah*, Roman Kamnik*, and Marko Munih*

*Laboratory of Robotics, Faculty of Electrical Engineering, University of Ljubljana, Slovenia

ABSTRACT

The paper presents a wearable sensory system aimed at tracking of motion parameters and estimation of kinematic data of the wearer for use in controlling active lower-limb ortho-prostheses. The sensory system comprises inertial and magnetic measurement units (IMUs) attached to body segments and sensorized insoles worn inside sneaker shoes. Through sensory fusion, the IMUs data is used to produce estimates of segment orientations, while the insoles provide information on vertical ground reaction force amplitude and distribution. The paper outlines the principles of the algorithms and shows the evaluation of them. Algorithms use both reference and wearable sensory data to extract information about the subject's kinematics, movement type and phase and track selected biomechanical stability descriptors. The paper discusses preliminary experimental results of the proposed algorithms.

Keywords: Wearable sensors, Active prosthesis, Sensory Fusion, Movement identification, Phase segmentation, Stability tracking.

1 INTRODUCTION

Loss of a lower-limb can be a great obstacle in the life of persons following lower-limb amputation. It may affect the person's general health. Amputation can be performed at different levels, such as foot level, knee level or at the level of the thigh. Above-knee amputation is usually the most stressful of amputation types as it greatly diminishes the person's natural area of movement [1]. Persons following amputation tend to consume more metabolic energy than healthy persons and require greater level of mental effort to move their body without falling [2]. The goal of the EU-funded CYBERLEGs project (The CYBERnetic LowEr-Limb CoGnitive Orto-prosthesis) is aimed at development of a robotic, wearable ortho-prosthesis.

Since currently available passive and active prostheses do not greatly diminish effort and energy consumption, the aim of the project is the development of a cognitive robotic system for persons following above-knee amputation that enables them to perform previously demanding movement manoeuvres with minimal mental effort and energy consumption. The manoeuvres of interest are steady-state

ground-level walking, stair ascent and descent as well as sit-to-stand and stand-to-sit movement. Laboratory of Robotics at the Faculty of Electrical Engineering, University of Ljubljana is involved in development of a wearable sensory system, sensory fusion, cognitive decision making and intention detection algorithms alongside tracking of stability parameters during movements.

In laboratory environment, systems used for assessment of kinematics parameters are usually based on an optical principle with the use of passive (Vicon) or active markers (Optotrak) [3]. Such laboratory systems have high accuracy but they are large, expensive, stationary, and have limited measurement space. Recently, the use of inertial and magnetic sensor for assessment of human kinematics parameters has become a common practice [4], [5]. Inertial sensors are less accurate compared to laboratory systems, however, they are lighter, cheaper, wearable and do not alter natural movement patterns. As such they can also be used outside the laboratory environment.

Fusion of sensory signals from inertial and magnetic sensors placed on body segments is used to assess the segments' orientation. Kalman filtering is a common approach for this task, where orientation is estimated by integrating angular velocity [6], [7]. Due to the drift of integrated gyroscope output and temperature dependency, the orientation error tends to grow with time.

Contact author: Luka Ambrožič¹, Roman Kamnik¹
¹{luka.ambrozic, roman.kamnik}@robo.fe.uni-lj.si
Laboratory of Robotics, Faculty of Electrical Engineering,
University of Ljubljana
Tržaška cesta 25, 1000 Ljubljana, Slovenia

To compensate for the error of integration, the integrated orientation estimate is fused with the orientation estimate obtained from accelerometer and magnetometer data.

The sensory system, via sensory fusion algorithms, should provide the cognitive system with information on the user's kinematic parameters while the cognitive system provides information about the user's current movement state and intention to the controller. Additionally, tracking of stability parameters should be incorporated into the cognitive system, in order to provide a basis for decisions on reactive movement when a sudden loss of balance is imminent.

This paper is divided into four parts that comprise the cognitive system of CYBERLEGS and build upon each other. The first section describes the wearable hardware used for extracting information on user movement and intention as well as body kinematic parameters. The second part describes the principles upon which the sensory fusion and extraction of kinematic parameters are based and also provides some preliminary evaluation data of the proposed algorithms. Based on sensory data, part three presents a cognitive machine that identifies current movement manoeuvre that user is performing and classifies phases within that particular movement type. The system provides data on steady states as well as transitions between these states. Furthermore, part four describes tracking of independent particular stability descriptive parameters are tracked based on a statistical anthropometric model, estimated kinematic parameters and directly measured sensory information.

2 WEARABLE SENSORY SYSTEM HARDWARE

The CYBERLEGS sensory and feedback system uses commercially-available components and custom-made sensing components, developed within the CYBERLEGS Consortium (see schematic shown in Figure 1). As the figure shows, the system comprises several wearable sensors - sensorized insoles, pressure-sensitive pads to measure human-robot interaction, inertial measurement units (IMUs), vibrotactile modules for afferent feedback and sensors for detection of amputee psychophysiological stress status - in addition to sensors for controlling the actuation system and measurement of the joint positions of the ortho-prosthesis. Communication with the main controller employs both UDP and SPI protocols, while sensory fusion runs on a separate real-time OS (xPC Target) based machine.

The inertial measurement units are used for assessing the orientation of human body segments, and shoe insoles for measuring ground reaction forces. The sensory data acquisition unit consists of two wireless receiver units (RU) for fetching data via 802.15.4 (ZigBee) and Bluetooth protocols. Transfer of acquired data employs Ethernet UDP communication to the controller that runs the tools for data processing and sensory fusion algorithms.

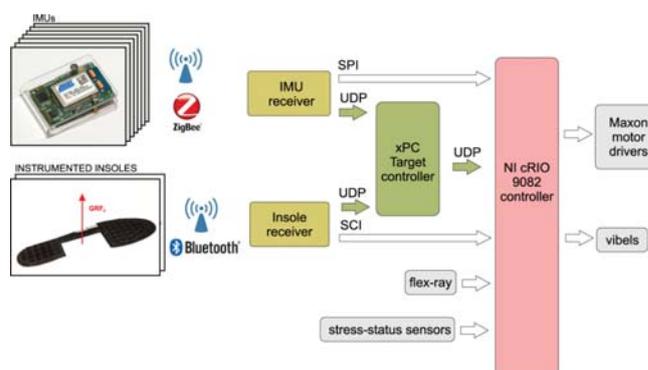


Figure 1 CYBERLEGS wearable sensory system: sensors and data acquisition unit.

2.1 SENSORIZED INSOLES

The CYBERLEGS pressure-sensitive insoles, developed at Scuola Superiore Sant' Anna, Pisa, Italy [8], comprise an array of 64 pressure sensors and fit into normal sneaker shoes. Each cell has a working range from 0 to 70 N. They are wireless, run on battery power and output vertical ground reaction force estimate as well as a distribution of this force along the sole (Centre of Pressure - COP).

2.2 INERTIAL MEASUREMENT SYSTEM

An inertial and magnetic measurement unit (Figure 2) consists of three sensors which measure 3D vectors of angular velocity (range ± 500 °/s), translational acceleration (range ± 2 G), and magnetic field (range ± 1.3 Ga) and is equipped with an on-board 8-bit processor [9]. The size of the IMU without the battery is $30 \times 20 \times 5$ mm. For measuring kinematic parameters of the human body, seven IMUs are used. Each of the IMU is placed on an individual segment of lower extremities: feet, shanks, thighs, and trunk. Placement of the IMU on the segment is determined with initial evaluation of IMU orientation during standstill.

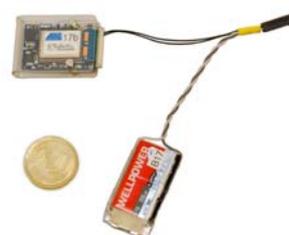


Figure 2 Inertial measurement unit with a battery.

3 ASSESMENT OF KINEMATIC PARAMETERS

Kinematics of the human body can be estimated using data provided by wearable sensors. Based on sensory signals from seven IMUs, we have developed an algorithm for kinematics parameters assessment by means of sensory data fusion.

A common approach for determining segment orientations is the use of an Unscented Kalman filter applied to the measurement data [9]. The approach is based on individual segment's angular velocity integration during motion and orientation correction with respect to gravity and the Earth's magnetic field. The accelerometer is used as an inclinometer by comparing measured acceleration vector to the vector of gravity in order to determine the intermediate angle of inclination. For successful implementation of the magnetometer into the algorithm it is assumed that magnetic field in space is locally constant (constant direction and length) and non-parallel to gravity. Considering this assumption, the angle of rotation around the gravity vector can be calculated by comparing the measured magnetic vector with the initial vector of the magnetic field.

