Comparison of Position and Torque Whole Body Control Schemes on the Humanoid Robot TALOS

N Ramuzat, G Buondonno, S Boria, Olivier Stasse

To cite this version:

N Ramuzat, G Buondonno, S Boria, Olivier Stasse. Comparison of Position and Torque Whole Body Control Schemes on the Humanoid Robot TALOS. 20th International Conference on Advanced Robotics (ICAR), Dec 2021, Virtual event, Slovenia. 10.1109/ICAR53236.2021.9659380. hal-03145141v2

HAL Id: hal-03145141
https://hal.science/hal-03145141v2
Submitted on 6 Jan 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Comparison of Position and Torque Whole-Body Control Schemes on the Humanoid Robot TALOS

N. Ramuzat*, G. Buondonno†, S. Boria*, O. Stasse†
* Airbus, Toulouse, France
† LAAS-CNRS, Université de Toulouse, France

Abstract—Most control architectures for legged locomotion are either torque or position controlled. In this paper, we investigate their differences and performances. Aiming to choose the most appropriate scheme for the robot TALOS, we benchmark three control schemes: The first one optimizes joint velocities based on hierarchical quadratic programming; the second one optimizes joint accelerations based on weighted quadratic programming; and the last one optimizes joint torques, also based on weighted quadratic programming. We compare these controllers in terms of tracking error, energy consumption and computational time by using Gazebo simulations of the robot walking on flat horizontal ground, tilted platforms, and stairs. Remarkably, our torque control scheme allowed TALOS to walk forward at 0.6 m/s, the highest walking velocity achieved so far in simulation.

I. INTRODUCTION

Bipedal locomotion of humanoid robots is considered as a difficult problem because of the complexity of the robot dynamics, the numerous constraints of the motion and the unknown environment. Three stages are usually considered to decompose this problem: the contact sequence generation, the trajectory planning and the whole-body control.

Most of trajectory planning methods use the centroidal dynamics to generate consistent behaviors for a legged robot. In addition, the concept of Divergent Component of Motion (DCM) [1] is associated to reduced dynamics models such as the Linear Inverted Pendulum (LIPM) [2] in trajectory generation. The DCM approach is also used in control, for admittance control on the Center of Mass (CoM) [3].

This paper focuses on the implementation and comparison of three real-time whole-body controllers using the task-function approach[4, 5]. The objectives to be performed by the robot are expressed in their respective task spaces, using reference trajectories given by the planning. Complex motions combine several nonlinear tasks and constraints. Quadratic Programming (QP) are fast optimization techniques used to solve such nonlinear problems, employing the whole-body kinematics or dynamics of the robot. In this paper two types of QP formulations are compared, a Hierarchical QP which imposes a strict hierarchy between the tasks [6, 7], and a weighted QP which sets weights to prioritise the tasks [8, 9].

In the recent literature there is a growing number of implementations of torque control algorithms to solve locomotion problems [8, 10, 11]. Indeed, due to the intrinsic compliance of the torque control formulation, it is more suitable for interactions with humans and for multi-contact problems where external interactions and several contact points are needed. However, the transition from the simulations to the real experiments are harder due to inaccuracies on the actuation chain model [12].

This paper intends to follow the existing benchmarking of humanoid robots control architectures [13]. It contributes toward the implementation and comparison of three whole-body control schemes: two using position control associated with DCM and CoM admittance controls and one using torque control. The first one is based on a Lexicographic QP using Inverse Kinematics (denoted IK in this paper), while the second and the third one use a Weighted QP (WQP) with Inverse Dynamics and an Angular Momentum (AM) task (denoted respectively TSID position and TSID torque). They are evaluated in Gazebo simulations on three locomotion problems: walking on flat, uneven terrains and stairs (Fig. 1), on the criterion of trajectory tracking, energy consumption, passivity and computational cost. As a first consequence of our torque control scheme, we achieve the highest walking velocity for the robot TALOS in simulation: 0.6 m/s.

