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

N Ramuzat, G Buondonno, S Boria, Olivier Stasse

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

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

***Abstract*—This paper presents a comparison of three control schemes applied on the commercially available TALOS Humanoid Robot. The aim is to highlight the advantages and drawbacks of each model applied on three locomotion problems: walking on flat and non-flat terrain and climbing stairs. The different models are based on position control (first and second models) or torque control (third model). The first one uses a hierarchical quadratic program at velocity level. The second one employs a weighted quadratic program named Task Space Inverse Dynamic (TSID) at acceleration level. Finally, the last one also uses TSID but at torque level. The controllers performances are compared in simulation, using Gazebo, on the accuracy of their tracking, their energy consumption and their computational time execution.**

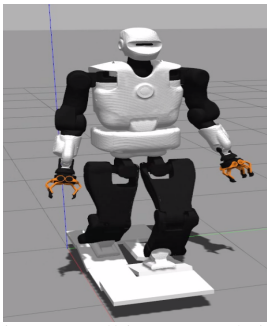
Fig. 1: Walking on Debris.

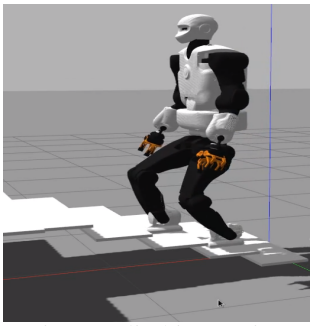
Fig. 2: Climbing Stairs.

INTRODUCTION

Bipedal locomotion of humanoid robots still offers challenges, because of the complexity of the robots dynamics, the numerous constraints of the motion (contact forces, actuation bounds...) and the unknown environment. To achieve the locomotion of such complex systems, an approach is to decompose the problem in three stages: contacts sequence generation, trajectory planning and whole-body control.

Most of the trajectory planning methods use the centroidal dynamics to generate consistent behaviors for a legged robot. In addition, the concepts of Divergent Component of Motion (DCM) [1] and Capture Point (CP) [2] associated to reduced dynamics models such as the Linear Inverted Pendulum (LIPM) [3] allow to simplify the trajectory generation.

Real-time whole-body controllers compute a control vector stabilizing the reference trajectories obtained from the previous stages. The reference trajectories are designed in the tasks space and complex motions combine several nonlinear tasks and constraints. The simultaneous execution of tasks can be achieved in two ways: by setting respective weights between the tasks [4, 5, 6], or by imposing a strict hierarchy [7, 8, 9]. Quadratic Programs are instantaneous optimization techniques used to solve such nonlinear problems, employing the whole-body kinematics or dynamics of the robot.

The DCM approach is also used in the control stage, associated to admittance controls on the linear part of the center of mass (CoM) and on the ankles to reach enough compliance to walk on non-flat terrain [10, 11]. This method was mostly used by position-controlled robots which gives, in general, the fastest walk for electrically actuated robots, until a recent publication [11]. Torque control needs model-based controllers tuned for each motor, achieved by modeling the actuators of the robot [12]. However, position control cannot meet the intrinsic compliance of the torque control ones. Torque control is more suitable for interactions with humans and also for multi-contact problems where external interactions and several contact points are needed. The approach using the AM dynamics in control has shown great results for walking in torque, even on non-flat terrain [4, 9, 13].

This paper intends to follow the benchmarking of humanoid robots control architectures [14]. It contributes toward the 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 Hierarchical Quadratic Program using Inverse Kinematics (called IK), the second and the third one use a Weighted Quadratic Program (WQP) with Inverse Dynamics and an Angular Momentum (AM) regularization task (called respectively TSID position and TSID torque). They are evaluated in Gazebo simulations on three locomotion problems, walking on flat and non-flat terrain and climbing stairs, on the criteria of trajectory tracking, energy consumption, passivity and computational cost. Furthermore, the proposed torque control implementation allows the TALOS robot to reach a $0.6m/s$ walk on flat floor.

Section I presents the methodologies used to plan the contacts sequence and the trajectories. Section II recalls the centroidal dynamics equations, the DCM/ZMP control and the AM task. Section III details the three whole-body control schemes. Section IV analyses the simulations results.

The authors want to thank Pierre Fernbach for his help with the multicontact-locomotion-planning framework.
This work is supported by the cooperation agreement ROB4FAM.