The approach often results in a drift during long-term dynamical movement due to gyroscope drift, errors introduced by separation of gravity and dynamic acceleration and changes of the magnetic field. To compensate for this effect, resetting during standstill was introduced. In order to further reduce the drift without the need for standstill, a kinematic model of the human body was incorporated into the sensory fusion algorithm.

To determine joint angles that describe relative position between segments, an error quaternion between two adjacent segments is calculated and presented with rotation angles as shown in Figure 3.

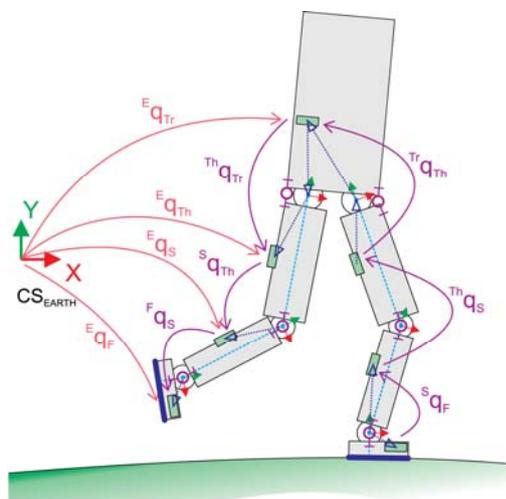


Figure 3 Calculation of body kinematic parameters.

Other parameters which can also be obtained with the presented wearable system (e.g. step duration, gait frequency, acceleration of centres of mass, etc.) exceed the scope of this paper.

4 MOVEMENT CLASSIFICATION AND PHASE DETECTION

In literature, authors presented segmentations for different movement types: ground-level walking [10], walking over obstacles [11], stair climbing [12], sit-to-stand and stand-to-sit [13]. A threshold-based decision tree was developed for

performing recognition of ground level walking with gait phases, stair ascent and descent and stand-to-sit and sit-to-stand manoeuvre based on wearable sensory data. In Figure 4 a conceptual state diagram with all the possible transitions between detected manoeuvres and gait phases is presented.

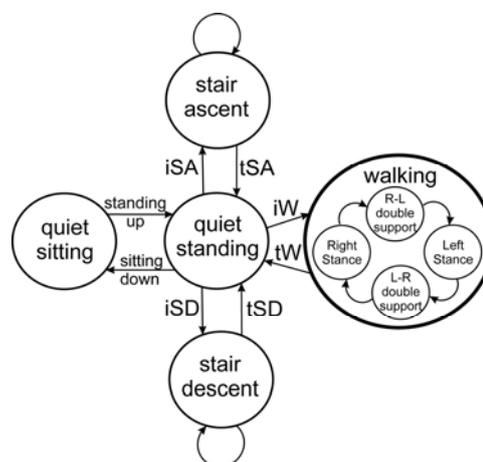


Figure 4 A conceptual diagram of the detected set of movement types and transitions; iW - initiation of walking; iSA - initiation of stair ascent; iSD - initiation of stair descent; tW - termination of walking; tSA - termination of stair ascent; tSD - termination of stair descent.

The decision tree was tuned off-line to perform movement identification and intention detection as well as phase segmentation within recognized manoeuvres. The algorithm was tested with healthy subjects online, in real-time. Results show that while this approach is successful for recognition of movement types, it is sensitive to unexpected behaviour. For this reason, a more robust, state-machine architecture for the algorithm was built. This approach allows transitions between states to occur only when they satisfy a number of specific conditions. The transition threshold values may be defined manually while specific conditions and the general order of magnitude of these values remain unchanged. On the other hand, this makes the classification and segmentation machine less tailored to a specific wearer of the sensory system.

As an alternative, a fuzzy logic approach was utilized. The approach resulted in creating soft (fuzzy) transition rules for the state-machine with the use of fuzzy clustering method. The method outputs probability with which the fused sensory data best fits a given signal cluster [14]. The centres of each cluster were taught before-hand, offline. Probability is calculated as the distance to each of the centres.

5 TRACKING OF BALANCE DESCRIPTORS

The human body is a multi-body system, supported by only one or two relatively small segments, which results in a fairly small supportive polygon. This polygon is a convex hull that includes all points of contact between the body and the outside world. Human posture is defined by reciprocal relationships of human-body segments and their orientation

with respect to the Earth's inertial frame. One of the goals of the CYBERLEGS project is to track estimates of particular balance descriptors using only data derived from wearable sensors.

Two balance descriptors were chosen for tracking: the whole body Centre Of Mass (COM) and the Zero-Moment Point (ZMP - a point on the ground where the horizontal components of the resultant moment are equal to zero).

The COM is determined from kinematic parameters of the user and warrants knowledge of mass and inertial parameters of particular segments of the human body. Thus, the segmental COM positions are estimated according to segment orientations and statistically-determined human body anthropometrical data [15].

The ZMP can either be measured (when the resultant horizontal moment is balanced) or estimated from a combination of kinematic and inertial data. The derivation of ZMP through forward kinematics is based on moment equilibrium in the ground plane, following Equation (1) [16]:

$$M_R = [0 \ 0 \ 0]^T = \sum_{i=1}^n (\vec{r}_{COM,i} \times (m_i \cdot \vec{a}_i) + I_i \cdot \dot{\omega}_i + \omega_i \times (I_i \cdot \omega_i)) - \vec{r}_{COM,i} \times m_{body} \cdot \vec{g} + \left(\sum_{i=1}^n m_i \cdot \vec{a}_i - m_{body} \cdot \vec{g} \right) \times \vec{r}_{ZMP} \quad (1)$$

The index i denotes the i -th segment, m_i mass, ω_i and $\dot{\omega}_i$ angular velocity and acceleration, respectively. I_i is the segment's inertia matrix, while $\vec{r}_{COM,i}$ and \vec{r}_{ZMP} denote vectors of segment COMs (former) and ZMP (latter).

The proposed algorithms build upon assumptions, which introduce some limitations to the accuracy of the estimates. They assume that all body segments are non-compliant, the ground is rigid and level and no sliding or slipping occurs. Furthermore, all anthropometric and kinematic information is assumed known [16].

6 EXPERIMENTAL EVALUATION

Experimental data was collected in experiments with multiple subjects performing walking, stair climbing and sit-to-stand manoeuvres. Using NDI Optotrak 3D optical position measurement system, spatial positions of active infrared markers, attached to bony landmarks of the subject's body, were measured. Linear and angular velocities as well as accelerations were either derived from 3D position data (only dynamic acceleration) or directly measured by the inertial measurement units, placed on the subject's body segments. In the latter case, each IMU was equipped with three IR markers, relaying information on placement position and orientation to the investigators. During the walking manoeuvre, data from sensorized

insoles was collected in order to determine foot-ground contact times and estimates of vertical ground reaction force and centre of pressure. Two IR markers were positioned at the side of the shoe soles to estimate (with knowledge of the sole outline with respect to these markers) the contact hull of support for each foot. Sensor placement is shown in Figure 5. Experimental validation of the wearable system for measuring movement kinematics of healthy subjects was performed by comparison of orientations obtained from IMUs with data acquired by the Optotrak system.



Figure 5 Experimental evaluation: reference system (Optotrak Certus and force plates) and wearable sensors (IMUs and insoles).

6.1 DETECTION OF WALKING PHASES

Initial testing and validation was performed for the walking manoeuvre. Figure 6 shows results of online recognition for the walking manoeuvre with gait phases using a hard-coded state-machine and fuzzy clustering method.

Results of preliminary comparison show more robust performance of the state-machine algorithm, while fuzzy logic offers easier adjustments to an individual subject. Currently, the fuzzy clustering method requires the clusters to be defined offline, based on a learning set, before applying them to the classification and segmentation engine.

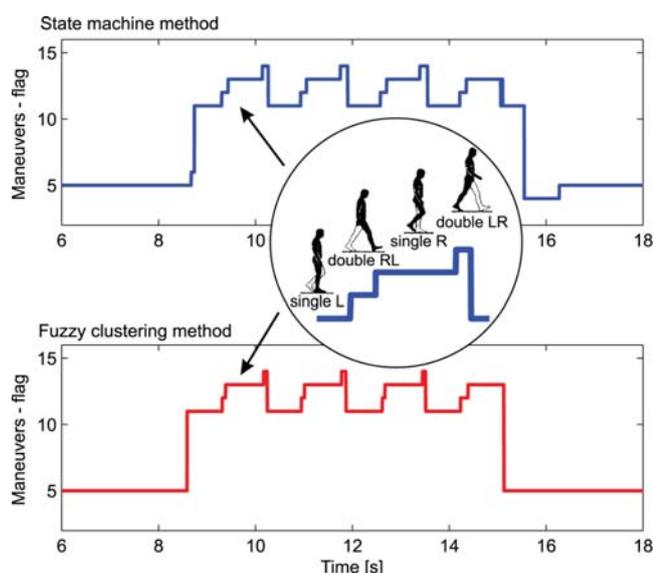


Figure 6 Results of manoeuvres and gait phase detection, comparing two different approaches (above: state machine, below: fuzzy logic clustering).

