TASK-ORIENTED CONTROL FOR REDUNDANT ROBOTS

Milutin Nikolić, Branislav Borovac, Mirko Raković, *Fakultet Tehničkih Nauka, Novi Sad, {milutinn, borovac, rakovicm}@uns.ac.rs*

Abstract – *This paper proposes task prioritization framework applicable on redundant robots. It considers tasks in form of both equalities and inequalities which are performed while maintaining actuators' driving torques between saturation limits. Having in mind that a prerequisite for realization of any task by biped robot is the maintenance of its upright position this issue is also in the focus of our study. Dynamic balance was ensured by allowing the ZMP to be anywhere within the support area. Simulations were performed, and the results proved the validity of the proposed approach.*

1 INTRODUCTION

One of the very important tasks that will be assigned to humanoid robots is to manipulate objects in their environment. In doing this, they will have to be ready to reach objects and grasp them, and this is expected to be realized in a way involving the trajectory shape and realization efficiency that are very similar to those of humans. While performing tasks, humanoids have to choose such solutions to avoid collision with the foreign objects, self-collision, maintain joint coordinates between the limits, maintenance of the robot's upright posture etc. This makes a need for prioritized control scheme where topmost tasks have to be performed without regret to lover priority tasks.

Already for a long time, a lot of research has been devoted to studying kinematic redundancy [1], what is often needed for trajectory generation with obstacle avoidance. Typical methods rely on the use of generalized inverses in order to solve underconstrained problems. From an infinite number of possible solutions, a specific one is selected on the basis of the chosen optimization criterion [2]. An important advancement was made by the introduction of Operational Space [3] to control motion and force of the end effector. That approach was later extended into task prioritization framework [4] to allow secondary tasks to be handled without interfering with the primary ones.

Although that framework has come a long way, it still has some space for improvements. For example, the framework could be applied only to two types of tasks (tasks dependant on joint coordinates or on contact force). However, the tasks to be performed by robots in the environment of humans can be much more general. Also, in reality, actuators can saturate, which is not taken into account in the previous works. Saturation can lead to the deviation of the trajectory from the desired one, even for the tasks of the highest priority. This can make the whole framework inconsistent. In section 2, the new framework has been defined. In sections 3 and 4, new features are introduced. In section 5, simulation results are presented to demonstrate the improvements, while conclusions are given in section 6.

2. THE FRAMEWORK

Let us consider a robotic system consisting of n links. The coordinate, velocity, and acceleration at the i-th joint are denoted by $q_i, \dot{q}_i, \ddot{q}_i$. The joint is driven by the torque τ_i. The differential equation governing the system dynamics is:

$$\mathbf{H}\ddot{\mathbf{q}} + \mathbf{h}_0 = \boldsymbol{\tau} \tag{1}$$

where $\mathbf{H}$ is a positive definite inertia matrix, and $\mathbf{h}_0$ is the vector that comprises the velocity and gravitation effects.

Let us assume that the system is to perform a task which can be written in the form:

$$\mathbf{A}(\mathbf{q}, \dot{\mathbf{q}}, t)\ddot{\mathbf{q}} = \mathbf{b}(\mathbf{q}, \dot{\mathbf{q}}, t) \tag{2}$$

where $\mathbf{A}$ is an $m \times n$ matrix, while $\mathbf{b}$ is a vector of dimension m. The objective is to find an appropriate torque $\boldsymbol{\tau}$ which needs to be applied to (1) so that the task set by (2) is fulfilled. It can be done by applying a torque obtained by minimizing Euclidean norm in the form of:

$$\begin{array}{ll} \text{minimize} & \| \mathbf{A}\ddot{\mathbf{q}} - \mathbf{b} \|_2^2 \\ \text{subject to} & \mathbf{H}\ddot{\mathbf{q}} + \mathbf{h}_0 = \boldsymbol{\tau} \end{array} \tag{3}$$