I. Locomotion Planning

A. Walking Pattern Generator

The trajectories used in the straight walk experiments have been computed using the algorithm described in [15, 16, 17]. It 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 experiments). This implementation uses the LIPM simplified model and the dynamic filter proposed in [15] computed with the Recursive Newton-Euler Algorithm [18] proposed by the Pinocchio library [19]. The CoM trajectory is modified to take into account the momentum generated by the limbs motion. The desired DCM ξ^* is deduced from the desired CoM c^* and desired ZMP z^* trajectories.

B. Multicontact-locomotion-planning

The trajectories used in the platforms and stairs experiments have been computed using the open-source framework *multicontact-locomotion-planning* [20]. Given the initial and final poses of the robot, the framework computes a reachability plan and a contacts sequence as in [21]. Then it optimizes the centroidal dynamics (see Section II) using two convex relaxations based on trust regions [22]. 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).

II. Centroidal Dynamics

The under-actuated part of the whole-body dynamics of a robot 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 [23]:

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

with $\dot{L} = \sum_k [R_k I_k \dot{w}_k - R_k (I_k w_k)_\times w_k]$ and $g = [0,\ 0,\ -9.81]^T$, where R_k is the orientation of the k^{th} body, 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. l_c and $k_c \in \mathbb{R}^3$ are respectively the linear and angular momentum around the CoM.

A. Divergent Component of Motion

Assuming that the CoM evolves on a plan $\ddot{c}_z = 0$, the problem can be simplified as a LIPM [3]. This formulation allows to decompose the CoM dynamics between its stable and unstable parts. This unstable part is called the DCM [1] and is defined by $\xi = c + \frac{\dot{c}}{\omega}$, with $\omega = \sqrt{g/c_z}$. It can be controlled to stabilize the system by applying a PI on the DCM [3, 24, 1, 10]. In terms of Zero Moment Point (ZMP) [10], the obtained control law is:

$$z^{ref} = z^* - \left[1 + \frac{kp_{dcm}}{\omega}\right](\xi^* - \xi) - \frac{ki_{dcm}}{\omega}\int(\xi^* - \xi) + \frac{kz_{dcm}}{\omega}(z^* - z) \tag{2}$$

with z, ξ the estimated ZMP and DCM and z^*, ξ^* their respective desired values given by the planning. Finally, this ZMP reference is used into a CoM admittance control as [25]:

$$\ddot{c} = \ddot{c}^* + kp_{adm}(z - z^{ref}) \tag{3}$$

The two position control schemes presented in this paper use this stabilization formulation.

B. Centroidal Momentum Tasks

The objective is to consider the angular momentum part generated by the contact transition [26]. Using the equation Eq. 1, the centroidal dynamics is therefore defined by $h_c = [l_c\ k_c]^T \in \mathbb{R}^6$. In [27], the task formulation of the centroidal dynamics control is given by $h_c = A_G(q)\dot{q}$ where $q, \dot{q}$ are the joint position and velocity vectors of the robot and A_G is the Centroidal Momentum Matrix [23].

The tasks dynamics are given by the following equations:

$$\begin{cases} \dot{l}_c & = m\left[\ddot{c}^* + kd_{com}(\dot{c}^* - \dot{c}) + kp_{com}(c^* - c)\right] \\ \dot{k}_c & = \dot{k}_c^* + kp_{am}(k_c^* - k_c) \end{cases} \tag{4}$$

with $\dot{l}_c$ and $\dot{k}_c$ the commanded rates of change of the linear momentum and angular momentum. The angular momentum task in TSID is expressed as in the equation Eq. 4, successfully implemented in [13]. Another choice is to express the robot dynamics equation at the centroidal momentum level in the QP [4].

III. Whole-body controller: Quadratic Program

A. Hierarchical Quadratic Program

The first controller used is a Hierarchical Quadratic Program (HQP) task-based inverse kinematics described in [28]. The task errors are implemented as a velocity-based tracking law between the desired feature s^* and the feature s : $\dot{s} = \dot{s}^* + K_P(s \ominus s^*)$.

Inverse Kinematics HQP: IK - This control scheme is based on a DCM controller, a CoM admittance controller and a HQP solving the inverse kinematics of the robot (see Fig. 3). Its implementation is open-source [29].