We organize the article as follows: Section II recalls the centroidal dynamics equations, the DCM control and the AM task. Section III details the three task-space whole-body control schemes compared in this paper. Section IV presents the energy criterion employed. Section V describes the planning methodologies used to obtain the reference trajectories for the simulations. Then, Section VI presents these simulations results and Section VII discusses them.
II. CENTROIDAL DYNAMICS

Our robot TALOS is a humanoid robot of 1.75m tall and about 100kg, composed of 32 joints and an under-actuated part called floating-base (38 Degrees-of-Freedom in total). The under-actuated part of the robot whole-body dynamics is called the centroidal dynamics. It uses the Newton-Euler equations of motion which couple the variations of the centroidal momentum with the contact forces [14]:

\[
\begin{cases}
   \dot{m}\dot{c} + \sum_i f_i + mg = i_c \\
   \dot{c} = \omega (\xi - c) \\
   \dot{\xi} = \omega (\xi - z) \\
   \xi = c + \xi
\end{cases}
\]

with \(c, \dot{c}, \ddot{c}\) the CoM position, velocity and acceleration, \(\dot{L} = \sum_i [R_i \dot{x}_i w_k - R_k (\dot{L}_i w_k - w_k)]\) and \(g = [0, 0, -9.81]^T\), where \(R_k \in SO(3)\) is the 3d rotation matrix between the k\(^{th}\) body frame and the inertial coordinate frame, \(I_k\) its inertial matrix, \(w_k\) its angular velocity, \(m\) is the mass of the robot, \(f_i \in \mathbb{R}^3\) the vector of contact forces at contact point \(i\), \(p_i \in \mathbb{R}^3\) their positions and \(\tau_i \in \mathbb{R}^3\) their contact torque (represented at the inertial coordinate frame). \(i_c, k_c, k_d \in \mathbb{R}\) are the linear and angular momentum around the CoM.

A. Divergent Component of Motion

We use the DCM formulation for the admittance control of the CoM. Under the assumptions of the LIPM, one can obtain the following set of equations [11, 13]:

\[
\begin{align*}
\dot{z} & = \omega \xi - c \\
\dot{\xi} & = \omega (\xi - z) \\
\dot{\xi} & = c + \xi
\end{align*}
\]

with \(z, \xi\) respectively the Zero Moment Point (ZMP) and DCM and \(\omega = \sqrt{\dot{g} / c}\). These equations show that the DCM diverges from the ZMP, while the CoM converges to the DCM. Thus, the DCM can be controlled to stabilize the system [2, 15, 11, 16]. We use in this paper a Proportional–Integral (PI) control on the DCM (the integral term is used to eliminate the steady-state error) presented in [3].

In terms of ZMP, the obtained control law is [3]:

\[
\begin{align*}
z^\ast & = z^{ref} - [1 + k_{com} \omega] (\xi^{ref} - \xi) \\
& + k_{com} (\xi^{ref} - z) - k_{com} \omega (\xi^{ref} - \xi)
\end{align*}
\]

with \(z^{ref}, \xi^{ref}\) the respective ZMP and DCM reference values, given by the planning. Finally, this desired ZMP is used into a CoM admittance control as [3]:

\[
e^\ast = z^\ast + kp_{adm} (z - z^\ast)
\]

The two position control schemes presented in this paper use this stabilization formulation. In the Fig 2, the Eq 3 is implemented in the DCM Ctrl blue block and the Eq 4 in the CoM Admittance Ctrl one. See Table I for the gains value used in the simulations.

B. Centroidal Momentum Tasks

The objective is to consider the angular momentum part of the Euler equation generated by the contact transition [17]. Using the equation Eq. 1, the centroidal dynamics is therefore defined by \(h_c = [l_c, k_c] \in \mathbb{R}^6\). In [18], the task formulation of the centroidal dynamics control is given by \(h_c = A_G(q) \dot{q}\).