6.2 MOTION KINEMATICS TRACKING IN STANDING-UP

Experimental validation of the wearable system for measuring movement kinematics of healthy subjects was performed by comparison of joint angles obtained from IMU with data acquired by a reference measurement system (Optotrak). Joint angles were derived from measured positions of markers placed on bony landmarks of the subject's body.

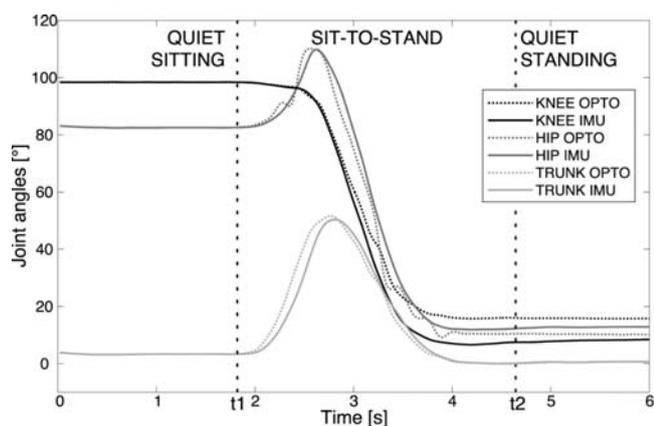


Figure 7 Typical knee and hip angles and trunk inclination during standing up, obtained with reference system (OPTO) and wearable system (IMU). Times t_1 and t_2 denote the start and finish of standing up, respectively.

Five IMUs were placed on body segments (shanks, thighs and trunk) along with IR active markers for position reference. Joint angles were obtained with decomposition of error quaternions between two segments. The algorithm was evaluated with experiments in standing-up and sitting-down of a healthy subject. Experimental protocol involved

several repetitions of sit-to-stand and stand-to-sit motion during three-minute-long measurements. Typical joint angle trajectories (knee and hip angles and trunk inclination) derived from Optotrak- or IMU-based data, respectively, are presented in Figure 7.

Statistical comparison of joint angles (knee and hip joint angle and trunk inclination angle) is presented in Figure 8 with boxplots of absolute error between angles obtained from the Optotrak system and those from the wearable system. Results show that the median value of absolute error of the wearable system is below 2° and is as such appropriate for measuring kinematics parameters.

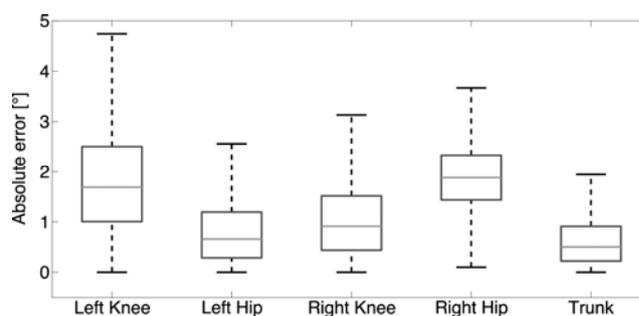


Figure 8 Boxplot of absolute error between joint angles calculated from IMU data and angles calculated from Optotrak data.

6.3 BALANCE DESCRIPTORS

For the stair climbing manoeuvre, the trajectories of the ZMP point, measured by the reference force sensors implemented in ground and stairs, were compared to the ZMP trajectory estimated from position-derived data. Seven healthy subjects performed multiple barefoot walks up the stairs. Results showed a combined RMSE error of 54 mm, thus proving the concept of ZMP assessment by a kinematics-based approach for a human subject in motion [17].

An algorithm that combines data provided from position sensors with data, collected by wearable sensors, was used to yield sample-based balance descriptor estimates with respect to a desired coordinate frame origin. Estimates of COM and ZMP trajectories from combined sensory data for a 4-step walking manoeuvre are shown in Figure 9.

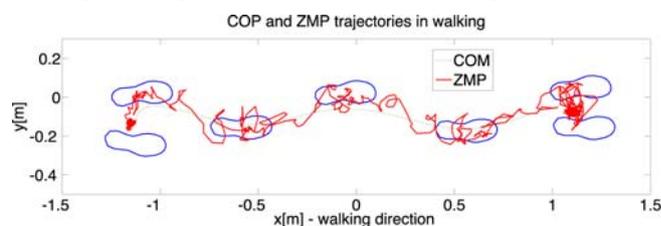


Figure 9 ZMP and COM trajectory with respect to the feet support for a 4-step walking manoeuvre with quiet standing at the beginning and the end of walk.

Preliminary results indicate that balance descriptors estimated from a combination of sensors are equally descriptive as those derived solely from position data. In addition, by combining subject kinematics with data from

the insoles, we are able to track the instantaneous base of support. As a result, the ZMP stability margin (minimal distance from the edge of the support polygon) can be assessed on-line.

Figure 10 shows one slow walk with 4 steps, starting and ending while the subject is in quiet standing. Stability is quantified as the distance from the closest point in which the resultant moment on the body, acting in the plane of the ground support, can be balanced by a resultant force applied to the ground by the subject. Depending on whether this point lies within the base of support or outside of it, the point is termed ZMP or Fictitious ZMP, respectively. The distance from the closest edge of the support polygon is positive when the resultant force should be applied outside and negative when inside the instantaneous base of support.

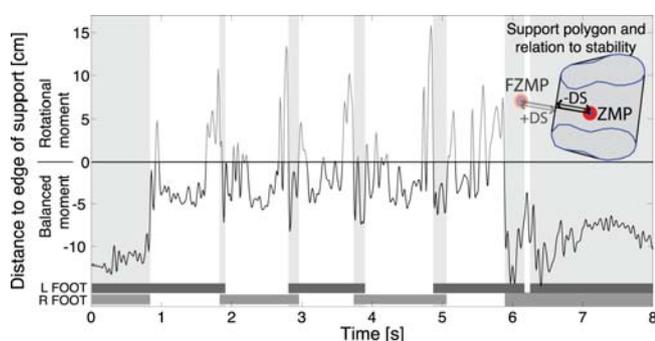


Figure 10 Assessed stability in terms of minimal distance of the ZMP from the edge of the support polygon. Negative values denote situations where the ZMP lies within the base of support while positive values convey the distance of the FZMP (a point outside of the support polygon) to the edge of the base of support. The bottom lines show contact times of the left and right foot, respectively. Grey patches over the entire figure mark double support instances, while white space marks single stance of a foot.

Preliminary investigation suggests that most of the unbalanced moment (tipping about the edge of support) is present at times of beginning and end of single support - that is right after toe off and right before heel strike of the swing leg. This conforms to the idea that humans take advantage of physical dynamics of the body during steady-state gait with regard to swing foot placement [18] and react to unbalanced moment by moving parts of the body that are not in contact with the environment and thus maximize stability [19].

7 CONCLUSIONS

This paper presents the sensory system for robotic orthoprosthesis which is based on wearable sensors incorporating the inertial measurements units and sensorized insoles worn by the subject. Data from the wearable system is fused by sensory fusion algorithms to assess the human body motion kinematics, identify motion manoeuvre and its phases, and track stability descriptors.

The algorithm for motion kinematics assessment is based on Extended Kalman filtering and fuses the integral of angular velocity with the estimate of linear acceleration, based on kinematic model of the human body. Experimental validation of the algorithm was accomplished on healthy subject performing sit-to-stand manoeuvre. Comparison of joint angles, assessed by wearable system, with those obtained from reference position measurements results in absolute error median smaller than 2° . The results confirm the suitability of wearable system for joint angles assessment in dynamic situations. Additionally, performance of detection of manoeuvres and gait phases was tested online, exploiting both measured and estimated data. Two different algorithms were developed and evaluated. A decision tree with threshold based rules generally performs more robust detection, while the fuzzy clustering method is more convenient when the acquisition of training data sets is possible. Furthermore, fuzzy clusters can be tailored to each subject. Making use of a statistical anthropometric model, segmental centres of mass and the estimated ZMP position, the support polygon was assessed and the ZMP stability margin quantified for ground level walking. Results suggest that most of the unbalanced moment (tipping over the edge of support) is present at instants of beginning and end of single support phase. This suggests that humans take advantage of physical dynamics of the body during steady-state gait, thus exhibiting globally stable gait.