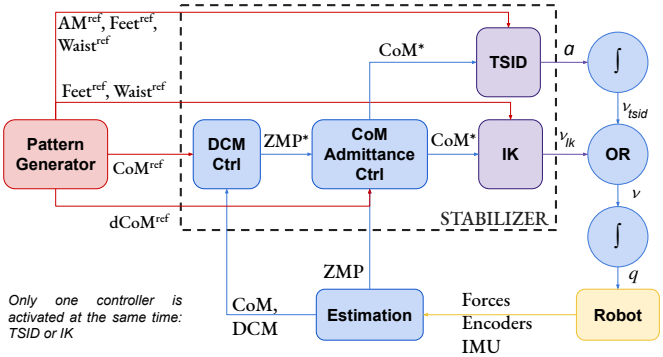

Fig. 3: Position control schemes: IK and TSID.

The tasks used during the experiments are:

- Feet contact(s) and feet tracking (same task, priority 0)

- CoM height tracking (priority 1)
- CoM lateral-sagittal tracking (priority 2)
- Waist orientation (priority 3)
- Posture regularization in half-sitting (priority 4)

The whole body controller provides a desired velocity/position for each actuator, $\dot{q}^*, q^*$. However, the actual control variable is the current i_j for each motor j, then, the position control is done by a PID on the system:

$$i_j = P_j(q_j^* - q_j) + I_j \int_0^t (q_j^*(t) - q_j(t))dt + D_j(\dot{q}^* - \dot{q}) \quad (5)$$

Proportional gains are very high (20000 for the legs) and I_J are low (5 for the legs). This makes the robot extremely rigid, but it is simple to deploy on the real robot.

B. Task Space Inverse Dynamics (TSID)

TSID [30] is a Weighted Quadratic Program (WQP) which sums selected task functions in the cost using weights to define their priority. The task errors are implemented as an acceleration-based tracking law: $\ddot{s} = \ddot{s}^* + K_P(s - s^*) + K_D(\dot{s} - \dot{s}^*)$. It solves the inverse dynamics of the robot in rigid contact with the environment [9] and have been successfully used on HRP2 robot in [31].

Inverse Dynamics WQP: TSID Position - This control scheme is based on a DCM controller, a CoM admittance controller and a WQP solving the inverse dynamics of the robot (with an angular momentum regularization task), see Fig. 3. Its implementation is in the same package than the controller TSID Torque.

The tasks considered during the experiments are:

- Feet tracking (constraint, weight 1)
- Feet contact(s) (constraint, weight $1e^{-3}$)
- CoM height tracking (cost, weight $1e^{10}$)
- CoM lateral-sagittal tracking (cost, weight $1e^{10}$)
- Waist orientation (cost, weight 1)
- Posture regularization in half-sitting (cost, weight 0.1)
- AM velocity-acceleration tracking (cost, weight $2e^{-2}$)

The motor current is computed similarly as in the IK scheme.

Inverse Dynamics WQP: TSID Torque - This control scheme is based on a WQP solving the inverse dynamics of the robot (with an angular momentum regularization task), as shown in Fig. 4. Its implementation is open-source [32].

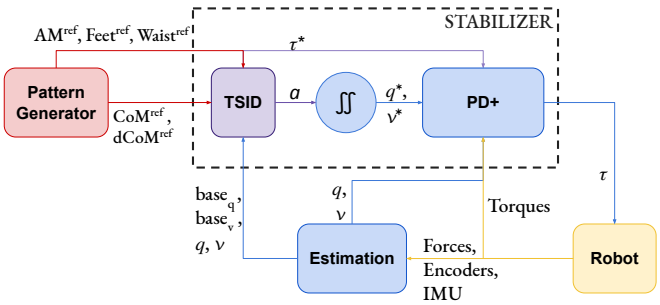

Fig. 4: TSID torque control scheme.

The tasks considered in the experiments are:

- Feet tracking (constraint, weight 1)

Tasks Gains	IK (20cm|debris)	TSID position (20cm|debris)	TSID torque (20-60cm|debris)
kp_{com}	100	1000	20|12
kd_{com}	-	300	3
kp_{comH}	100	1000	-
kd_{comH}	-	300	-
kp_{waist}	300	100	100
kd_{waist}	-	20	20
$kp_{contacts}$	1000	30	30-100|30
$kd_{contacts}$	-	11	11-0|11
kp_{feet}	1000	2000	1200|500
kd_{feet}	-	20	12
kp_{am}	-	10	10
$kp_{posture}$	100	see below	see below
$kd_{posture}$	-	$2\sqrt{kp_{posture}}$	$2\sqrt{kp_{posture}}$
kp_{comAdm}	15|45	12	-
kp_{dcm}	8|25	8	-
ki_{dcm}	0	1|0	-
kz_{dcm}	0	1	-
$gamma_{dcm}$	0.2	0.2	-

TABLE I: Tasks gains of the control schemes.