The angular momentum task in TSID is expressed as in the equation Eq. 5, successfully implemented in [19] (the gains are defined in Table I).

A. Lexicographic Quadratic Programming

The first controller used is a Lexicographic QP task-based inverse kinematics described in [19]. In this controller, the task errors \(e\) is reduced in the cost function are implemented as velocity-based tracking laws in the Lie group \(SE(3)\). Having the robot configuration vector \(q\) and the joint velocity \(\dot{q}\) as control input, a task-function is a derivable function \(x(q)\) whose space is named the task-space. The task errors \(e\) are expressed as:

\[
\begin{align*}
\dot{e}(q, t) & = \dot{x}(q) - \dot{x}^\ast(t) \\
\dot{x}(q) & = J\dot{q}
\end{align*}
\]

with \(J = \frac{\partial \eta}{\partial \eta} \frac{\partial x}{\partial \eta}\) the Jacobian according to the robot state vector.

The following dynamics is imposed on these errors:

\[
\begin{align*}
\dot{e}(q, t) & = K_P (x(q) \otimes x^\ast(q)) \\
\Leftrightarrow \dot{x}(q) & = \dot{x}^\ast(t) + K_P (x(q) \otimes x^\ast(q))
\end{align*}
\]

with \(\otimes\) the difference operator of Lie group.

Inverse Kinematics PQ: IK - This control scheme is based on a DCM controller (Eq[3]), a CoM admittance controller (Eq[4]) and a Lexicographic PQ solving the inverse kinematics of the robot (see Fig 2). The authors have implemented this scheme in an open-source package [20], based on the PQ in [19], adding the DCM and CoM admittance controllers.

The tasks used during the simulations are (the priority 0 is the highest one):

- Feet tracking (priority 0)
- CoM height tracking (priority I)
- CoM lateral-sagittal tracking (priority II)
- Waist orientation (priority III)
- Posture regularization in half-sitting (priority IV)

The respective task gains are defined in Table I.
B. Task Space Inverse Dynamics (TSID)

TSID \[21\] is a WQP which sums the task functions in a general cost function using weights to define their priorities (as opposed to the IK controller it is not a strict hierarchy, it has only two strict layers: the constraint and the cost). In this controller, the task errors \(e\) to be reduced are implemented as acceleration-based tracking laws in the task space. Having the robot configuration vector \(q\) and the joint acceleration \(\ddot{q}\) as control input, a task-function is a second-order derivative function \(x\) of \(q\). And the task errors \(e\) are expressed as:

\[
\begin{align*}
\dot{e}(q,t) &= \ddot{x}(q) - \ddot{x}^*(t) \\
\ddot{e}(q,t) &= (J\dddot{q} + \dot{J}\ddot{q}) - \dddot{x}^*(t)
\end{align*}
\]  

(8)

The following dynamics is imposed on these errors:

\[
\begin{align*}
\dot{\ddot{x}}(q) &= \dot{x}^*(t) + K_P(x(q) \ominus x^*(t)) + K_D(\dot{x}(q) \ominus \dot{x}^*(t)) + \\
\dddot{x}(q) &= \dddot{x}^*(t) + K_P(x(q) \ominus x^*(t)) + K_D(\ddot{x}(q) \ominus \dddot{x}^*(t))
\end{align*}
\]  

(9)

TSID solves the inverse dynamics of the robot in rigid contact with the environment \[7\] and has been successfully used on HRP-2 robot in \[22\].

Inverse Dynamics WQP: TSID Position - This control scheme is based on a DCM controller (Eq[4]), a CoM admittance controller (Eq[5]) and a WQP solving the inverse dynamics of the robot, see Fig.2. Compared to the previous controller, this one implements an AM task, which regulates the angular momentum to zero, using the formulation of Eq[5]. The authors have implemented this controller using the TSID \[21\] library in the same package than the controller TSID Torque, with the DCM and CoM admittance controllers.

