



**HAL**  
open science

# 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<sup>\*†</sup>, G. Buondonno<sup>†</sup>, S. Boria<sup>\*</sup>, O. Stasse<sup>†</sup>

<sup>\*</sup> Airbus, Toulouse, France

<sup>†</sup> 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 0.6m/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

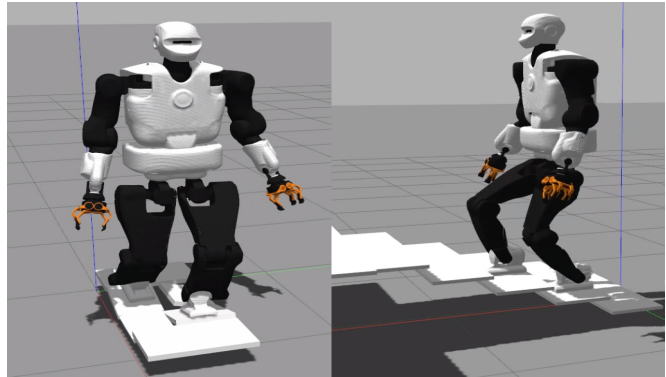


Fig. 1: Walking on Tilted Platforms and Climbing Stairs.

problems [8, 7, 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: 0.6m/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.

The authors want to thank Pierre Fernbach for his help with the *multicontact-locomotion-planning* framework and Nahuel Villa for the proofreading of the paper. This work is supported by the cooperation agreement ROB4FAM.

## II. CENTROIDAL DYNAMICS

Our robot TALOS is an 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]:

$$\begin{cases} m\ddot{c} &= \sum_i f_i + mg &= \dot{l}_c \\ mc_\times(\ddot{c} - g) + \dot{L} &= \sum_i (p_i - c_i) \times f_i + \tau_i &= \dot{k}_c \end{cases} \quad (1)$$

with  $c, \dot{c}, \ddot{c}$  the CoM position, velocity and acceleration,  $\dot{L} = \sum_k [R_k I_k \dot{w}_k - R_k (I_k w_k) \times w_k]$  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).  $l_c$  and  $k_c \in \mathbb{R}^3$  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 [1, 11]:

$$\begin{cases} \dot{c} &= \omega(\xi - c) \\ \dot{\xi} &= \omega(\xi - z) \\ \xi &= c + \frac{\dot{c}}{\omega} \end{cases} \quad (2)$$

with  $z, \xi$  respectively the Zero Moment Point (ZMP) and DCM and  $\omega = \sqrt{g/c_z}$ . 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]:

$$\begin{aligned} z^* &= z^{ref} - \left[1 + \frac{k_{p_{dcm}}}{\omega}\right] (\xi^{ref} - \xi) \\ &+ \frac{k_{i_{dcm}}}{\omega} (z^{ref} - z) - \frac{k_{i_{dcm}}}{\omega} \int (\xi^{ref} - \xi) \end{aligned} \quad (3)$$

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]:

$$\ddot{c}^* = \ddot{c}^{ref} + k_{p_{adm}}(z - z^*) \quad (4)$$

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  $h_c = [l_c \ k_c]^T \in \mathbb{R}^6$ . In [18], the task formulation of the centroidal dynamics control is given by  $h_c = A_G(q)\dot{q}$

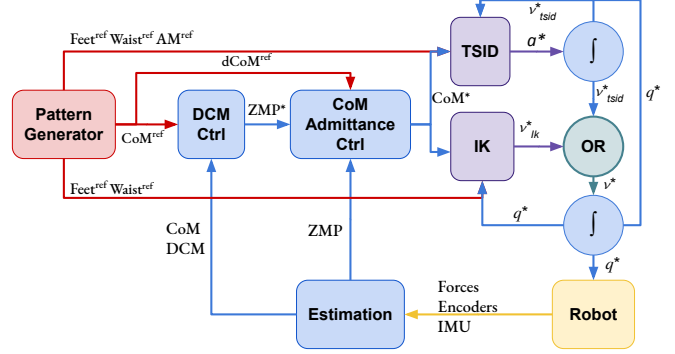


Fig. 2: Position control schemes: IK and TSID. The *OR* block is used to activate only one controller at a time.

where  $q, \dot{q}$  are the joint position and velocity vectors of the robot and  $A_G$  is the Centroidal Momentum Matrix [14].