TSID Gains	Legs	Torso
$kp_{posture}$	[10, 5, 5, 1, 10, 10]	[100, 100]
P	[1300, 1300, 1300, 1300, 50, 50]	[100, 100]
D	[20, 20, 20, 20, 0, 0]	[10, 10]
kp_{meas}	[2.5, 2.5, 2.5, 2.5, 7, 7]	[2.5, 2.5]]
TSID Gains	Arms	Head
$kp_{posture}$	[50, 10, 10, 10, 50, 10, 10, 10]	[100, 100]
P	[500, 500, 500, 500, 500, 500, 500, 10]	[10, 10]
D	[5, 5, 5, 5, 5, 5, 5, 0.1]	[0.1, 0.1]
kp_{meas}	[2.5, 2.5, 2.5, 2.5, 2.5, 0, 0, 0]	[0, 0]

TABLE II: Gains of the posture task and the PD+ in TSID.

- Feet contact(s) (constraint, weight $1e^{-3}$)
- CoM tracking (cost, weight $1e^{6}$)
- Waist orientation (cost, weight 1)
- Posture regularization in half-sitting (cost, weight 0.1)
- AM tracking (cost, weight $2e^{-2}$)

The whole body controller provides a desired torque for each joint, τ^*. The actuator is then controlled through feedback [33], using a PD+:

$$\tau_{pd} = \tau^* + P(q - q^*) + D(\dot{q} - \dot{q}^*) \quad (6)$$

$$\tau_{pd} = \tau_{pd} + kp_{meas}(\tau - \tau_{pd}) \quad (7)$$

with τ^* the QP desired torque, τ the measured torque, $q, \dot{q}$ the estimated position/velocity and $q^*, \dot{q}^*$ obtained from the integration of the desired acceleration. There is a nonlinear relationship between the joint torque and the motor current, but in simulation it is considered as linear.

IV. Simulation Results

The experiments realized in this paper have been made using Gazebo on a standard laptop (Intel CPU i7-8850H @ 2.6 GHz). The TALOS robot is an humanoid robot of 1.75m tall and 100kg which is composed of 32 joints and the floating-base (6Degree-of-Freedom).

A. Tasks gains of the control schemes

The task gains used by the different control schemes are presented in this section (see Table I, *debris* and *stairs* simulations uses the same gains). For the TSID posture task and the PD+ the gains depend on each joint (see Table II).

B. *Straight walk of* $20cm$

In the simulation, the robot executes 6 steps forward at $0.2m/s$ and a final step to go back to the initial position (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 $\sim 0.20m$). The controllers have also been successfully tested with steps of $0.711s$ for single support phase and $0.089s$ for double support phase. The Fig. 5 and Fig. 6 present a comparison of the three control schemes on the tracking of the ZMP (Fig. 5) and on the forces applied on the ground along the z-axis (on the left foot, Fig. 6). The tracking of the CoM and the feet are accurately followed by the three controllers.

The two position controllers achieve similar results, tracking correctly the ZMP reference of Eq. 2 (and thus the DCM). However the results of TSID is slightly more oscillating. The torque control also tracks the desired ZMP in Fig. 5 even though there is no explicit control n the ZMP or the DCM. Noticeably, in the sagittal plane it naturally shapes the reference ZMP found by the position schemes.

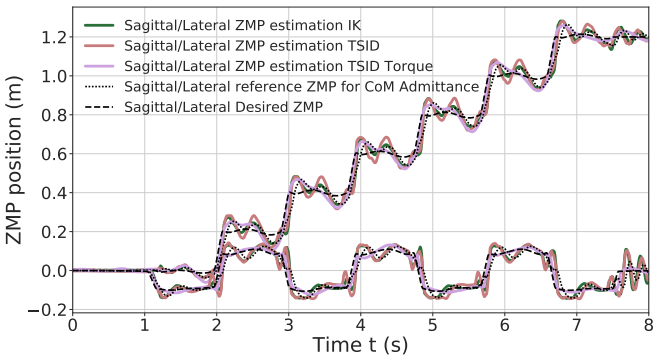

Fig. 5: ZMP tracking comparison of the $20cm$ walk.

The Fig. 6 illustrates the well-known 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. One can notice the force oscillations of the IK and TSID position controllers when the foot is in the air ($\sim 0N$), due to the high control gains on the ankle (PID gains of the low-level position control). Because the experiments are performed in simulation, the PD+ of the torque controller has almost no effect because the commanded torques are perfectly applied. Moreover, it introduces rigidity in the control because of the feedback in position and velocity. Thus, in the following simulations it has been deactivated. Note that it could be required during real experiments on the robot.