The tasks considered during the simulations are:

- Feet tracking (priority 0)
- Feet contacts (priority 0)
- CoM height tracking (priority I, weight 10^3)
- CoM lateral-sagittal tracking (priority I, weight 10^3)
- Waist orientation (priority I, weight 1)
- Posture regularization in half-sitting (priority I, weight 0,1)
- AM velocity-acceleration regularization (priority I, weight \(2 \times 10^{-5}\))

The respective task gains are defined in Table II. The weights and gains have been chosen through trials and errors with an apriori heuristic.

Inverse Dynamics WQP: TSID Torque - This control scheme is based on a WQP solving the inverse dynamics of the robot (with an AM regularization task, using the formulation of Eq[5], as shown in Fig.3). From the desired acceleration computed by the QP, TSID retrieves the associated torque by using the robot equation of the dynamics. The authors have implemented this controller using the TSID \[21\] library in the open-source package \[23\].

The tasks considered in the simulations are the same as TSID position, with different gains (see Table II).

C. Remark on the state feedback

For position control, it is needed to integrate the result of the QP (one time for IK and two times for TSID position, see Fig[2] to obtain the desired command. To avoid instabilities, the control loop of both QP use these integrated values in the next iteration instead of the measured ones. The measured position and velocity of the robot are only used to compute the CoM, DCM and ZMP for the admittance control in the position schemes. In contrary, the torque control scheme uses the measured values at each iteration of the QP (see Fig[3] and in particular the position and velocity of the robot base (or free-flyer).

IV. ENERGETIC COMPARISON CRITERION

A. Energy cost

Based on \[24\], a relevant criteria to compare the energy consumption of the control schemes is the cost of transport. It can be computed as the energetic cost of transport \(C_{et}\) using the whole mechanical work of the actuation system \(E_m\) or as the mechanical cost of transport \(C_{mt}\) using only the positive one \(E_{m+}\).

\[
C_{et} = \frac{E_m}{mgD} \quad C_{mt} = \frac{E_{m+}}{mgD} \quad E_m = \sum_{i=0}^{N} \left| \tau_i(t) \omega_i(t) \right| dt
\]

with \(m\) the mass of the system, \(g\) the gravity constant, \(D\) the distance traveled by the system and \(\tau_i, \omega_i\) the respective torque and velocity of each robot joint for all (N) joints.

B. Passivity Gait Measure

Another interesting energetic criteria is the ability to minimize joint torques to increase the passivity of the walk\[24\]. The Passivity Gait Measure (PGM) \[25\] quantifies the passivity of a biped walking motion:

\[
PGM = 1 - \frac{RMS(\tau_{sa})}{RMS(\tau_{tot})}
\]

(11)

\[
RMS(\tau_{tot}) = \sqrt{\frac{\int_0^T \left[ \sum_{i=0}^{N} \tau_i(t)^2 \right] dt}{T}}
\]

(12)

where RMS is the Root Mean Square along the period of time \(T\), \(\tau_{sa}\) stands for the torque on the stance ankle joint and \(\tau_{tot}\) for the torque on all robot joints.

V. LOCOMOTION PLANNING

A. Walking Pattern Generator

The trajectories used in the straight walk simulations have been computed using the algorithm described in \[26\] \[27\] \[28\]. This algorithm provides desired trajectories for the ZMP \(z^*\), the CoM \(c^*\), and the feet \(p_i^*\), for a given set of footsteps (pre-defined in these simulations). This implementation uses
the centroidal dynamics and the dynamic filter proposed in [26] computed with the Recursive Newton-Euler Algorithm [29] implemented in the Pinocchio library [30]. The CoM trajectory is modified to take into account the momentum generated by the swing leg thanks to iterations to the pattern generator method, it takes into account the centroidal dynamics (see Section II) using two convex relaxations based on trust regions [33]. Similarly to the pattern generator method, it takes into account the momentum generated by the swing leg thanks to iterations between a kinematic whole-body formulation and the centroidal dynamic optimization. In contrast, when solving Eq. [2], it does not assume that \( \dot{L} = 0 \) (see Section II).