The tasks dynamics are given by the following equations:

$$\begin{cases} \dot{l}_c &= m[\ddot{c}^* + K_{Dcom}(\dot{c}^* - \dot{c}) + K_{Pcom}(c^* - c)] \\ \dot{k}_c &= \dot{k}_c^* + K_{Pam}(k_c^* - k_c) \end{cases} \quad (5)$$

The angular momentum task in TSID is expressed as in the equation Eq. 5, successfully implemented in [10] (the gains are defined in Table I).

## III. WHOLE-BODY CONTROLLER

### 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  $e$  to be 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. And the task errors  $e$  are expressed as:

$$\begin{aligned} \dot{e}(q, t) &= \dot{x}(q) - \dot{x}^*(t) \\ \dot{x}(q) &= J\dot{q} \end{aligned} \quad (6)$$

with  $J = \frac{\partial e}{\partial q} = \frac{\partial x}{\partial q}$  the Jacobian according to the robot state vector.

The following dynamics is imposed on these errors:

$$\begin{aligned} \dot{e}(q, t) &= K_P(x(q) \ominus x^*(q)) \\ \Leftrightarrow \dot{x}(q) &= \dot{x}^*(t) + K_P(x(q) \ominus x^*(q)) \end{aligned} \quad (7)$$

with  $\ominus$  the difference operator of Lie group.

*Inverse Kinematics QP: IK* - This control scheme is based on a DCM controller (Eq.3), a CoM admittance controller (Eq.4) and a Lexicographic QP 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 QP 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 derivable function  $x$  of  $q$ . And the task errors  $e$  are expressed as:

$$\begin{aligned}\ddot{e}(q, t) &= \ddot{x}(q) - \ddot{x}^*(t) \\ \dot{e}(q, t) &= (J\ddot{q} + \dot{J}\dot{q}) - \dot{x}^*(t)\end{aligned}\quad (8)$$

The following dynamics is imposed on these errors:

$$\begin{aligned}\ddot{e}(q, t) &= K_P(x(q) \ominus x^*) + K_D(\dot{x}(q) - \dot{x}^*(t)) \\ \Leftrightarrow \ddot{x}(q) &= \ddot{x}^*(t) + K_P(x(q) \ominus x^*(t)) + \\ &K_D(\dot{x}(q) - \dot{x}^*(t))\end{aligned}\quad (9)$$

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.3), a CoM admittance controller (Eq.4) 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 0, 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  $2 \times 10^{-2}$ )

The respective task gains are defined in Table I. 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 I).

## 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

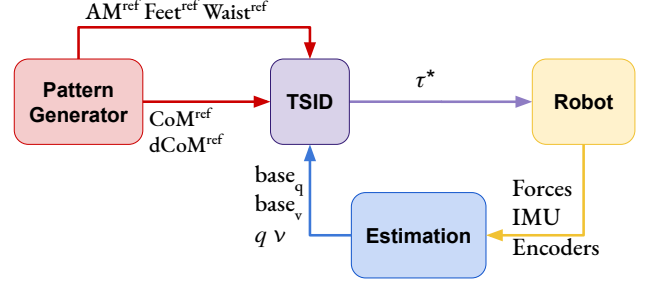


Fig. 3: TSID torque control scheme.