C. *Straight walk of* $60cm$ *in torque*

Until the publication of Mesesan et al. [11], electrically actuated torque control robots did not achieve results similar to the position ones during locomotion. In [11] the humanoid robot TORO successfully performed a walk on flat terrain with a step length of $55cm$ (single/double support time of

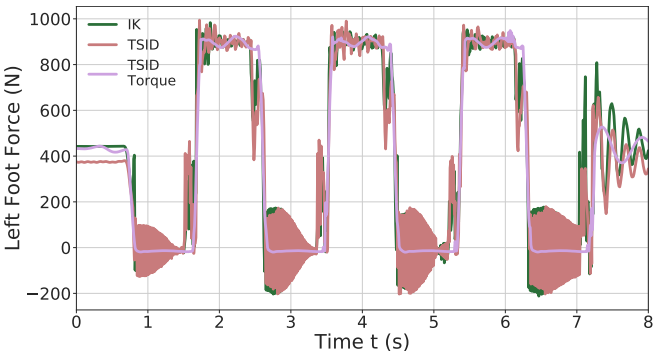

Fig. 6: Z-axis left foot force comparison of the $20cm$ walk.

$1.1/0.4s$). In the following simulation, the torque controller is pushed to its limits to show its capability to achieve a similar result. The TALOS robot 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 $\sim 60cm$).

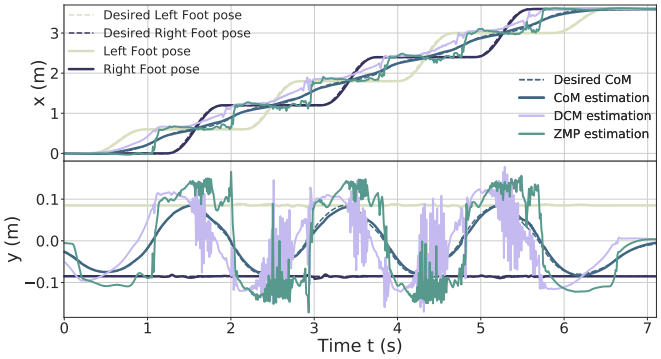

Fig. 7: Feet, CoM, DCM, ZMP tracking of the $60cm$ torque walk.

Figure 7 presents the results obtained on the tracking of the feet and the CoM; the ZMP and DCM estimations. The feet tracks well the desired trajectories along the y-axis (maximum error of $4mm$) however, along the x-axis, they show some delay due to the latency introduced by the switch of tasks (from contact to motion task, maximum error of $6cm$). Thus, it induces greater tracking errors on the x-axis for the CoM too (maximum of $3.7cm$ along the x-axis and $1.3cm$ along the y-axis). Nevertheless, the walk is successfully performed thanks to the AM task which stabilizes the robot (without the AM task it falls). 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 oscillations on the ground when landing).

In Fig. 8, 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.

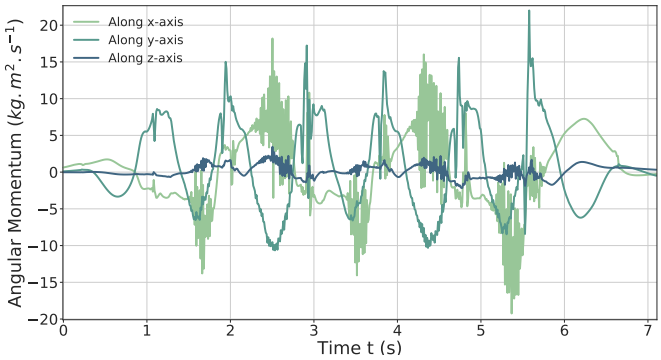

Fig. 8: AM behaviour during the $60cm$ steps walk in torque.

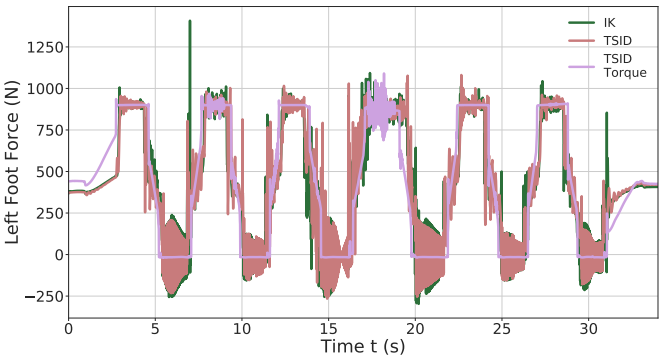

Fig. 10: Z-axis left foot force comparison on debris.