In future developments, the presented contributions will be combined to form a unique sensory system for motion parameters tracking based on wearable sensory data only.

ACKNOWLEDGEMENTS

The CYBERLEGs project is financed within the scope of the 7th framework of the European Community (FP7/2007-2013) under grant number 287894. This study was partially supported by the Slovenian Research Agency (ARRS) under contract number 1000-11-310147.

REFERENCES

- [1] K. Hagberg and R. Brånemark, Consequences of non-vascular trans-femoral amputation: a survey of quality of life, prosthetic use and problems. *Prosthetics and orthotics international*, vol. 25, no. 3, pp. 186-194, 2001.
- [2] W. C. Miller, M. Speechley, and B. Deathe, The prevalence and risk factors of falling and fear of falling among lower extremity amputees. *Archives of physical medicine and rehabilitation*, vol. 82, no. 8, pp. 1031-1037, 2001.
- [3] C. Kirtley, *Clinical gait analysis: theory and practice*. Churchill Livingstone, 2006.
- [4] P. Bonato, Wearable sensors/systems and their impact on biomedical engineering. *IEEE Engineering in Medicine and Biology Magazine*, vol. 22, no. 3, pp. 18-20, 2003.

- [5] D. Roetenberg, H. Luinge, and P. Slycke, *Xsens MVN: full 6DOF human motion tracking using miniature inertial sensors*. Xsens Motion Technologies BV, Tech. Rep, 2009.
- [6] A. M. Sabatini, Quaternion-based extended Kalman filter for determining orientation by inertial and magnetic sensing. *IEEE transactions on bio-medical engineering*, vol. 53, no. 7, pp. 1346-56, Jul. 2006.
- [7] X. Yun and E. R. Bachmann, Design, implementation, and experimental results of a quaternion-based Kalman filter for human body motion tracking. *IEEE Transactions on Robotics*, vol. 22, no. 6, pp. 1216-1227, 2006.
- [8] S. M. M. De Rossi et al., Development of an in-shoe pressure-sensitive device for gait analysis. *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2011, pp. 5637-5640.
- [9] D. Novak, P. Rebersek, T. Beravs, J. Podobnik, and M. Munih, Development and validation of a wearable inertial measurement system for use with lower limb exoskeletons. *Proceedings of IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics*, 2011, pp. 212-217.
- [10] M. W. Whittle, *Gait analysis: an introduction*. Butterworth-Heinemann, 1996.
- [11] T. Li, M. Ceccarelli, T. Essomba, M. A. Laribi, and S. Zeghloul, Analysis of human biped obstacle overcoming by motion capture system. *Presented at the international Workshop on Robotics in Alpe--Adria--Danube Region 2012 (RAAD 2012)*, 2012, pp. 85-92.
- [12] B. J. McFadyen and D. A. Winter, An integrated biomechanical analysis of normal stair ascent and descent. *Journal of biomechanics*, vol. 21, no. 9, pp. 733-744, 1988.
- [13] A. Kralj, R. J. Jaeger, and M. Munih, Analysis of standing up and sitting down in humans: definitions and normative data presentation. *Journal of biomechanics*, vol. 23, no. 11, pp. 1123-1138, 1990.
- [14] S. Gagula-Palalic, *Fuzzy Clustering Models and Algorithms for Pattern Recognition*. International University of Sarajevo, 2008.
- [15] P. De Leva, Adjustments to Zatsiorsky-Seluyanov's segment inertia parameters. *Journal of biomechanics*, vol. 29, no. 9, pp. 1223-1230, 1996.
- [16] M. H. P. Dekker, *Zero-Moment Point Method for Stable Biped Walking*. Internship report DCT Nr.: 2009.072, 2009.
- [17] L. Ambrozic, *Analiza biomehanskih parametrov pri hoji po stopnicah*. Bachelor thesis, University of Ljubljana, 2012.
- [18] J. S. Matthis and B. R. Fajen, Humans exploit the biomechanics of bipedal gait during visually guided walking over complex terrain. *Proceedings of the Royal Society B: Biological Sciences*, vol. 280, no. 1762, 2013.
- [19] T. Robert, B. C. Bennett, S. D. Russell, C. A. Zirker, and M. F. Abel, Angular momentum synergies during walking. *Experimental brain research*, vol. 197, no. 2, pp. 185-197, 2009.

DETECTION OF NEEDLE INSERTION POINT FOR PHLEBOTOMY IN HUMAN FOREARMS

Thiusius Rajeeth Savarimuthu

Anders Stengaard Sørensen

The Maersk Mc-Kinney Moller Institute
University of Southern Denmark

ABSTRACT

The number of blood samples drawn in modern health care systems is increasing rapidly and they are now bottlenecks and causing personnel strain at many hospitals. One solution to this problem is automated phlebotomy using robots. In this paper, we present a new method for selection of a needle insertion point, which can be used as part of automatic phlebotomy using robots. The algorithm is derived from a study of clinical needle insertion on real patients, by professional phlebotomists, and is verified against 139 clinical needle insertions. The method relies on NIR optical images of the forearm, and is thus faster, easier and more hygienic than previously proposed methods using mechanical or sonographic imaging.

Keywords: Blood sampling, Needle insertion, Image processing

1 INTRODUCTION

Venepuncture is one of the most routinely performed procedures at all in modern medicine. Venepuncture gives intravenous access for the purpose of intravenous therapy or obtaining a venous blood sample for blood analysis. The number of venous blood samples produced in the health care systems today is enormous and for some countries the production of blood samples is becoming a bottleneck in the system. For instance in Denmark, the number of venous blood drawn is estimated to be 16 million samples in 2010. For 5.7 million citizens, that is over 2 samples in average for each citizen per year. The number of blood samples has been growing a steady 6-7% per year since 1987. This growth has largely been balanced by an increased productivity, due to automation of the blood sample processing. However, an automation of the venepuncture itself has not been developed and due to the high demand this has become relevant in the recent years. Various methods to detect blood vessels have been studied for many years. Most predominant in the area

of retinal image processing where the contribution from [3] has been the foundation for many other papers [1, 9, 4, 12]. From being used for vessels detection in retinal images some of the methods got transferred to detection of vessels like patterns in other areas such as hand vein pattern extraction [14]. In [11] vessel pattern of fingers are extracted for the use of personal identification.

Lately products have been introduced to aid the phlebotomist to find the blood vessels during the venepuncture procedure. One of these products — VeinViewer — is presented in [13, 5]. Also methods towards automatic venepuncture has been presented in [15] and lately in [2].

In this paper we present a new method for detection a needle insertion point for venepuncture in the human forearm.

2 METHODS

This section describes the equipment used for NIR image collection. The hardware used to collect data for this paper is shown in Figure 1. It consist of a NIR camera, marked 'A' in Figure 1, constructed using the description given in [7]. The camera used in this setup is a Stingray F-033B with PXC815B(KP) lens.

Contact author:

Thiusius R. Savarimuthu¹, Anders S. Sørensen².

Niels Bohrs Allé 1, 5230 Odense, Denmark

¹ trs@mmmi.sdu.dk, ² anss@mmmi.sdu.dk

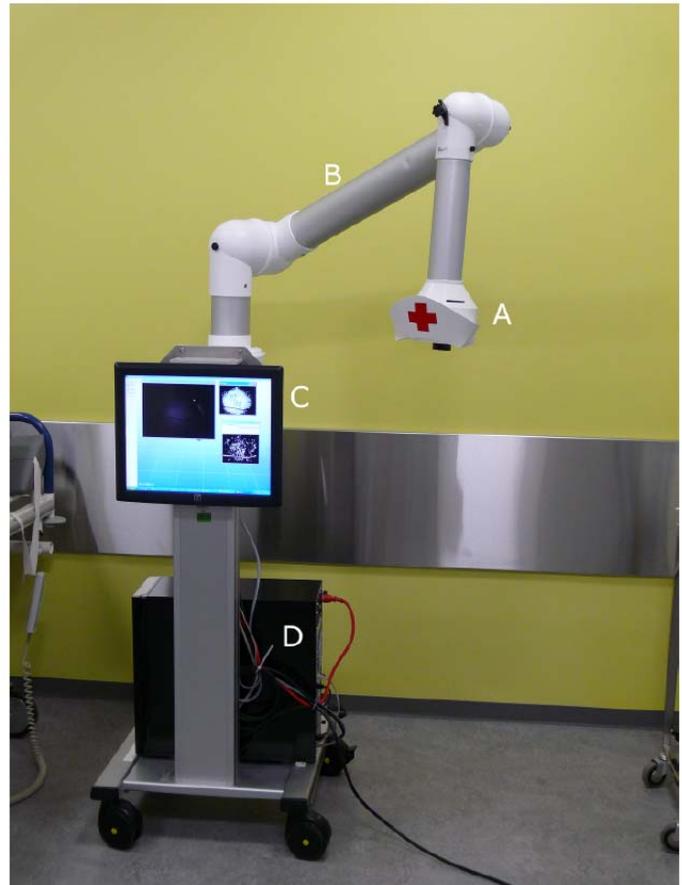
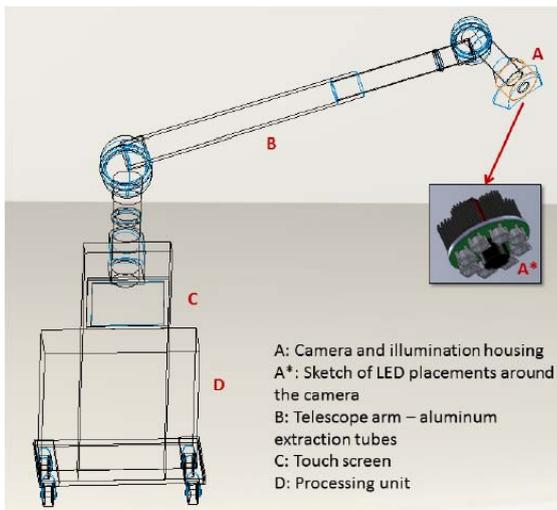