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  $E_{m+}$ .

$$C_{et} = \frac{E_m}{mgD} \quad C_{mt} = \frac{E_{m+}}{mgD} \quad E_m = \int_0^T \sum_{i=0}^N |\tau_i(t) \omega_i(t)| dt \quad (10)$$

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:

$$PGM = 1 - \frac{RMS(\tau_{sa})}{RMS(\tau_{tot})} \quad (11)$$

$$RMS(\tau_{tot}) = \sqrt{\frac{\int_0^T \left[ \sum_{i=0}^N \tau_i(t)^2 \right] dt}{T}} \quad (12)$$

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 foot steps (pre-defined in these simulations). This implementation uses

Tasks Gains	IK (20cm stairs)	TSID position (20cm stairs)	TSID torque (20-60cm stairs)
$K_{pcom}$	100	1000	20 12
$K_{dcom}$	-	300	3
$K_{pcomH}$	100	1000	-
$K_{dcomH}$	-	300	-
$K_{pwaist}$	300	100	100
$K_{dwaist}$	-	20	20
$K_{pcontacts}$	1000	30	30-100 30
$K_{dcontacts}$	-	11	11-0 11
$K_{pfeet}$	1000	2000	1200 500
$K_{dfeet}$	-	20	12
$K_{pam}$	-	10	10
$K_{pposture}$	100	see below	see below
$K_{dposture}$	-	$2\sqrt{K_{pposture}}$	$2\sqrt{K_{pposture}}$
$K_{pcomAdm}$	15 45	12	-
$K_{pdcM}$	8 25	8	-
$K_{1dcm}$	1	1	-
$K_{zdcM}$	1	1	-
TSID Gains		Legs	Torso
$K_{pposture}$	[10, 5, 5, 1, 10, 10]	[100, 100]	
	Arms	Head	
$K_{Dposture}$	[50, 10, 10, 10, 50, 10, 10, 10]	[100, 100]	

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

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 limbs motion. The desired DCM  $\xi^*$  is deduced from the desired CoM  $c^*$  and desired ZMP  $z^*$  trajectories (see Eq.2).

### B. Multicontact-locomotion-planning

The trajectories used in the tilted platforms and stairs simulations have been computed using the open-source framework *multicontact-locomotion-planning* [31]. Given the initial and final poses of the robot, the framework computes a reachability plan and a contacts sequence as in [32]. Then it optimizes 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. 1, 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>. 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).

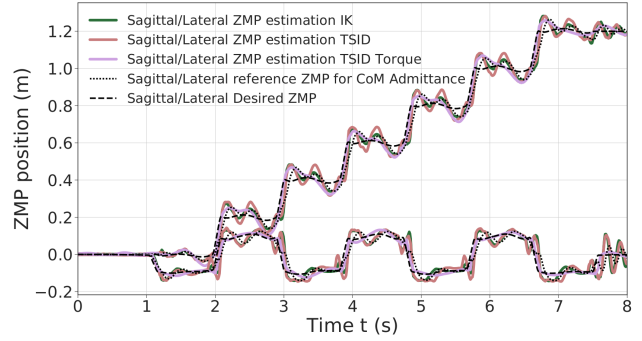


Fig. 4: ZMP estimation of the 20 cm step walk.

Control Scheme	Axis	Average	Standard deviation	Peaks
$IK$	x-axis	0.019m	0.022m	0.131m
	y-axis	0.022m	0.026m	0.150m
$TSID$	x-axis	0.028m	0.025m	0.142m
	y-axis	0.025	0.027m	0.138m
$position$	x-axis	0.026m	0.021	0.078m
	y-axis	0.011m	0.014m	0.078m

TABLE II: ZMP error of the 20 cm step walk simulation.

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, for the torque scheme, the estimated ZMP is compared to the desired ZMP (from the planning). In particular, in the lateral plane, the error is quite low, 1cm in average.

The Fig. 5 illustrates the ground impacts problem in position control compared to the better foot landing observed in torque control. Indeed, each time the left foot comes into contact with the ground (1.5s, 3.5s,...), the  $IK$  and  $TSID$  *position* schemes show peaks in the foot force ( $\sim 400N$ ) which are avoided in  $TSID$  *torque*. This explains also the peaks in the ZMP errors (around 15cm) because during an impact the foot bounces on the ground. The force oscillations of the  $IK$  and  $TSID$  *position* controllers when the foot is in the air are due to the high control gains on the ankle (Proportional-Integral-Derivative (PID) gains of the low-level position control in Gazebo), it is mainly noises.