The importance of the matrix $\mathbf{A}\mathbf{H}^{-1/2}$, demonstrated in [5] inspired us to rewrite the optimization problem (3) in the form:

$$\begin{array}{ll} \text{minimize} & \| \mathbf{B}\ddot{\mathbf{r}} - \mathbf{b} \|_2^2 \\ \text{subject to} & \ddot{\mathbf{r}} + \mathbf{p} = \mathbf{T} \end{array} \tag{4}$$

where $\mathbf{B} = \mathbf{A}\mathbf{H}^{-1/2}$ is the task matrix, $\ddot{\mathbf{r}} = \mathbf{H}^{1/2}\ddot{\mathbf{q}}$, $\mathbf{p} = \mathbf{H}^{-1/2}\mathbf{h}_0$ and $\mathbf{T} = \mathbf{H}^{-1/2}\boldsymbol{\tau}$. General olution of such probilem is:

$$\mathbf{T} = \mathbf{B}^+(\mathbf{b} + \mathbf{B}\mathbf{p}) + (\mathbf{I} - \mathbf{B}^+\mathbf{B})\mathbf{u} \tag{5}$$

where $\mathbf{B}^+$ denotes the Moore-Penrose inverse of $\mathbf{B}$. The matrix $\mathbf{I}$ is an $n \times n$ identity matrix, while $\mathbf{u}$ is an arbitrary vector. A very important implication of (5) is that if the system is to move according to (2), the torque in the form of (5) has to be applied. That torque consists of two parts, the first one is fixed and can be calculated by solving the optimization problem (4). The other part is arbitrary in the null space of matrix $\mathbf{B}$. This is possible only if the rank of the task matrix $\mathbf{B}$ is lower than n, so $\mathbf{I}$-$\mathbf{B}^+\mathbf{B}$ is a non-zero matrix. In such a case, the presence of the arbitrary term ensures an infinite number of torques that can be applied, so that the task (2) remains fulfilled. This gives a possibility to select the vector $\mathbf{u}$ in such a way that some lower-priority task can be fully or partially fulfilled. The lower-priority tasks can be controlled in the null space of higher priority tasks. This will be the basis for our task prioritization scheme.

2.1 Types of tasks

The simplest task that can be written in the form of (2) is when the task coordinates can be expressed only in terms of joint coordinates and time:

$$\mathbf{f}(\mathbf{q}, t) = \mathbf{x}_{des} \tag{6}$$

Here $\mathbf{x}_{des}$ represents an $m \times 1$ vector of desired task space coordinates. After differentiation, (6) assumes the form:

$$\frac{\partial \mathbf{f}}{\partial \mathbf{q}}\ddot{\mathbf{q}} + \left(\frac{d}{dt}\frac{\partial \mathbf{f}}{\partial \mathbf{q}}\right)\dot{\mathbf{q}} + \frac{d}{dt}\frac{\partial \mathbf{f}}{\partial t} = \ddot{\mathbf{x}}_{des} \tag{7}$$

Thus, we have obtained the holonomic task (6) in the form of (2). This class of tasks can be illustrated by the following examples: reaching a point in the space around the robot, control of joint position, positioning of CoM, etc.

A second type of tasks, apart from being dependent on joint coordinates and time, is dependent on joint velocities, and it can be written in the form:

$$\mathbf{f}\left(\mathbf{q}, \dot{\mathbf{q}}, t\right) = \mathbf{x}_{des} \tag{8}$$

After differentiation the above equation ausumes the form:

$$\frac{\partial \mathbf{f}}{\partial \dot{\mathbf{q}}}\ddot{\mathbf{q}} + \frac{\partial \mathbf{f}}{\partial \mathbf{q}}\dot{\mathbf{q}} + \frac{\partial \mathbf{f}}{\partial t} = \dot{\mathbf{x}}_{des} \tag{9}$$

Again, we have obtained the task described in the form of (2). This type of task can be illustrated by the application of a constant force to the visco-elastic body, control of the ZMP position if the foot-ground contact is not rigid.

Last type of task that can be written in form of (2) is the task that is affine in joint accelerations and can be written in the form:

$$\mathbf{C}\left(\mathbf{q}, \dot{\mathbf{q}}, t\right)\ddot{\mathbf{q}} + \mathbf{d}\left(\mathbf{q}, \dot{\mathbf{q}}, t\right) = \mathbf{x}_{des} \tag{10}$$

An example of tasks belonging to this type is the control of ZMP position in the case when the foot-ground contact is rigid.