Figure 1 Picture of the camera system. On the left is a sketch of the setup and on the right is the final implementation of the setup. A marks the camera and illumination mount, B marks is a reconfigurable mount that can be adjust by the user, C marks the touch screen witch is the user interface and D marks the processing unit.

In front of that is mounted a long-pass filter LP830-40. Finally the polarizing filter, PI035, is mounted in front of the long-pass filter. The illumination source is constructed using 8 high power led of the type ACL01-SC-III-005-C01-L from Aculed. They are mounted in a circular pattern around the camera lens to approximate a uniform illumination of the arm. Figure 2(A) shows the Quantum Efficiency-Wavelength of the filter and (B) shows the spectral distribution of the leds. The camera device is mounted on a cart with a reconfigurable arm, marked 'B' in Figure 1. The cart together with the arm has 5 degree of freedom that can be used by the phlebotomist to adjust the camera device to point at the arm of the patient. The user interface, marked 'C', and the data collecting and processing unit, marked 'D', is mounted on the cart. Figure 3 shows how the hardware is placed in respect to the patient and the phlebotomist. The next subsection describes the type of data collected and how it is collected at the hospital.

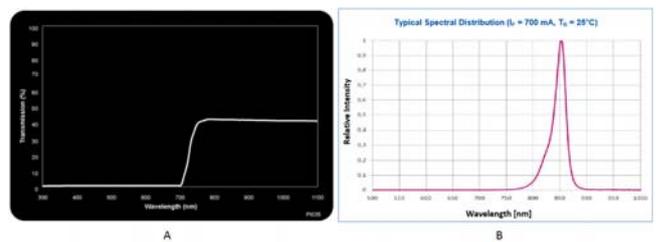


Figure 2 Quantum Efficiency-Wavelength of the LP830-40 filter - Source: <http://www.midopt.com/lp830.html> (left, A); spectral distribution of the leds used - Source: http://www.excelitas.com/downloads/dts_aculedvhl_ir.pdf. (right, B)



Figure 3 Picture of the camera system being used by a phlebotomist.

2.1 DATA COLLECTION

The data used in this paper is collected at Lillebaelt Hospital in Vejle, Denmark. It is collected over a period of approximately 9 months, using the hardware described above. The hardware is used by the phlebotomist during the venepuncture procedure. The data used in this paper consist of image pairs collected for each patient contact using the following procedure. The first image in an image pair is called “before” and is acquired before the phlebotomist inserts the needle in the patients arm. The second image in the image pair is acquired right after the needle is inserted in the patients arm and this image is called “after”. Figure 5 shows an example of an image pair. Subimage 5(a) is the “before” picture of this image pair and subimage 5(b) is the “after” picture of this particular pair. Each image pair belongs to the same patient and is of the same arm of that patient.

During the data collection period of 9 months a total of 167 patient contacts were registered using the data collection hardware.

Figure 4 shows four plots of the distribution of the 167 patients’ height (4(a)), age (4(b)), weight (4(c)) and gender (4(d)). As seen in the plots in figure 4 the patient population has an age span between 20 to 80 years and include more women (62%) than men (38%). Apart from the age distribution the patient population sampled in this test is statistically comparable with the normal population at the above mentioned site [6].

From the 167 patient contacts 28 image pairs were removed from the data set due to data collection errors such as bad illumination settings, only one image in the image pair etc. The remaining 139 image pairs are used in the data processing steps described in the next subsection.

2.2 DATA PROCESSING

The goal of our study of the image pairs is to devise an algorithm that will enable a computer to estimate a good needle insertion point. We do not have an objective way to define an ‘optimal’ insertion point for comparison, but we can extract the phlebotomists original needle insertion point (ONP) from the “after” picture. We then define a good needle insertion point to be in the same vessel. The result of the data processing algorithm is an estimate of a needle insertion point (ENP).

The input for the algorithm is the “before” picture together with the two pixel coordinates for the points marking each side of the arm by the cubital fossa anterior (elbow pit). This additional input is currently detected manually, but will be detected automatically in later versions.

The output from the algorithm is an estimated needle insertion point (ENP), which is found using the following steps.

The first step of the algorithm is to find the region of interest for this application. The region of interest is defined to be the area of the arm in between the two points marking the side of the arm by the cubital fossa anterior. More precisely the region of interest for general phlebotomy patients can be narrowed down to the area around the line connecting the two points. In this paper we choose to represent the region of interest by a circle with centre in the halfway mark between the two points. The radius of the circle is 30% of the distance between the two points. Figure 6 shows the input image to the left and the input image after the region of interest is calculated to the right.

After the region of interest is calculated the matched filtering algorithm given in [8, 7], it is applied with $\sigma = 5$; $L = 9$; $\theta = 15$. This gives 12 kernels in the kernel bank. To optimize runtime the kernel can be applied only to the calculated region of interest. The result of the matched filtering is then threshold so that the highest 10% of the matched filter output in the region of interest remains after the threshold. This can be done by calculating the cumulative histogram of the matched filter outputs of all the pixels in the region of interest and then use the bin value of the 90th percentile as the threshold value. The thresholding algorithm sets all matched filter outputs below the threshold to zero and values above the threshold remains the same.

The last step before estimating a needle insertion point is to apply a modified ultimate erosion method on the remaining pixels after the threshold. The reason for applying this last step is to add more value to larger and thicker vessel like structures compared with smaller and thinner structures. Ultimate erosion is a process of peeling of the contour of an blob until only the barycenter of the blob is remaining. This method is well know morphologic method in image processing and is described in [10]. In this paper the concept ultimate erosion is used it to accumulate the response from the matched filtering process. The accumulation is performed within a 8-connected blob.