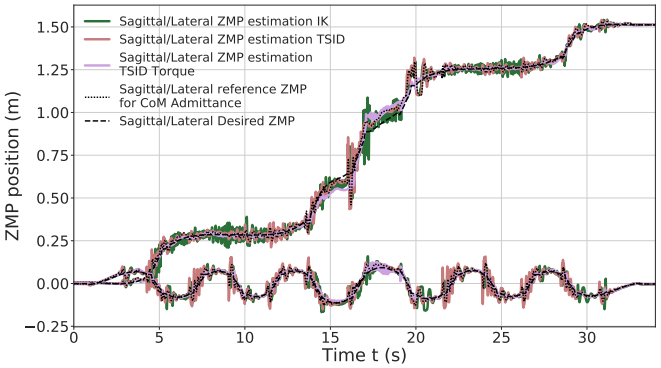

Fig. 9: ZMP tracking comparison of the walk on debris.

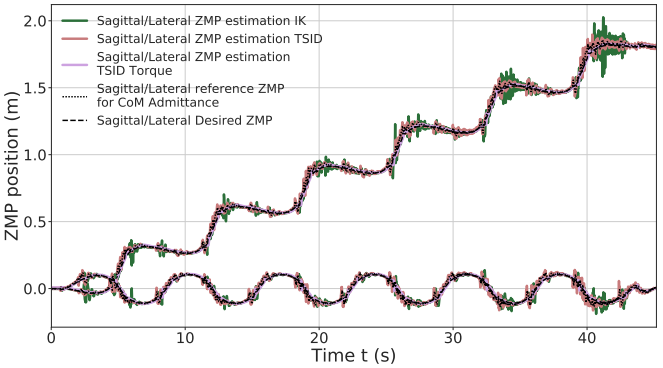

Fig. 11: ZMP tracking comparison of stairs climbing.

D. Walk on the platforms: Debris

In this third simulation, the robot walks on tilted platforms which represent debris (Fig. 2). This walk is achieved using the multicontact-locomotion-planning trajectories (see Section I-B). It ensures the stability of the controllers on non-flat terrain when the ankles are tilted.

Figure 9 illustrates the tracking performance of the controllers. The ones in position present the largest oscillations whereas TSID torque is the most stable. Both the IK and the torque control show oscillations at $t =\sim 18s$; it corresponds to the worst case where the robot has its two ankles tilted to keep its balance on two opposite debris leading to small slippages of the feet. These oscillations are larger in the case of the IK scheme because it does not separate the contact tasks and motion tasks on the feet which leads to slippage that are avoided in TSID (this behavior can be observed in the attached video). This is confirmed by the presence of similar oscillations on the contact forces in this part of the motion (see Fig. 10)), 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 helps (at the cost of a more rigid behavior).

Finally the same result on the feet forces is obtained in this simulation (see Fig. 10) with respect to the *20cm* experiment. Due to the high gains on the DCM to avoid the slippage of the robot, the IK control presents bigger peaks of force.

E. 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. 11 shows the ZMP evolution of each controllers, where the result is similar to the debris experiment. 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 attached video).

F. Energy cost and Passivity Gait Measure

The energy consumption of the three control schemes is compared using their cost of transport. This quantity 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+} [34].

$$C_{et} = \frac{E_m}{mgD} \quad C_{mt} = \frac{E_{m+}}{mgD} \quad E_m = \int_0^T |\tau(t)\,\omega(t)|dt \tag{8}$$

with m the mass of the system, g the gravity constant, D the distance traveled by the system, τ the torques applied at the robot joints and ω the velocity of the robot joints. The results of the four experiments are presented in the Table III, depending on the control scheme.

Compared to the results obtained in the paper [14] on iCub (their results have to be divided by g to be coherent), the control in torque has a similar cost for the *20cm* walk. However, the cost of the position controllers is higher, because of their higher gains. The human efficiency is closer to the torque control, walking with a C_{et} around $0.2J/kg/m$

Control Scheme	Simulation	E_{mech} [J]	$E_{mechpos}$ [J]	C_{et} [J/kg/m]	C_{mt} [J/kg/m]
	20cm	1983.9	1359.3	1.68	1.15
IK	debris	5418.7	3769.2	3.7	2.6
	stairs	7249.5	2145.3	4.1	1.2
TSID	20cm	2324.5	764.1	1.97	0.65
position	debris	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
TSID	60cm	3147.2	1583.8	0.89	0.45
torque	debris	1378.6	668.5	0.93	0.45
	stairs	1861.1	1205.5	1.1	0.68

TABLE III: Results of the specific cost of transport.