### VI. Simulation Results

The simulations realized in this paper have been made using Gazebo. A video illustrating the simulations is available at the following link: [https://peertube.laas.fr/videos/watch/4b5d3a5b-2355-47a0-8197-f41ed4f885c6](https://peertube.laas.fr/videos/watch/4b5d3a5b-2355-47a0-8197-f41ed4f885c6). The chosen simulations are walking on flat or uneven terrains and stair climbing. Based on [24], they cover different aspects of locomotion skills for a stationary environment with and without unexpected disturbances.

#### A. Straight walk of 20 cm steps

In the simulation, the robot executes 6 steps forward at 0.2m/s and a final step (traveled distance of 1.2m). The time distribution is 0.9s for single support phase and 0.115s for double support phase (leading to steps of approx. 0.20m).

The controllers have also been successfully tested on a faster walk with single/double support time of 0.711/0.089s. The Fig. 4 presents a comparison of the three control schemes on their estimated ZMP, on the sagittal (x-axis, top curves on the figure) and lateral (y-axis, bottom curves) planes only, because the desired height of the CoM is constant. Fig. 5 shows the forces applied on the ground along the z-axis on the left foot. The tracking of the Com and the feet are accurately followed by the three controllers (tracking error lesser than 1cm).

The two position controllers achieve similar results, tracking correctly the ZMP reference of Eq. [3] with an average error of 2cm (see Table II). Noticeably, the torque control presents a ZMP which is close to the position control results in Fig. 4 even though there is no explicit control on the ZMP nor the DCM. In the Tables presenting the error on the ZMP, the controllers when the foot is in the air are due to the high control gains on the ankle and feet position control compared to the better foot landing observed in Fig. 3.

#### B. Straight walk of 60 cm steps in torque control

In [16] the humanoid robot TORO successfully performed a walk on flat terrain with a step length of 55cm (single/double support time of 1.1/0.4s). In the following simulation, the torque controller is pushed to its limits to show

<table>
<thead>
<tr>
<th>Tasks Gains (20cm stairs)</th>
<th>IK</th>
<th>TSID position (20cm stairs)</th>
<th>TSID torque (20-60cm/stairs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( K_{\text{room}} )</td>
<td>100</td>
<td>1000</td>
<td>20( \text{%} )</td>
</tr>
<tr>
<td>( K_{\text{dom}} )</td>
<td>-</td>
<td>300</td>
<td>3</td>
</tr>
<tr>
<td>( K_{\text{roomH}} )</td>
<td>100</td>
<td>1000</td>
<td>-</td>
</tr>
<tr>
<td>( K_{\text{roomL}} )</td>
<td>-</td>
<td>300</td>
<td>-</td>
</tr>
<tr>
<td>( K_{\text{waist}} )</td>
<td>300</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>( K_{\text{waistH}} )</td>
<td>-</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>( K_{\text{contacts}} )</td>
<td>1000</td>
<td>30</td>
<td>30-100/30</td>
</tr>
<tr>
<td>( K_{\text{contactsH}} )</td>
<td>-</td>
<td>11</td>
<td>11-011</td>
</tr>
<tr>
<td>( K_{\text{foot}} )</td>
<td>1000</td>
<td>2000</td>
<td>1000/500</td>
</tr>
<tr>
<td>( K_{\text{footL}} )</td>
<td>-</td>
<td>20</td>
<td>12</td>
</tr>
<tr>
<td>( K_{\text{footR}} )</td>
<td>-</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>( K_{\text{posture}} )</td>
<td>100</td>
<td>see below</td>
<td>see below</td>
</tr>
<tr>
<td>( K_{\text{postureL}} )</td>
<td>2( \sqrt{K_{\text{posture}}} )</td>
<td>2( \sqrt{K_{\text{posture}}} )</td>
<td></td>
</tr>
<tr>
<td>( K_{\text{roomAdm}} )</td>
<td>15/45</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>( K_{\text{adm}} )</td>
<td>8/25</td>
<td>8</td>
<td>-</td>
</tr>
<tr>
<td>( K_{\text{dom}} )</td>
<td>1</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>( K_{\text{domL}} )</td>
<td>1</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>( K_{\text{domR}} )</td>
<td>1</td>
<td>1</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TSID Gains</th>
<th>Legs</th>
<th>Torso</th>
</tr>
</thead>
<tbody>
<tr>
<td>( K_{\text{posture}} )</td>
<td>[10, 5, 5, 1, 10, 10]</td>
<td>[100, 100]</td>
</tr>
<tr>
<td>( K_{\text{adm}} )</td>
<td>[50, 10, 10, 10, 50, 10, 10, 10]</td>
<td>[100, 100]</td>
</tr>
</tbody>
</table>