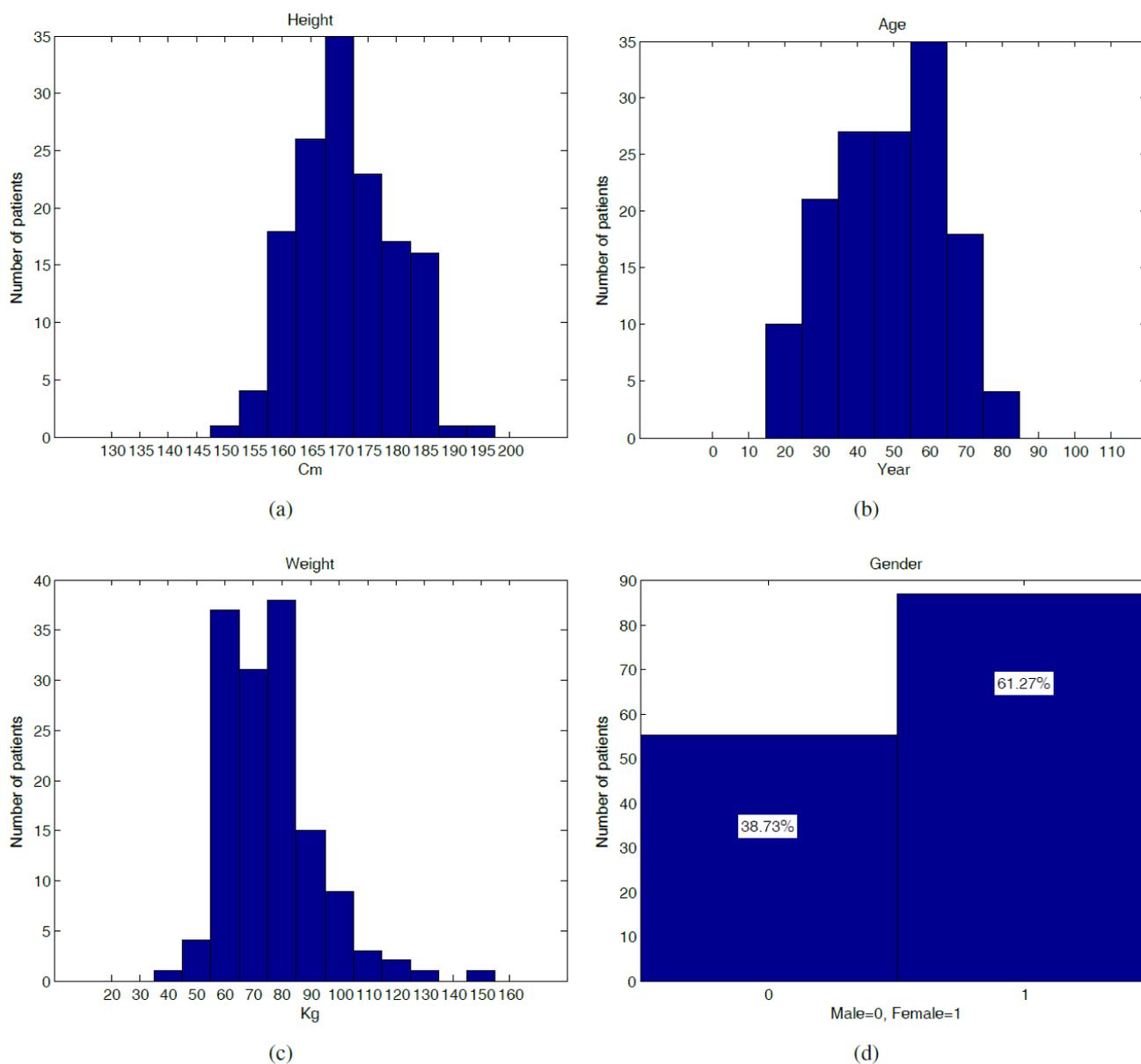


Figure 4 The four plots show the distribution at height (a), age (b), weight (c) and gender (d) of the 167 patients.

Each time a layer of the contour would have been peeled off a blob, the filter results of the “removed” pixels are added to the “remaining” contour pixels. The added value to a given “remaining” contour pixels is the sum of all its 8-connected neighbours that would have been removed. Figure 7 shows an illustration of how the modified ultimate erosion method works. The new value of the pixel(2,2), the red 6 in B, is calculated by summarizing the values of the 8-connected pixels in the boundary with its initial value. That is $Pixel(2,2) = 1+(1+1+1+1+1) = 6$ and in the same way $Pixel(3,2)=2+(1+1+1)=5$.

Finally for $Pixel(3,3) = 4+(6+5+6+5+5+6+5+6) = 48$

The final step is to find the pixels with the highest value inside the region of interest. If there is more than one pixel with the highest value, the pixel closet to the centre of the region of interest is selected. This is a theoretical possibility; in practice no one of the 139 images had more than one pixel with the highest value. Figure 8(a) shows the final outcome of the data processing presented in this section. The red point marks the estimated needle insertion point and the green point marks the original needle insertion point.

The next section contains the results of the described data processing applied to all 139 images.

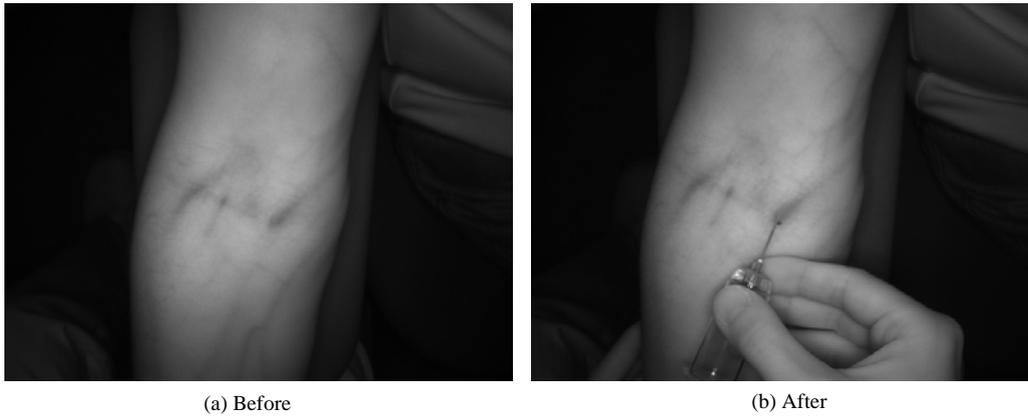
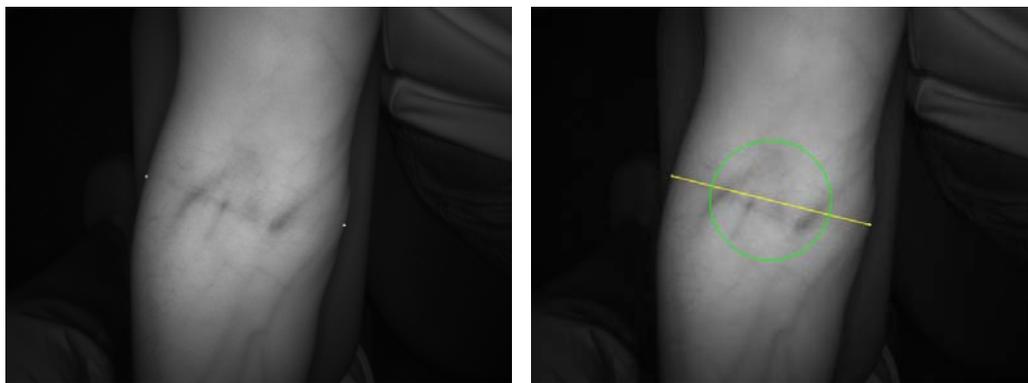


Figure 5 Patient arm before (a) and after (b) the needle insertion.



(a) Input - original image with points marking each side of the arm by the cubital fossa anterior. (b) Region of interest marked by the green circle with centre at the halfway point between the two input points.

Figure 6 The input picture (a) and the result of the region of interest selection (b).

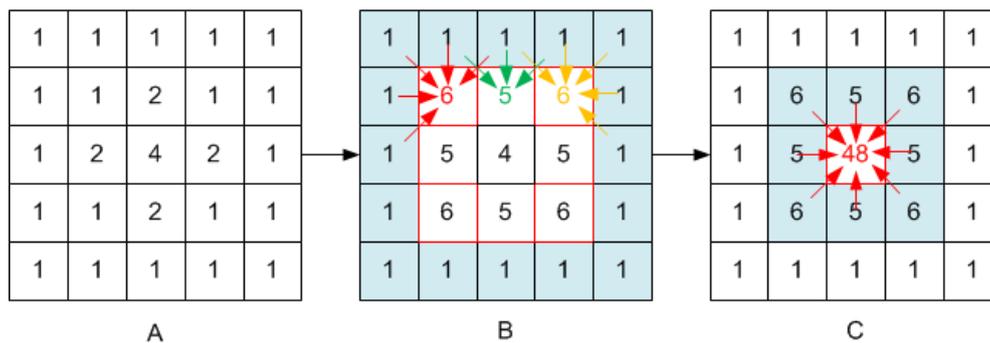


Figure 7 Modified ultimate erosion algorithm where all boundary values are added toward the centre of a blob. A is the initial blob and B is the new value of the blob after accumulating the values of the outer most layer of the blob. Here marked with light blue colour. The new value of the pixel(2,2), the red 6, is calculated by summarizing the values of the 8-connected pixels in the boundary with its initial value. C is the resulting blob after accumulating all the layers into the centre of the blob.

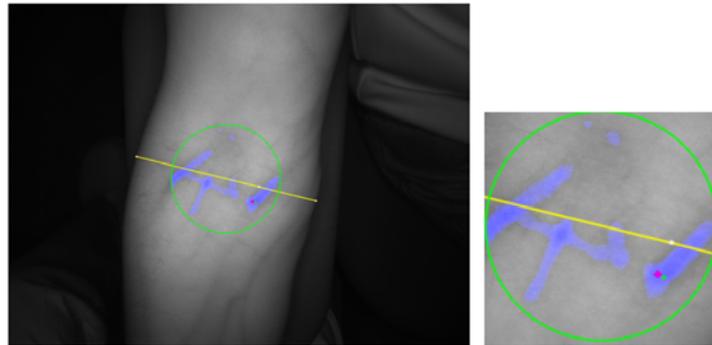


Figure 8 The final result of the data processing (a). A close up of the region of interest is shown in (b) where green point marks the ONP and the red point marks ENP.