2.2 Building the framework

Let us suppose that our goal is to control the system described by (1) in such a way to fulfil *p* tasks given by:

$$\mathbf{A}_i\ddot{\mathbf{q}} = \mathbf{b}_i \quad i = 1\ldots p \tag{11}$$

where the size of the matrix $\mathbf{A}_i$ is $m_i \times n$, and the length of the vector $\mathbf{b}_i$ is m_i. The tasks are prioritized, whereby the first task has the highest and the *p*-th the lowest priority. The higher-priority task will be realized without regard to the lower priority task. However, the lower-priority task is to be performed in such a manner that it does not interfere with the higher priority one. Since first task has the highest priority, it has to be solved first:

$$\begin{aligned}&\text{minimize} \quad \| \mathbf{B}_1\ddot{\mathbf{r}} - \mathbf{b}_1 \|_2^2 \\ &\text{subject to} \quad \ddot{\mathbf{r}} + \mathbf{p} = \mathbf{T}\end{aligned} \tag{12}$$

where the matrix $\mathbf{B}_1 = \mathbf{A}_1\mathbf{H}^{-1/2}$. As was already shown, the general solution of such minimization problem is:

$$\mathbf{T} = \mathbf{B}_1^{+}\left(\mathbf{b}_1 + \mathbf{B}_1\mathbf{p}\right) + \mathbf{N}_1\mathbf{u}_1. \tag{13}$$

The matrix $\mathbf{N}_1$ denotes the null space of the task matrix $\mathbf{B}_1$, and it can be calculated as $\mathbf{N}_1 = \mathbf{I} - \mathbf{B}_1^{+}\mathbf{B}_1$. The vector $\mathbf{u}_1$ is of the length *n*, and it can be arbitrarily selected. In the sequel, the first term of (13) will be denoted as $\mathbf{T}_1$.

Torques for performing the task of second highest priority have to be computed. This will be done in the null space of the task of highest priority, so to ensure that it will not interfere with the previous task. Based on that, it is necessary to determine $\mathbf{u}_1$, which is a solution of the following optimization problem:

$$\begin{aligned}&\text{minimize} \quad \| \mathbf{B}_2\ddot{\mathbf{r}} - \mathbf{b}_2 \|_2^2 \\ &\text{subject to} \quad \ddot{\mathbf{r}} + \mathbf{p} = \mathbf{T}_1 + \mathbf{N}_1\mathbf{u}_1\end{aligned} \tag{14}$$

A general solution of (14) can be obtained in the form:

$$\mathbf{u}_1 = \left(\mathbf{B}_2\mathbf{N}_1\right)^{+}\left(\mathbf{b}_2 - \mathbf{B}_2\left(\mathbf{T}_1 - \mathbf{p}\right)\right) + \mathbf{N}_2\mathbf{u}_2 \tag{15}$$

Now, the matrix $\mathbf{N}_2$ represents the null space of the product of the task matrix $\mathbf{B}_2$ and the null space of the task matrix $\mathbf{B}_1$, i.e, $\mathbf{N}_2 = \mathbf{I} - \left(\mathbf{B}_2\mathbf{N}_1\right)^{+}\mathbf{B}_2\mathbf{N}_1$. Similarly to the previous case, from now on, $(\mathbf{B}_2\mathbf{N}_1)^{+}(\mathbf{b}_2-\mathbf{B}_2(\mathbf{T}_1-\mathbf{p}))$ will be denoted as $\mathbf{T}_2$.