[35]. Noticeably, the energy costs in torque for the *debris* and *stairs* trajectories are still less important than the simplier walk in position. The C_{mt} in torque control never exceeds 1, even for the *60cm* walk. Overall, the controller TSID position consumes less energy than the IK.

Another interesting energetic criteria is the ability to minimize joint torques to increase the passivity of the walk[34]. The Passivity Gait Measure (PGM)[36] quantifies the passivity of a biped walking motion:

$$PGM = 1 - \frac{RMS(\tau_{sa})}{RMS(\tau_{tot})} \tag{9}$$

$$RMS(\tau_{tot}) = \sqrt{\frac{\int_0^T \left[\sum_{i=0}^{N} \tau_i(t)^2\right] dt}{T}} \tag{10}$$

with RMS the Root Mean Square of the torque applied on the stance ankle (τ_{sa}) or on each joint (τ_{tot}), N the total number of joint. The comparison is made during single support (SS or stance foot), double support (DS) and flying foot (FF) in Table IV. The human results is given as an indicator [36].

	Simulation	Double Support	Stance Foot	Flying Foot
Human	50cm	1.0	0.6	$\sim$1.0
	20cm	0.35	0.89	0.24
IK	debris	0.27	0.85	0.31
	stairs	0.46	0.86	0.36
TSID	20cm	0.37	0.74	0.37
position	debris	0.27	0.86	0.30
	stairs	0.55	0.86	0.34
	20cm PD+	0.79	0.71	1.0
TSID	20cm	0.93	0.87	1.0
torque	60cm	0.87	0.79	1.0
	debris	0.87	0.8	0.91
	stairs	0.97	0.89	1.0

TABLE IV: Results of the PGM on DS, SS and FF.

The results of the position control schemes (IK and TSID) 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. Note that there is no ankle admittance to control the ankle orientation for non-flat terrain [10]. If added, one can expect completely different PGM results, it may increase the actuation of the ankle during the stance phase.

The control scheme in torque shows much more passive behavior (except on the stance foot), with a completely passive foot during the flying phase. As said before, removing the PD+ increases the passivity, because there is no more feedback control on the position/velocity of the ankle (less rigid). Without PD+, 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 [36], where the torque controlled robot has a higher control on its stance ankle ($PGM = 0.2$).

Finally, on the *debris*, the double support phase corresponds to the worst case where the robot has its two ankles tilted to keep its balance on two opposite debris. 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 debris (highest slope), also leading to a bigger actuation of the ankle.

G. Execution time of the control schemes

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

TABLE V: Comparison of the execution time.

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 the IK 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.

CONCLUSION

Three whole-body control implementations are compared in this paper. Two of them are position based (with DCM and CoM admittance control): a HQP using inverse kinematics and a WQP using TSID with an AM task. The last is a WQP using TSID in torque with an AM task. They are evaluated in Gazebo on straight walk, walk on debris and stairs climbing; on their performances on trajectory tracking, stability, energy consumption, passivity and computational cost.

The two control schemes in position present in general the same results, with lesser 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 another 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 [4]). Nonetheless, the IK scheme has the best computational time, but in average the TSID controllers reach the $1kHz$ of control loop, necessary for real-time control.

Future works include tests and evaluations of the three controllers on the real TALOS robot.

References

[1] J. Englsberger, C. Ott, and A. Albu-Schäffer, "Three-dimensional bipedal walking control based on divergent component of motion," *IEEE Transactions on Robotics*, vol. 31, no. 2, 2015.

[2] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture point: A step toward humanoid push recovery," in *Int. Conf. on Humanoid Robotics (ICHR)*, 2006.

[3] 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 *Int. Conf. on Intelligent Robots and Systems (IROS)*, 2010.

[4] 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," *Int. Journal of Humanoid Robotics*, vol. 13, 2016.

[5] 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 *Int. Conf. on Intelligent Robots and Systems (IROS)*, 2018.