3 RESULTS

By applying the data processing algorithm to all 139 image pairs results for the entire set, using 30% of the distance between the two input points as radius for the region of interest, in a detection rate of 90% of ONPs in the 139 images. Figure 9 shows an example of an image where the ONP is outside the calculated region of interest and hence the ONP is not detected.

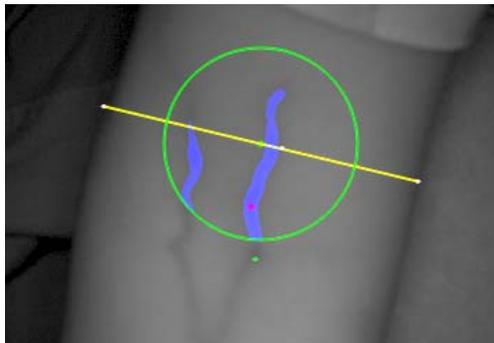


Figure 9 An example of an image, where ONP is outside of the calculated region of interest.

The size of radius is a trade off between detection rate of ONP on one side and computation time, edge artefacts and noise in data on the other side. It is clear that by increasing the size of radius for the image in Figure 9 the ONP would have been detected, but if the radius increased to much the lower edge of the patients clothing would become a part of the region of interest and could be a potential source of noise.

Modelling the region of interest with a circle is a very simple solution. It is clear that other ways of modelling the region of interest would ensure more of the ONPs to be detected. Such an alternative model could be an ellipse. In 87 of the 139 images or the ENP and ONP are in the same vessel. In percentage this equals 62,5% and Figure 8(a) is a clear example of this. But the remaining 27,5% of the images includes ENPs that are not in the same vessels as the ONPs but are in a vessel that could be a potential good vessel for phlebotomy. Figure 10 shows an example where ONP and ENP is in two different vessels, but it cannot be ruled out that the ENP is a suitable place to collect a blood sample. This is an interesting problem, which will be investigated in the future.

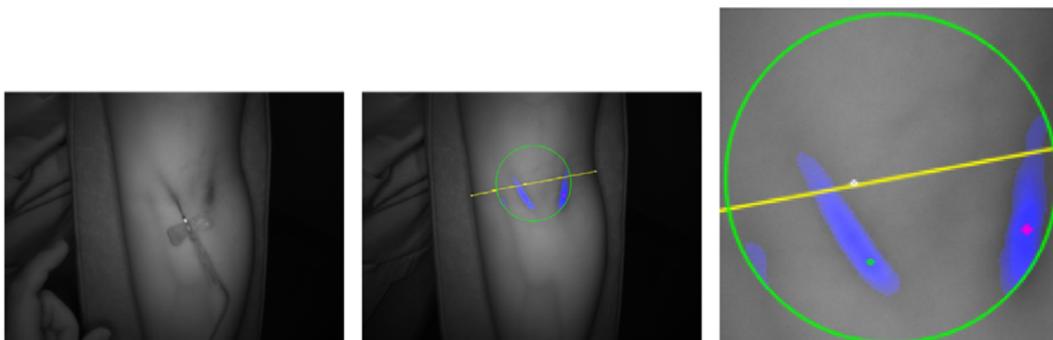


Figure 10 The image 10(a) shows which vessel is used for ONP in this particular case and image 10(b) shows the result of proposed algorithm. Image 10(c) shows a close up of the result and here it can be seen that the proposed algorithm elects another vessel for ENP (red mark) than the vessel used for ONP.

4 CONCLUSIONS

A new method for estimating needle insertion points for phlebotomy has been presented. The method shows that it is possible to detect 90% of the original needle insertion points using the proposed method. Furthermore the estimated needle insertion points lies within the same vessel structure in 62,5% of the cases. The results are based on data collected at Lillebaelt Hospital in Vejle, Denmark. Further development and test of the method is already scheduled and one of the more interesting questions to test is whether the estimated needle insertion points can be used for phlebotomy in the clinic.

REFERENCES

- [1] Al-Rawi M., Qutaishat M. and Arrar M., An improved matched filter for blood vessel detection of digital retinal images, *Computers in biology and medicine*, Vol. 37, No. 2, pp. 262 – 267, 2007.
- [2] Brewer R. and Salisbury J., Visual vein-finding for robotic IV insertion, (*ICRA*), *2010 IEEE International Conference on Robotics and Automation*, Anchorage, Alaska, May 3 - 8, 2010, pp. 4597–4602.
- [3] Chaudhuri S., Chatterjee S., Katz N., Nelson M. and Goldbaum M., Detection of blood vessels in retinal images using two-dimensional matched filters, *IEEE Transactions on Medical Imaging*, Vol. 8, No. 3, pp. 263–269, 1989.
- [4] Fang B., Hsu W. and Lee M., On the Detection of Retinal Vessels in Fundus Images, *Computer Science (CS)*, 2003.
- [5] Paquit V., Price J., Mériaudeau F., Tobin K., and Ferrell Th. L., Combining near-infrared illuminants to optimize venous imaging, *Proceeding of Medical Imaging 2007: Visualization and Image-Guided Procedures*, San Diego, CA, USA, 2007.
- [6] T. Savarimuthu, Towards automatic bloodsampling, Syddansk Universitet, Det Tekniske Fakultet, 2011.
- [7] Savarimuthu T. and Kjaer-Nielsen A., Real-time medical video processing, enabled by hardware accelerated correlations, *Journal of Real-Time Image Processing*, Vol. 6, No. 3, pp. 187-197, 2011.
- [8] Savarimuthu T. and Sorensen A., Detection of Vessels in Human Forearms Using 2D Matched Filtering, *The Signal and Image Processing 2008 Conference*, Kailua-Kona, Hawaii, USA pp. 294-299, 2008.
- [9] Sofka M. and Stewart C. V., Retinal Vessel Extraction Using Multiscale Matched Filters, Confidence and Edge Measures, *Technical Report #05-20*, 2005.
- [10] M. Sonka M., V. Hlavac V. and Boyle R., *Image Processing, Analysis, and Machine Vision*, 2nd edition, PWS - an Imprint of Brooke and Cole Publishing, 1998,
- [11] Vlachos M. and Dermatas E., Vein segmentation in infrared images using compound enhancing and crisp clustering, *Proceedings of the sixth international conference on Computer vision systems*. Springer-Verlag, pp. 393–402, 1998.
- [12] F. Zana F. and Klein J., Segmentation of vessellike patterns using mathematical morphology and curvature evaluation, *IEEE Transactions on Image Processing*, Vol. 10, No. 7, pp. 1010–1019, 2001.
- [13] Zeman H., Lovhoiden G., and Vrancken C., The clinical evaluation of vein contrast enhancement, *Engineering in Medicine and Biology Society, 2004. IEMBS '04. 26th Annual International Conference of the IEEE*, San Francisco, CA, USA, Vol. 1, pp. 1203-1206, 2004.
- [14] Zhao S. and Wang Yiding and Wang Yunhong., Extracting Hand Vein Patterns from Low-Quality Images: A New Biometric Technique Using Low-Cost Devices, *Fourth International Conference on Image and Graphics, ICIG 2007*. Sichuan, China, pp. 667–671, 2007.
- [15] Zivanovic A. and Davies B., A robotic system for blood sampling, *IEEE Transactions on Information Technology in Biomedicine*, Vol. 4, No. 1, pp. 8-14, 2000.

TEMPLATE FOR PREPARING PAPERS FOR PUBLISHING IN INTERNATIONAL JOURNAL OF MECHANICS AND CONTROL

Author1* Author2**

* affiliation Author1

** affiliation Author2

ABSTRACT

This is a brief guide to prepare papers in a better style for publishing in International Journal of Mechanics and Control (JoMaC). It gives details of the preferred style in a template format to ease paper presentation. The abstract must be able to indicate the principal authors' contribution to the argument containing the chosen method and the obtained results. (max 200 words)

Keywords: keywords list (max 5 words)

1 TITLE OF SECTION (E.G. INTRODUCTION)

This sample article is to show you how to prepare papers in a standard style for publishing in International Journal of Mechanics and Control.

It offers you a template for paper layout, and describes points you should notice before you submit your papers.

2 PREPARATION OF PAPERS

2.1 SUBMISSION OF PAPERS

The papers should be submitted in the form of an electronic document, either in Microsoft Word format (Word'97 version or earlier).

In addition to the electronic version a hardcopy of the complete paper including diagrams with annotations must be supplied. The final format of the papers will be A4 page size with a two column layout. The text will be Times New Roman font size 10.