### 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

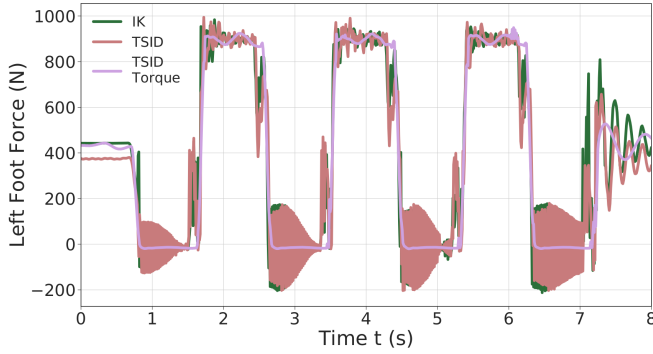


Fig. 5: Z-axis left foot force of the 20 cm step walk.

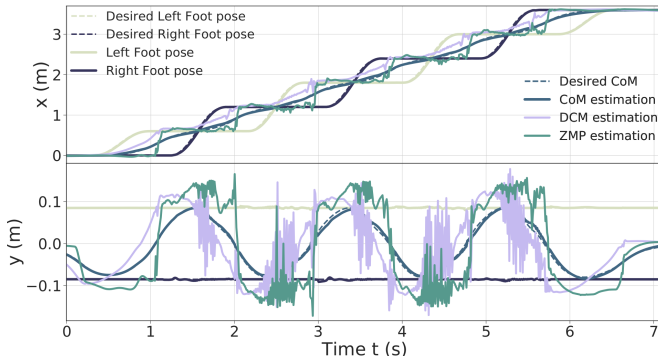


Fig. 6: Feet, CoM, DCM and ZMP of the 60 cm step walk.

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. 60cm).

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 6mm) however, along the x-axis, they show some delay (maximum error of 6cm). Thus, it induces greater tracking errors on the x-axis for the CoM (peaks of 5cm along the x-axis and 1.5cm 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 60cm 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.

	Axis	Average	Standard deviation	Peaks
CoM	x-axis	0.018m	0.013m	0.050m
	y-axis	0.004m	0.003m	0.015m
Left Foot	x-axis	0.014m	0.013m	0.063m
	y-axis	0.001	0.001m	0.005m
Right Foot	x-axis	0.016m	0.016	0.063m
	y-axis	0.001m	0.001m	0.006m

TABLE III: CoM and Feet error of the 60 cm step walk.

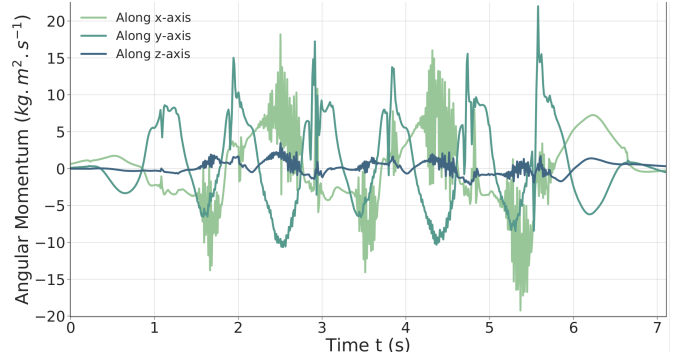


Fig. 7: AM behaviour during the 60 cm step walk in torque.

Figure 8 illustrates the tracking performance of the controllers. The ones in position present the largest oscillations as *TSID torque* is the most stable (see Table IV). Both the *IK* and the torque control show oscillations at  $t \approx 18s$ ; it corresponds to the worst case where the robot has its two feet tilted to keep its balance on two opposite platforms leading to small slippages of the feet (this behavior can be observed in the linked video). These oscillations are larger in the case of the *IK* scheme. Similar oscillations on the contact forces in this part of the motion have also been observed, which are smaller in the case of the torque control. Increasing the gains on the feet only generates more instability, but raising the ones on the DCM and admittance control lessen the oscillations (at the cost of a more rigid behavior).