After these two iterations we are able to write a recursive formula to calculate desired torques for all the remaining tasks. Each of them will be performed in the null space of all previous (higher-priority) tasks. Thus, $\mathbf{N}_{i|prev}$ is defined as the null space of all tasks of higher priority than the task i, $\mathbf{T}_{i|prev}$ is defined as a sum of all torques to be applied to the robot joints to perform all the tasks of higher priority than the task *i*. The desired torques for all tasks can be calculated from the following expressions:

$$\begin{aligned}\mathbf{T}_i &= \left(\mathbf{B}_i\mathbf{N}_{i|prev}\right)^{+}\left(\mathbf{b}_i - \mathbf{B}_i\left(\mathbf{T}_{i|prev} - \mathbf{p}\right)\right) \\ \mathbf{T}_{i+1|prev} &= \mathbf{T}_{i|prev} + \mathbf{N}_{i|prev}\mathbf{T}_i\end{aligned} \tag{16}$$

$$\begin{aligned}\mathbf{N}_i &= \mathbf{I} - \left(\mathbf{B}_i\mathbf{N}_{i|prev}\right)^{+}\mathbf{B}_i\mathbf{N}_{i|prev} \\ \mathbf{N}_{i+1|prev} &= \mathbf{N}_{i|prev}\mathbf{N}_i\end{aligned} \tag{17}$$

The total vector of driving torques that have to be applied to the robotic system in order to perform the set of *p* tasks in a prioritized manner is given by:

$$\boldsymbol{\tau} = \mathbf{H}^{1/2}\sum_{i=1}^{p}\mathbf{N}_{i|prev}\mathbf{T}_i = \mathbf{H}^{1/2}\mathbf{T}_{p+1|prev} \tag{18}$$

3 HANDLING THE TASKS IN THE FORM OF INEQUALITIES

Not all tasks to be performed by robots can be written in the form of equalities. The tasks like 'angle at the 5-th joint should not exceed 40 degrees', 'the ZMP should be inside the support polygon' can be written only in the form of inequalities. Thus, it is of high importance to handle these types of tasks properly and to include them in the proposed framework. There are three types of tasks in the form of inequalities, and these are:

$$\mathbf{f}\left(\mathbf{q}, t\right) \preceq \mathbf{x}_{limit} \tag{19}$$

$$\mathbf{f}\left(\mathbf{q}, \dot{\mathbf{q}}, t\right) \preceq \mathbf{x}_{limit} \tag{20}$$

$$\mathbf{C}\left(\mathbf{q}, \dot{\mathbf{q}}, t\right)\ddot{\mathbf{q}} + \mathbf{d}\left(\mathbf{q}, \dot{\mathbf{q}}, t\right) \preceq \mathbf{x}_{limit} \tag{21}$$

3.1 Handling of inequalities independent of accelerations

We will introduce the threshold value $\mathbf{x}_{thr}$, in such a way that $\mathbf{x}_{thr} \prec \mathbf{x}_{limit}$. Thus, while the $\mathbf{f}\left(\mathbf{q}, t\right) \preceq \mathbf{x}_{thr}$ is fulfilled, no torque will be applied, and the framework looks as if no constraints are imposed. When the inequality becomes violated, some action is needed in order to ensure fulfillment of (19). We will separate the scalar functions which violate (19) by extracting the set of violated inequalities $S=\{i \mid f_i(\mathbf{q},t)>x_{i|thr}\}$. These functions would be handled as the tasks in the form of (6), and o react properly, correctional torques have to be applied to ensure that the values of these functions are repelled from their respective limits.

The inequalities in the form of (20) are handled in a way similar to the previous case. The only difference is that these functions are differentiated only once. So, rather than a desired acceleration, we use the desired velocity defined in (9), which would be equal to a negative gradient of some potential field.

3.2 Handling of inequalities affine in joint accelerations

Let us assume that there are *p* tasks to be performed, where the task *k* belongs to inequality type in the form of (21), given by the matrix $\mathbf{A}_k$ and vector $\boldsymbol{b}_k$. Since exact values need not to be tracked, we need to solve the feasibility problem in the form:

$$\begin{aligned} &\text{find} && \mathbf{u}_{k-1} \\ &\text{subject to} && \ddot{\mathbf{r}} + \mathbf{p} = \mathbf{T}_{k|prev} + \mathbf{N}_{k|prev}\mathbf{u}_{k-1} \\ & && \mathbf{B}_k \ddot{\mathbf{r}} \preceq \mathbf{b}_k \end{aligned} \qquad (22)$$

If the system (22) is infeasible, the task *k* will be turned into a task of equality type, in order to minimize the amount of inequality (21) that is violated.