2.2 DETAILS OF PAPER LAYOUT

2.2.1 Style of Writing

The language is English and with UK/European spelling. The papers should be written in the third person. Related work conducted elsewhere may be criticised but not the individuals conducting the work. The paper should be comprehensible both to specialists in the appropriate field and to those with a general understanding of the subject. Company names or advertising, direct or indirect, is not permitted and product names will only be included at the discretion of the editor. Abbreviations should be spelt out in full the first time they appear and their abbreviated form included in brackets immediately after. Words used in a special context should appear in inverted single quotation mark the first time they appear. Papers are accepted also on the basis that they may be edited for style and language.

2.2.2 Paper length

Paper length is free, but should normally not exceed 10000 words and twenty illustrations.

2.2.3 Diagrams and figures

Figures and Tables will either be entered in one column or two columns and should be 80 mm or 160 mm wide respectively. A minimum line width of 1 point is required at actual size. Captions and annotations should be in 10 point with the first letter only capitalised *at actual size* (see Figure 1 and Table VII).

Contact author: author1¹, author2²

¹Address of author1.

²Address of author2 if different from author1's address
E-mail: author1@univ1.com , author2@univ2.com

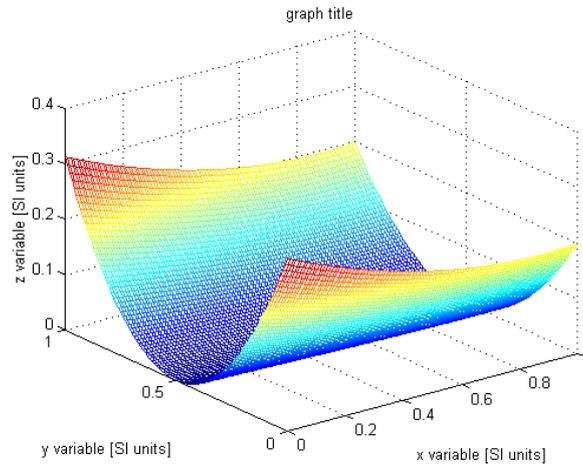


Figure 1 Simple chart.

Table VII - Experimental values

Robot Arm Velocity (rad/s)	Motor Torque (Nm)
0.123	10.123
1.456	20.234
2.789	30.345
3.012	40.456

2.2.4 Photographs and illustrations

Authors could wish to publish in full colour photographs and illustrations. Photographs and illustrations should be included in the electronic document and a copy of their original sent. Illustrations in full colour ...

2.2.5 Equations

Each equation should occur on a new line with uniform spacing from adjacent text as indicated in this template. The equations, where they are referred to in the text, should be numbered sequentially and their identifier enclosed in parenthesis, right justified. The symbols, where referred to in the text, should be italicised.

- point 1
 - point 2
 - point 3
- 1. numbered point 1
- 2. numbered point 2
- 3. numbered point 3

$$W(d) = G(A_0, \sigma, d) = \frac{1}{T} \int_0^{+\infty} A_0 \cdot e^{-\frac{d^2}{2\sigma^2}} dt \quad (1)$$

3 COPYRIGHT

Authors will be asked to sign a copyright transfer form prior to JoMaC publishing of their paper. Reproduction of any part of the publication is not allowed elsewhere without permission from JoMaC whose prior publication must be cited. The understanding is that they have been neither previously published nor submitted concurrently to any other publisher.

4 PEER REVIEW

Papers for publication in JoMaC will first undergo review by anonymous, impartial specialists in the appropriate field. Based on the comments of the referees the Editor will decide on acceptance, revision or rejection. The authors will be provided with copies of the reviewers' remarks to aid in revision and improvement where appropriate.

5 REFERENCES (DESCRIPTION)

The papers in the reference list must be cited in the text. In the text the citation should appear in square brackets [], as in, for example, "the red fox has been shown to jump the black cat [3] but not when...". In the Reference list the font should be Times New Roman with 10 point size. Author's first names should be terminated by a 'full stop'. The reference number should be enclosed in brackets. The book titles should be in *italics*, followed by a 'full stop'. Proceedings or journal titles should be in *italics*. For instance:

REFERENCES (EXAMPLE)

- [1] Smith J., Jones A.B. and Brown J., *The title of the book*. 1st edition, Publisher, 2001.
- [2] Smith J., Jones A.B. and Brown J., The title of the paper. *Proc. of Conference Name*, where it took place, Vol. 1, paper number, pp. 1-11, 2001.
- [3] Smith J., Jones A.B. and Brown J., The title of the paper. *Journal Name*, Vol. 1, No. 1, pp. 1-11, 2001.
- [4] Smith J., Jones A.B. and Brown J., *Patent title*, U.S. Patent number, 2001.

International Journal of Mechanics and Control – JoMaC
Published by Levrotto&Bella
TRANSFER OF COPYRIGHT AGREEMENT

<p>NOTE: Authors/copyright holders are asked to complete this form signing section A, B or C and mail it to the editor office with the manuscript or as soon afterwards as possible.</p>	<p><i>Editor's office address:</i> Andrea Manuello Bertetto Elvio Bonisoli <i>Dept. of Mechanics</i> <i>Technical University – Politecnico di Torino</i> <i>C.so Duca degli Abruzzi, 24 – 10129 Torino – Italy</i> <i>e_mail: jomac@polito.it</i> <i>fax n.: +39.011.564.6999</i></p>
--	---

The article title:

By: _____

To be Published in *International Journal of Mechanics and Control JoMaC*
Official legal Turin court registration Number 5320 (5 May 2000) - reg. Tribunale di Torino N. 5390 del 5 maggio 2000

A Copyright to the above article is hereby transferred to the JoMaC, effective upon acceptance for publication. However the following rights are reserved by the author(s)/copyright holder(s):

1. All proprietary rights other than copyright, such as patent rights;
2. The right to use, free or charge, all or part of this article in future works of their own, such as books and lectures;
3. The right to reproduce the article for their own purposes provided the copies are not offered for sale.

To be signed below by all authors or, if signed by only one author on behalf of all co-authors, the statement A2 below must be signed.

A1. All authors:

SIGNATURE _____ DATE _____ SIGNATURE _____ DATE _____

PRINTED NAME _____ PRINTED NAME _____

SIGNATURE _____ DATE _____ SIGNATURE _____ DATE _____

PRINTED NAME _____ PRINTED NAME _____

A2. One author on behalf of all co-authors:

"I represent and warrant that I am authorised to execute this transfer of copyright on behalf of all the authors of the article referred to above"

PRINTED NAME _____

SIGNATURE _____ TITLE _____ DATE _____

B. The above article was written as part of duties as an employee or otherwise as a work made for hire. As an authorised representative of the employer or other proprietor. I hereby transfer copyright to the above article to *International Journal of Mechanics and Control* effective upon publication. However, the following rights are reserved:

1. All proprietary rights other than copyright, such as patent rights;
2. The right to use, free or charge, all or part of this article in future works of their own, such as books and lectures;
3. The right to reproduce the article for their own purposes provided the copies are not offered for sale.

PRINTED NAME _____

SIGNATURE _____ TITLE _____ DATE _____

C. I certify that the above article has been written in the course of employment by the United States Government so that no copyright exists, or by the United Kingdom Government (Crown Copyright), thus there is no transfer of copyright.

PRINTED NAME _____

SIGNATURE _____ TITLE _____ DATE _____

CONTENTS

- 3 Walking with Adaptive Oscillator and Dynamic Movement Primitives**
T. Petrič, L. Žajpah, G. Garofalo, C. Ott
- 11 Design and Simulation for Kinematic Characteristics of a Tripod Mechanism for Biped Robots**
M. Wang, M. Ceccarelli
- 19 Design and Operation of Cassino Hexapod II**
F. Tedeschi, D. Cafolla, G. Carbone
- 27 Development of a Hybrid Path/Force Control Architecture for an Assisting Robot**
B. Großmann, T. Schlegl
- 35 Extending Continuous Path Trajectories to Point-to-Point Trajectories by Varying Intermediate Points**
H. Gatringer, M. Oberherber, K. Springer
- 45 Implementation of a 3-DOF Parallel Robot Adaptive Motion Controller**
J. Casalilla, M. Vallés, V. Mata, M. Díaz-Rodríguez, A. Valera
- 53 Wearable Sensory System for Robotic Prosthesis**
L. Ambrožič, M. Gorišič, S. Šlajpah, R. Kamnik, M. Munih
- 61 Detection of Needle Insertion Point for Phlebotomy in Human Forearms**
T.R. Savarimuthu, A.S. Sørensen