Finally the same result on the feet forces is obtained in this simulation with respect to the 20cm steps one. Due to the high gains on the DCM, to avoid the slippage of the robot, the *IK* control presents bigger peaks of force.

### D. Climbing Stairs

In the last simulation the robot is climbing 6 stairs of 10cm height and 30cm 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 1cm, 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

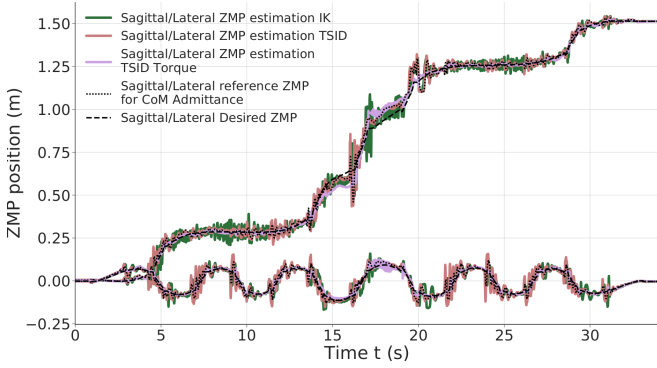


Fig. 8: ZMP estimation of the *tilted platforms* simulation.

Control Scheme	Axis	Average	Standard deviation	Peaks
<i>IK</i>	x-axis	0.021m	0.024m	0.278m
	y-axis	0.016m	0.018m	0.118m
<i>TSID</i>	x-axis	0.012m	0.017m	0.197m
<i>position</i>	y-axis	0.015	0.019m	0.127m
<i>TSID</i>	x-axis	0.013m	0.021	0.107m
<i>torque</i>	y-axis	0.005m	0.006m	0.058m

TABLE IV: ZMP error of the *tilted platforms* simulation.

also presented for comparison (computed using Eq.10), as the human ones. The lower the energy consumption is, the better, and similarly, getting closer to the human cost of transport is an improvement.

Compared to the results obtained on iCub, the control in torque has a similar cost for the 20cm steps simulation. However, the cost of the position controllers presented in this paper is higher, because of their higher gains. The human efficiency is closer to the torque control, walking with a  $C_{et}$  around  $0.2J/kg/m$  [34]. Noticeably, the energy costs in torque for the *tilted platforms* and *stairs* trajectories are still less important than the simpler walk in position; the  $C_{mt}$  never exceeds 1, even for the 60 cm walk. Overall, the controller *TSID position* consumes less energy than the *IK*.

The Passivity Gait Measure comparison of the different simulations is reported in Table VII for three gait stages: Single Support (Single S. corresponding to the stance ankle), Double Support (Double S.) and Flying Foot (Flying F. where the foot has no contact with the ground). The human results is given as an indicator [25], the robot behavior is expected to be similar during double support and flying foot phase where the ankle should be passive.

The results of the position control schemes show a behavior which is the opposite of the human one. The passivity of the ankle is higher during the stance phase because of the control of the ZMP which minimizes the ankle torque. And it is weaker during the double support and flying phases, due to the high PID gains of the low-level position control.

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 ( $PGM \sim 0.9$ ) 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

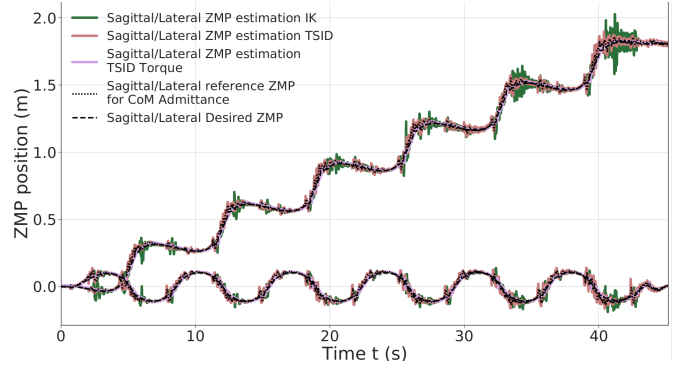


Fig. 9: ZMP estimation of *stairs climbing*.