**TABLE I:** Tasks gains of the control schemes. tilted platforms and stairs simulations use the same gains.

**TABLE II:** ZMP error of the 20 cm step walk simulation.
its capability to achieve a similar result. The robot TALOS executes 6 steps forward of 0.6m/s and a final one to go back to the initial position. The time distribution used is of 0.9s for single support phase and 0.115s for double support phase (leading to steps of approx. 60 cm).

Figure 6 presents the results obtained on the tracking of the feet and the CoM (see Table III); the ZMP and DCM estimations. The feet tracks well the desired trajectories along the y-axis (maximum error of 6 mm) however, along the x-axis, they show some delay (maximum error of 6 cm). Thus, it induces greater tracking errors on the x-axis for the CoM (peaks of 5 cm along the x-axis and 1.5 cm along the y-axis).

One can notice that the DCM and ZMP along the x-axis are more stable, whereas along the y-axis they present large oscillations (which are caused by the feet impacts on the ground when landing).

In Fig. 7 the AM behavior is shown along the three axes. The AM task minimizes the momentum to zero. The x and y momentum components are the most solicited, leading to the inclination of the torso forward and backward and to important moves of the arms to compensate the delay of the CoM and succeed the 60 cm steps. The authors observed that without this AM task, the walk cannot be achieved.

C. Walk on the tilted platforms: Uneven terrain

In this third simulation, the robot walks on tilted platforms which represent uneven terrain (Fig. 1). This walk is achieved using the multicontact-locomotion-planning trajectories (see Section V-B). The framework ensures the stability of the controllers on non-flat terrain when the feet are tilted.

D. Climbing Stairs

In the last simulation the robot is climbing 6 stairs of 10 cm height and 30 cm long (see Fig. 1). The trajectories are planned with the multicontact-locomotion-planning. Fig. 9 shows the ZMP evolution of each controllers, where the result is similar to the uneven terrain simulation. The TSID torque scheme behave significantly better than the others, with a ZMP matching the one planned (errors lesser than 1 cm, see Table V). Noticeably, the IK scheme presents higher oscillations at the end of the move in the lateral plane. The robot ends displaced on the right compared to the desired trajectories, due to slippages of the feet when it finishes to climb a stair (shown in the linked video).

E. Energy cost and Passivity Gait Measure

The results obtained for the cost of transport of the four simulations are presented in the Table VI depending on the control scheme. The results obtained for iCub in [13] are
The control scheme in torque shows much more passive behavior (except on the stance foot), with a completely passive foot during the flying phase. During the double support phase, the ankle is almost passive (PGM ~ 0.9) which is close to the human result. These results are better than the one expected in [25], where the torque controlled robot has a higher control on its stance ankle (PGM = 0.2).