If the system (22) is feasible, we are sure that its solution exists, but do not know how to select a solution, i.e. what is the appropriate torque to be applied. If there exists some lower-priority task, we will solve the optimization problem in the form:

$$\begin{aligned} &\text{minimize} && \| \mathbf{B}_{k+1}\ddot{\mathbf{r}} - \mathbf{b}_{k+1} \|_2^2 \\ &\text{subject to} && \ddot{\mathbf{r}} + \mathbf{p} = \mathbf{T}_{k+1|prev} + \mathbf{N}_{k+1|prev}\mathbf{u}_k \\ & && \mathbf{B}_k \ddot{\mathbf{r}} \preceq \mathbf{b}_k \end{aligned} \qquad (23)$$

Since no torque and null space were determined for the task *k*, it follows that $\mathbf{T}_{k+1|prev}$ and $\mathbf{N}_{k+1|prev}$=$\mathbf{N}_{k|prev}$. As these identities hold, and (22) is feasible, it is possible to find solution of the minimization problem (23), and such $\mathbf{u}_k$ would equal to $\mathbf{T}_{k+1}$. In order to find null space of the task *k+1*, a set of indexes of inequalities that have reached the equality sign $E = \{i \mid \mathbf{B}_{k,i}\ddot{\mathbf{r}} = b_{k,i}\}$ is defined. The next task to be optimized *k+2*, must be performed in the null space of task *k+1*. Also, it has to be performed in such a way that all inequalities remain fulfilled. Thus, it has to be performed in the null space of $\mathbf{B}_{kE}$, so the null space of the task *k+1* will be redefined as:

$$\mathbf{N}_{k+1} = \mathbf{I} - \left(\begin{bmatrix} \mathbf{B}_{k+1} \\ \mathbf{B}_{kE} \end{bmatrix} \mathbf{N}_{k+1|prev} \right)^+ \begin{bmatrix} \mathbf{B}_{k+1} \\ \mathbf{B}_{kE} \end{bmatrix} \mathbf{N}_{k+1|prev} \qquad (24)$$

The inequalities belonging to the set *E* have reached their limits and can not be allowed that new solutions 'move' in these directions. This is why these inequalities are added to the task matrix $\mathbf{B}_{k+1}$ when calculating null space.

4 FRAMEWORK FOR CONSTRAINED SYSTEMS

Until now, the discussion has been focused on an unconstrained robotic system. But, in the case the robot interacts with its surroundings, it would be constrained. When standing or moving, humanoid robots are in contact with the ground, which makes the robotic system constrained. Both types of constraints, holonomic and nonholonomic, can be described [5] by:

$$\mathbf{A}_0 \ddot{\mathbf{q}}_0 = \mathbf{b}_0 \qquad (25)$$

Joint accelerations of the unconstrained system equals $\mathbf{a} = \mathbf{H}^{-1}(\boldsymbol{\tau} - \mathbf{h}_0)$. According to [5], the acceleration of the constrained system is:

$$\ddot{\mathbf{q}} = \mathbf{a} + \mathbf{H}^{-1/2}\left(\mathbf{A}_0\mathbf{H}^{-1/2}\right)^+ \left(\mathbf{b}_0 - \mathbf{A}_0\mathbf{a}\right) \qquad (26)$$

which can be rewritten as:

$$\ddot{\mathbf{r}} + \mathbf{p} = \mathbf{B}_0^+\left(\mathbf{b}_0 + \mathbf{B}_0\mathbf{p}\right) + \left(\mathbf{I} - \mathbf{B}_0^+\mathbf{B}_0\right)\mathbf{T}. \qquad (27)$$