Control Scheme	Axis	Average	Standard deviation	Peaks
<i>IK</i>	x-axis	0.022m	0.026m	0.257m
	y-axis	0.015m	0.017m	0.151m
<i>TSID</i>	x-axis	0.009m	0.013m	0.151m
<i>position</i>	y-axis	0.012	0.015m	0.119m
<i>TSID</i>	x-axis	0.008m	0.006	0.049m
<i>torque</i>	y-axis	0.006m	0.005m	0.047m

TABLE V: ZMP error of the *stairs* simulation.

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

Control Scheme	Simulation	$E_m$ [J]	$E_{m+}$ [J]	$C_{et}$ [J/kg/m]	$C_{mt}$ [J/kg/m]
<i>Human</i>	-	-	-	0.2	0.05
<i>iCub</i>	-	-	-	-	-
<i>position</i>	20cm	-	-	-	0.49
<i>torque</i>	20cm	-	-	-	0.26
<i>IK</i>	20cm	1983.9	1359.3	1.68	1.15
	platforms	5418.7	3769.2	3.7	2.6
	stairs	7249.5	2145.3	4.1	1.2
<i>TSID</i>	20cm	2324.5	764.1	1.97	0.65
<i>position</i>	platforms	5377.5	1413.6	3.6	2.0
	stairs	6812.6	2059.6	3.8	1.2
	20cm	521.8	259.3	0.44	0.22
<i>TSID</i>	60cm	3147.2	1583.8	0.89	0.45
<i>torque</i>	platforms	1378.6	668.5	0.93	0.45
	stairs	1861.1	1205.5	1.1	0.68

TABLE VI: Results of the specific cost of transport.

	Simulation	Double S.	Single S.	Flying F.
<i>Human</i>	50cm	1.0	0.6	~ 1.0
	20cm	0.35	0.89	0.24
<i>IK</i>	platforms	0.27	0.85	0.31
	stairs	0.46	0.86	0.36
<i>TSID</i>	20cm	0.37	0.74	0.37
<i>position</i>	platforms	0.27	0.86	0.30
	stairs	0.55	0.86	0.34
	20cm	0.93	0.87	1.0
<i>TSID</i>	60cm	0.87	0.79	1.0
<i>torque</i>	platforms	0.87	0.8	0.91
	stairs	0.97	0.89	1.0

TABLE VII: Results of the PGM on three gait stages.

to unrealistic and dangerous behaviors on the real robot. The authors are currently testing the simulations on the simulator of the TALOS constructor, PAL robotics, which models the actuator dynamics of the robot. To do so, the torque control scheme is plugged to the constructor low-level controller, which computes a new command respecting the actuator dynamics. Only some tuning of the task gains seems to be needed to achieve good results.

Another concern is that the real robot is subject to imperfections such as errors in the actuation chain model, sensor noise, limited torque bandwidth or delays. In the presented implementation some actions against the possible problems have been implemented [35], such as adding filters on the signals retrieved from the robot and using the force sensors on the feet to improve the robustness of the base-estimator. Moreover, a Proportional-Derivative controller with a feed-forward term (PD+) has been implemented for the torque controller, to stiff the ankles, improving the quality of the feet landing. However, in simulation, this PD+ only increases the rigidity because of the feedback in position and velocity, and decreases the PGM, thus it has not been presented.