Finally, on the uneven terrain, the double support phase corresponds to the worst case where the robot has its two feet tilted to keep its balance on two opposite platforms. This leads to a greater actuation than on flat floor (decreasing the passivity). Similarly, the stance phase corresponds to the left support phase on the final platform (highest slope), also leading to a bigger actuation of the ankle.

**F. Execution time of the control schemes**

The computational time obtained during the execution of one control loop of the three schemes are presented in Table VIII according to the simulations.

The computational time of the IK is better due to the computational efficiency of the null space projectors of the tasks. Exploiting this specific structure allows it to keep its control frequency higher than 1kHz in average with 4 hierarchy levels. In TSID this method can only be used once because it is composed of two strict layers: the constraints and the cost.

**VII. DISCUSSION**

For the PGM results of the position schemes, the authors believe that adding an admittance control [3] on the ankle orientation may improve the results. If added, one can expect an increase of the actuation of the ankle during the stance phase, leading to a smaller PGM value.

In general the IK scheme presents higher oscillations and slippages when adding contacts. The authors think that this issue is mitigated in TSID position because it separates the feet task into a contact task and a tracking task. It allows to have different gains depending on the context (contact or not), indeed, the TSID schemes have higher gains for tracking tasks than for the contact ones.

One major point to discuss is the transition from the simulations to the real experiments. For torque control, in the Gazebo simulator, the joint torque control is almost perfect because the dynamics of the motor is completely neglected. However, not taking these dynamics into account will lead
The control system as external disturbances. Nonetheless, to achieve the experiments, it will be necessary to take into account this flexibility [36]. It is important to mention that the final real robot implementations will require slightly different gains and weights.

**TABLE VIII: Comparison of the execution time.**

<table>
<thead>
<tr>
<th>Control Scheme</th>
<th>Simulation</th>
<th>20cm (60cm)</th>
<th>Platforms</th>
<th>Stairs</th>
</tr>
</thead>
<tbody>
<tr>
<td>IK</td>
<td>Average</td>
<td>0.5ms</td>
<td>0.7ms</td>
<td>0.6ms</td>
</tr>
<tr>
<td></td>
<td>Peaks</td>
<td>2ms</td>
<td>4ms</td>
<td>4ms</td>
</tr>
<tr>
<td>TSID</td>
<td>Average</td>
<td>1.2ms</td>
<td>1.2ms</td>
<td>1.5ms</td>
</tr>
<tr>
<td>position</td>
<td>Peaks</td>
<td>4.5ms</td>
<td>4.3ms</td>
<td>4.2ms</td>
</tr>
<tr>
<td>torque</td>
<td>Peaks</td>
<td>2.8ms (6ms)</td>
<td>5ms</td>
<td>5.5ms</td>
</tr>
</tbody>
</table>

**CONCLUSION**

Three whole-body control implementations are compared in this paper. Two of them are position based (with DCM and CoM admittance control): a Lexicographic QP using inverse kinematics and a WQP using TSID with an AM task. The last one is a WQP using TSID in torque with an AM task. They are evaluated in Gazebo on flat, uneven terrains and stairs climbing; on the criterion of trajectory tracking, energy consumption, passivity and computational cost.

In general, both position control schemes present the same results, with less energy consumption and higher passivity for the TSID position controller. A better tuning of the tasks gains may improve its results on the ZMP tracking.

On the other hand, the TSID torque controller shows better results in terms of smoothness of the trajectory tracking, energy consumption, passivity of the walk - without impacts and can achieve a 60cm walk with steps of 1s in simulation. This confirms the high capabilities of a torque control scheme coupled with an angular momentum regularization (see for instance Atlas in DARPA robotics challenge [8]). In average, the TSID controllers reach the 1kHz of control loop, necessary for real-time control, nonetheless, the IK scheme has the best computational time.

For our future works, we plan to control the hip flexibility of TALOS, so that we can evaluate the three controllers on the real robot. Moreover, it would be interesting to compare the controllers on different robotics platforms.

**REFERENCES**