From (27) it is clear that the applied torque will act in the null space of the constraint $\mathbf{N}_c = \mathbf{I} - \mathbf{B}_0^+\mathbf{B}_0$. The first term on the right-hand side of (27) we will denote as $\mathbf{T}_c = \mathbf{B}_0^+\left(\mathbf{b}_0 + \mathbf{B}_0\mathbf{p}\right)$. Since the equation governing dynamics of the system has changed, in order to perform the task *i*, we will minimize:

$$\begin{aligned} &\text{minimize} && \| \mathbf{B}_i\ddot{\mathbf{r}} - \mathbf{b}_i \|_2^2 \\ &\text{subject to} && \ddot{\mathbf{r}} + \mathbf{p} = \mathbf{T}_c + \mathbf{N}_c\mathbf{T}_{i|prev} + \mathbf{N}_c\mathbf{N}_{i|prev}\mathbf{u}_{i-1} \end{aligned} \qquad (28)$$

Only the definition of $\mathbf{N}_i$ has changed, and now it is:

$$\mathbf{N}_i = \mathbf{I} - \left(\mathbf{B}_i\mathbf{N}_c\mathbf{N}_{i|prev}\right)^+ \mathbf{B}_i\mathbf{N}_c\mathbf{N}_{i|prev}. \qquad (29)$$

In this way, the task prioritization scheme has been adapted for constrained robotic systems.

5 SIMULATION RESULTS

In this chapter we present the simulation results obtained by applying the proposed task prioritization framework on the example of a complex and redundant humanoid robot. As first, we describe briefly the model, and then present the simulation result that illustrate the proposed method, its capabilities and improvements.

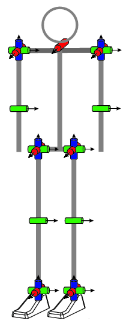

Figure 1: Kinematic structure of the robot

5.1 Humanoid model

The humanoid model is very similar to the models used by the same authors in the previous papers [6] The humanoid is modeled as a set of kinematic chains connected to the central (base) link and interconnected by rotational joints with only one DOF. Fig. 1 shows a sketch of the kinematic structure of robot consisting of 12 links. The base link (link to which all other kinematic chains are connected) is the trunk. The joints with more DOFs (shoulder, ankle and hip) are modeled as a series of one-DOF joints connected by massless and lengthless links. The position and orientation of the trunk are defined by a vector consisting of three translational and three angular coordinates. The position of each joint is described by one joint coordinate. Each joint has its own actuator that generates the corresponding driving torque. First six DOFs are unactuated. The differential equations governing the system dynamics is (1).

5.2 Simulation

In order to emphasize the dynamic effects, the whole motion was assumed to be fast. This kind of movement induces high accelerations, which can jeopardize dynamic balance. High accelerations can drive the actuators into saturation, which can cause deviations from desired paths. At the highest rank were the tasks that ensured structural integrity of the robot, i.e. to avoid damaging itself. Task of the highest priority is 'the torques at all joints must be between saturation values'. The task of second highigest priority is the constraint on the ZMP position in the *x* and *y* directions was introduced. Third highest priority task is that the coordinate at each joint had to be kept between the upper and lower limits, and fourth was to maintain the CoM above the center of the support area. This task ensured statical balance after completing the motion. In the second group of tasks the robot had to act in a desired manner. The first and the second task of this group are of the same priority: reaching a desired height with both hands simultaneously. The lowest-priority task, is the task of preserving the posture. This ensures that the robot should remain motionless after completing all higher-priority tasks. This ensures the highest possible maneuverability, since the joint limits are furthest apart. For each of the tasks of equality type, a simple PD controller with the velocity saturation was employed.

The simulated robot movement is shown in Fig.2. It can be seen that the hands followed closely the desired linear path. The coordinates of the left hand in the *x* and *z* directions are shown in Fig. 2; the *y* direction is omitted since the hands move in the planes orthogonal to it. The rate of the change of the coordinates in the interval of 0-1*s* is constant, because of the saturation of the employed PD controller. The positions of the ZMP and CoM in the direction are also shown (Fig. 2). As can be seen from Fig. 2, the ZMP always remained within the safety zone, which ensured that the robot's dynamic balance and upright posture were not jeopardized. It is also evident that the torque at the hip joint never exceeded the predefined saturation value. This means that the movement was performed exactly as expected.