For position control, the simulations are dependent of the low-level control PID of the Gazebo plugin. These gains however have been modified to fit the real behavior of the motors. Thus, the position controllers should not need important modifications to be tested on the real robot. Yet, because the robot TALOS presents a flexibility in the hips which is not measured by the encoders, the simulations cannot be entirely realized on the real robot. Indeed, this flexibility leads to errors in the landing positions of the feet which cannot be compensated without a proper identification and modelling of the flexibility. In torque control however,

Control Scheme	Simulation	20cm (60cm)	Platforms	Stairs
<i>IK</i>	Average	0.5ms	0.7ms	0.6ms
	Peaks	2ms	4ms	4ms
<i>TSID</i>	Average	1.2ms	1.2ms	1.2ms
<i>position</i>	Peaks	4.5ms	4.3ms	4.2ms
<i>TSID</i>	Average	1ms (1.4ms)	1.2ms	1.1ms
<i>torque</i>	Peaks	2.8ms (6ms)	5ms	5.5ms

TABLE VIII: Comparison of the execution time.

this issue is mitigated because the flexibility is considered by 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.

## 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

- [1] T. Takenaka, T. Matsumoto, and T. Yoshiike, "Real time motion generation and control for biped robot-1st report: Walking gait pattern generation-," in *IROG*, 2009.
- [2] S. Kajita, M. Morisawa, K. Miura, S. Nakaoka, K. Harada, K. Kaneko, F. Kanehiro, and K. Yokoi, "Biped walking stabilization based on linear inverted pendulum tracking," in *IROG*, 2010.
- [3] S. Caron, A. Kheddar, and O. Tempier, "Stair climbing stabilization of the hrp-4 humanoid robot using whole-body admittance control," in *ICRA*, 2019.
- [4] C. Samson, M. Leborgne, and B. Espiau, "Robot control. the task-function approach," in *Oxford Engineering Science Series, vol. 22*. Oxford University Press, 1991.