[6] K. Bouyarmane, J. Vaillant, K. Chappellet, and A. Kheddar, "Multi-robot and task-space force control with quadraticprogramming," *IEEE Transactions on Robotics*, vol. 35, no. 1, pp. 64–77, 2019.

[7] E. M. Hoffman, A. Laurenzi, L. Muratore, N. G. Tsagarakis, and D. G. Caldwell, "Multi-priority cartesian impedance control based on quadratic programming optimization," in *Int. Conf. on Robotics and Automation (ICRA)*, 2018.

[8] B. Henze, A. Dietrich, and C. Ott, "An approach to combine balancing with hierarchical whole-body control for legged humanoid robots," *IEEE Robotics and Automation Letters (RAL)*, vol. 1, pp. 700–707, 2016.

[9] 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.

[10] S. Caron, A. Kheddar, and O. Tempier, "Stair climbing stabilization of the hrp-4 humanoid robot using whole-body admittance control," in *Int. Conf. on Robotics and Automation (ICRA)*, 2019.

[11] G. Mesesan, J. Englsberger, G. Garofalo, C. Ott, and A. Albu-Schäffer, "Dynamic walking on compliant and uneven terrain using dcm and passivity-based whole-body control," in *Int. Conf. on Humanoid Robotics (ICHR)*, 2019.

[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] S. Lee and A. Goswami, "A momentum-based balance controller for humanoid robots on non-level and non-stationary ground," *Autonomous Robots*, 2012.

[14] G. Romualdi, S. Dafarra, 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," 2019.

[15] 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 *Int. Conf. on Robotics and Automation (ICRA)*, 2003.

[16] 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*, vol. 22, no. 6, pp. 589–611, 2008.

[17] GEPETTO Team LAAS-CNRS, "jrl-walkgen," https://github.com/stack-of-tasks/jrl-walkgen.

[18] R. Featherstone, *Rigid Body Dynamics Algorithms*. Springer, 2008.

[19] J. Carpentier, G. Saurel, G. Buondonno, J. Mirabel, F. Lamiraux, 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.

[20] GEPETTO Team LAAS-CNRS, "multicontact-locomotion-planning," https://github.com/loco-3d/multicontact-locomotion-planning.

[21] S. Tonneau, D. Song, P. Fernbach, N. Mansard, M. Taïx, and A. D. Prete, "SL1M: Sparse L1-norm Minimization for contact planning on uneven terrain," in *Int. Conf. on Robotics and Automation (ICRA)*, 2020.

[22] B. Ponton, A. Herzog, S. Schaal, and L. Righetti, "On time optimisation of centroidal momentum dynamics," 2018.

[23] D. Orin, A. Goswami, and S.-H. Lee, "Centroidal dynamics of a humanoid robot," *Autonomous Robot*, vol. 35, pp. 161–176, 2013.

[24] T. Sugihara, "Standing stabilizability and stepping maneuver in planar bipedalism based on the best com-zmp regulator," in *Int. Conf. on Robotics and Automation (ICRA)*, 2009.

[25] S. Kajita, H. Hirukawa, K. Harada, and K. Yokoi, *ZMP and Dynamics*. Springer Berlin Heidelberg, 2014.

[26] 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 *Int. Conf. on Intelligent Robots and Systems (IROS)*, 2003.

[27] P. M. Wensing and D. E. Orin, "Generation of dynamic humanoid behaviors through task-space control with conic optimization," in *Int. Conf. on Robotics and Automation (ICRA)*, 2013.

[28] 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 Ad-*

vanced Robotics (ICAR), 2009.
[29] GEPETTO Team LAAS-CNRS, "sot-talos-balance," https://github.com/loco-3d/sot-talos-balance.
[30] A. del Prete, "Tsid," https://github.com/stack-of-tasks/tsid.
[31] 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," *International Journal of Humanoid Robotics*, 2016.
[32] GEPETTO Team LAAS-CNRS, "talos-torque-control," https://github.com/stack-of-tasks/talos-torque-control.
[33] T. Flayols, A. D. Prete, P. Wensing, A. Mifsud, M. Benallegue, and O. Stasse, "Experimental evaluation of simple estimators for humanoid robots," in *Int. Conf. on Humanoid Robotics (ICHR)*, 2017.
[34] 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.
[35] S. Collins, A. Ruina, R. Tedrake, and M. Wisse, "Efficient bipedal robots based on passive-dynamic walkers," *Science*, 2005.
[36] C. Mummolo and J. H. Kim, "Passive and dynamic gait measures for biped mechanism: formulation and simulation analysis," in *Robotica*, 2012.