7 CONCLUSION

In this paper we proposed improved task prioritization scheme for redundant robots. The desired torques are calculated by minimizing the square of Euclidean norm of deviation of the actual path from the desired one. In this way, the need for the definition of a Jacobian matrix for the relationship between joint coordinates and task coordinates was bypassed. This enabled us to define the task in a more general form. The tasks of inequality type were resolved in a way involving the smallest possible movements at the joints to perform the desired tasks. This allows the robot to perform a larger number of tasks at the same time. The calculation of the desired torque was modified only slightly, to accommodate the system to multiple contacts and constraints. Simulation has shown that robot controlled by using proposed framework succeded in compleeting all of the tasks, although some of them were conflicting other ones. In this way, the improvements were verified and the usefulness of the proposed framework was justified.

ACKNOWLEDGEMENTS

This work was funded by the Ministry of Science and Technological Development of the Republic of Serbia in part under contract TR35003 and in part under contract III44008 and by Provincial Secretariat for Science and Technological Development under contract 114-451-2116/2011.

REFERENCES

[1] J. Hollerbach and Ki Suh. “Redundancy resolution of manipulators through torque optimization.” *IEEE Journal of Robotics and Automation* , vol. 3(4), pp. 308-316, August 1987.

[2] I.A. Gravagne and I.D. Walker. “On the structure of minimum effort solutions with application to kinematic redundancy resolution”. *IEEE Trans. on Robotics and Automation,* vol. 16(6), pp. 855-863, december 2000.

[3] O. Khatib. “A unified approach for motion and force control of robot manipulators: “The operational space formulation.” IEEE Journal of Robotics and Automation, vol 3(1), pp. 43-53, february 1987.

[4] L. Sentis and O. Khatib. “Synthesis of whole-body behaviors through hierarchical control of behavioral primitives”. *Int. Journal of Humanoid Robotics,* vol. 2(4), pp. 505-518, december 2005.

[5] F. E. Udwadia and R. E. Kalaba. "Analytical Dynamics: A New Approach".Cambridge University Press, 1996. ISBN 978-0521048330.

[6] B. Borovac, M. Nikolić, and M. Raković. “How to ompensate for the disturbancesthat jeopardize dynamic balance of a humanoid robot?” *Int. Journal of Humanoid .Robotics,* vol 8(3) , pp. 533-578, september 2011

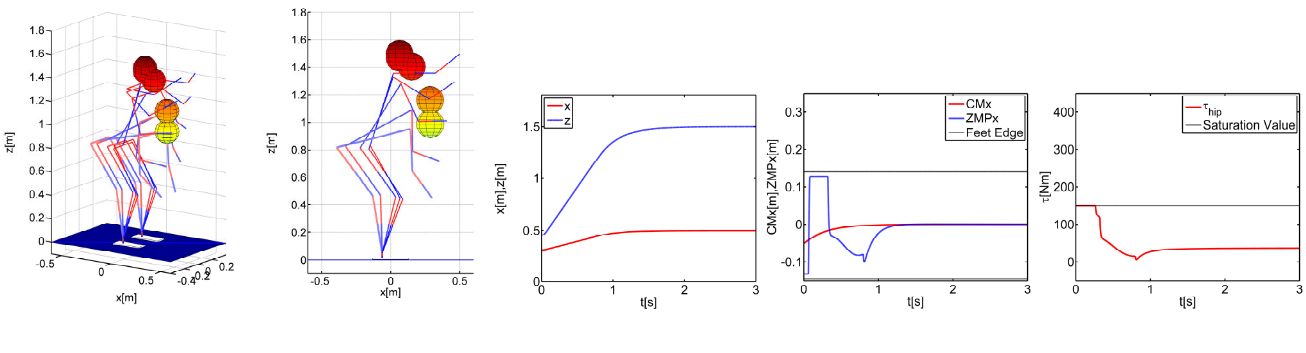

Figure 3: Movement and data obtained by simulating the system with added tasks in the form of inequalities:a) movement in isometric view b)movement in sagittal plane c) position of the left hand, d) position of C oM and ZMP in the x direction and e) torque at the hip