- [5] A. Escande, N. Mansard, and P. Wieber, "Hierarchical quadratic programming: Fast online humanoid-robot motion generation," *IJRR*, 2014.
- [6] B. Henze, A. Dietrich, and C. Ott, "An approach to combine balancing with hierarchical whole-body control for legged humanoid robots," *RAL*, 2016.
- [7] A. Herzog, N. Rotella, S. Mason, F. Grimminger, S. Schaal, and L. Righetti, "Momentum control with hierarchical inverse dynamics on a torque-controlled humanoid," *Autonomous Robots*, 2014.
- [8] T. Koolen, S. Bertrand, G. Thomas, T. de Boer, T. Wu, J. Smith, J. Englesberger, and J. Pratt, "Design of a momentum-based control framework and application to the humanoid robot atlas," *IJHR*, 2016.
- [9] R. Cisneros, M. Benallegue, A. Benallegue, M. Morisawa, H. Audren, P. Gergondet, A. Escande, A. Kheddar, and F. Kanehiro, "Robust humanoid control using a qp solver with integral gains," in *IROS*, 2018.
- [10] S. Lee and A. Goswami, "A momentum-based balance controller for humanoid robots on non-level and non-stationary ground," *Autonomous Robots*, 2012.
- [11] J. Engelsberger, C. Ott, and A. Albu-Schäffer, "Three-dimensional bipedal walking control based on divergent component of motion," *ITRO*, 2015.
- [12] N. Ramuzat, F. Forget, V. Bonnet, M. Gautier, S. Boria, and O. Stasse, "Actuator model, identification and differential dynamic programming for a talos humanoid robot," in *European Control Conference (ECC)*, 2020.
- [13] G. Romualdi, S. Daffarra, Y. Hu, P. Ramadoss, F. A. Chavez, S. Traversaro, and D. Pucci, "A benchmarking of dcm based architectures for position, velocity and torque controlled humanoid robots," *IJHR*, 2019.
- [14] D. Orin, A. Goswami, and S.-H. Lee, "Centroidal dynamics of a humanoid robot," *Autonomous Robot*, 2013.
- [15] T. Sugihara, "Standing stabilizability and stepping maneuver in planar bipedalism based on the best com-zmp regulator," in *ICRA*, 2009.
- [16] G. Mesesan, J. Engelsberger, G. Garofalo, C. Ott, and A. Albu-Schäffer, "Dynamic walking on compliant and uneven terrain using dcm and passivity-based whole-body control," in *ICHR*, 2019.
- [17] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Resolved momentum control: Humanoid motion planning based on the linear and angular momentum," in *IROS*, 2003.
- [18] P. M. Wensing and D. E. Orin, "Generation of dynamic humanoid behaviors through task-space control with conic optimization," in *ICRA*, 2013.
- [19] N. Mansard, O. Stasse, P. Evrard, and A. Kheddar, "A versatile generalized inverted kinematics implementation for collaborative working humanoid robots: The stack of tasks," in *International Conference on Advanced Robotics (ICAR)*, 2009.
- [20] GEPETTO Team LAAS-CNRS, "sot-talos-balance," <https://github.com/loco-3d/sot-talos-balance>.
- [21] —, "Tsid," <https://github.com/stack-of-tasks/tsid>.
- [22] A. D. Prete, N. Mansard, O. Ponce, O. Stasse, and F. Nori, "Implementing torque control with high-ratio gear boxes and without joint-torque sensors," *IJHR*, 2016.
- [23] GEPETTO Team LAAS-CNRS, "talos-torque-control," <https://github.com/stack-of-tasks/talos-torque-control>.
- [24] D. Torricelli, J. Gonzalez-Vargas, J. Veneman, K. Mombaur, N. Tsagarakis, A. del Ama, A. Gil-Agudo, J. Moreno, and J. Pons, "Benchmarking bipedal locomotion: A unified scheme for humanoids, wearable robots, and humans," *IEEE Robotics Automation Magazine*, 2015.
- [25] C. Mummolo and J. H. Kim, "Passive and dynamic gait measures for biped mechanism: formulation and simulation analysis," in *Robotica*, 2012.
- [26] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by using preview control of zero-moment point," in *ICRA*, 2003.
- [27] O. Stasse, B. Verrelst, P.-B. Wieber, B. Vanderborght, P. Evrard, A. Kheddar, and K. Yokoi, "Modular architecture for humanoid walking pattern prototyping and experiments," *Advanced Robotics, Special Issue on Middleware for Robotics –Software and Hardware Module in Robotics System*, 2008.
- [28] GEPETTO Team LAAS-CNRS, "jrl-walkgen," <https://github.com/stack-of-tasks/jrl-walkgen>.
- [29] R. Featherstone, *Rigid Body Dynamics Algorithms*. Springer, 2008.
- [30] J. Carpentier, G. Saurel, G. Buondonno, J. Mirabel, F. Lamiroux, O. Stasse, and N. Mansard, "The pinocchio c++ library – a fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives," in *Int. Symp. on System Integrations*, 2019.
- [31] GEPETTO Team LAAS-CNRS, "multicontact-locomotion-planning," <https://github.com/loco-3d/multicontact-locomotion-planning>.
- [32] S. Tonneau, D. Song, P. Fernbach, N. Mansard, M. Taïx, and A. D. Prete, "SLIM: Sparse L1-norm Minimization for contact planning on uneven terrain," in *ICRA*, 2020.
- [33] B. Ponton, A. Herzog, S. Schaal, and L. Righetti, "On time optimisation of centroidal momentum dynamics," *ICRA*, 2018.
- [34] S. Collins, A. Ruina, R. Tedrake, and M. Wisse, "Efficient bipedal robots based on passive-dynamic walkers," *Science*, 2005.
- [35] J. Engelsberger, G. Mesesan, A. Werner, and C. Ott, "Torque-based dynamic walking - a long way from simulation to experiment," in *ICRA*, 2018.
- [36] N. A. Villa, P. Fernbach, N. Mansard, and O. Stasse, "Addressing flexibility in biped locomotion with robust control and closed-loop model-predictive control," in *ICRA*, submitted in 2022.