A Whole Body Attitude Stabilizer for Hybrid Wheeled-Legged Quadruped Robots

Juan Castano, Enrico Mingo Hoffman, Arturo Laurenzi, Luca Muratore, Malgorzata Karnedula, Nikos Tsagarakis

A Whole Body Attitude Stabilizer for Hybrid Wheeled-Legged Quadruped Robots

Juan A. Castano[1], Enrico Mingo Hoffman[1], Arturo Laurenzi[1], Luca Muratore[1 2], Małgorzata Kamedula[2] and Nikos G. Tsagarakis[1]

***Abstract*— This work presents a new attitude balancing strategy implemented and validated on a quadrupedal robot equipped with a custom hybrid wheel-legged mobility system. The proposed method uses an inverse kinematics solution scheme based on Quadratic Programming optimization to generate full body motions that ensure the desired balancing performances. The strategy generates a compliant behaviour to cope with the applied external forces resulting in a stable and smooth reaction response. Furthermore, the method takes advantage of the robot hybrid wheeled-legged mobility system to provide new motion capabilities and balancing reactions as it will be shown through the paper. Extensive simulation studies on the *Centauro* robot are presented. Results show the efficiency of the propose method demonstrating significant contribution in the rejection of the applied external disturbances.**

I. Introduction

Quadruped wheeled-legged robots are an attractive solution in field robotics as they combine the advantage of the wheels efficiency [1]–[3], under moderate terrain roughness, with the ability of the articulated legs to cope with uneven terrains that are not suitable for wheeled based mobility. Furthermore, legged locomotion is more suitable to handle stairs and other obstacles as shown in several works [4]–[6].

Compared to humanoids, the quadruped body form also provides enhanced intrinsic balance capabilities due to the larger support polygon and the ability to modulate it with higher stability using the richer combinations and repertoire of motions that can be generated by the four legs. This makes quadrupedal robots hardware attractive for applications in complex environments with unstructured terrains including rescue scenarios, wild terrains, and exploration scenarios.

To provide a proper locomotion in unstructured terrain, attitude control plays an important role [7], allowing the body to keep certain orientation when traversing slopes and surpassing obstacles as shown in [8] where the authors used optimization to permit the robot traverse sloped terrain and performed a step maneuver while reducing the stress on the joints and controlling the body posture.

The research leading to these results has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement No. 644839 (CENTAURO).

[1]Advanced Robotics Department (ADVR), Istituto Italiano di Tecnologia, Genova, Italy

[2]School of Electrical and Electronic Engineering, The University of Manchester, M13 9PL, UK

{juan.castano, enrico.mingo, arturo.laurenzi, luca.muratore, malgorzata.kamedula, nikos.tsagarakis}@iit.itt

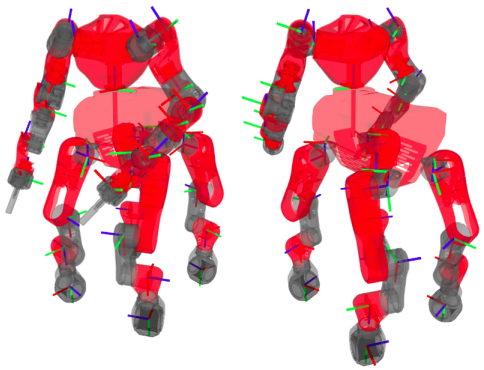

Fig. 1. Kinematics structure of the *Centauro* robot

As presented by [9] and [10] the posture of the robot can be exploited to modify the stability margins when performing wheel, legged or hybrid wheel legged locomotion. In this sense, attitude stabilization increases the stability margins when traversing rough terrain and performing other maneuvers as those presented in [11].

In the present work, we implement and validated a new whole body inverse kinematics (IK) balancing strategy that explores the whole body mobility capabilities of a hybrid wheeled-legged quadruped platform to reject external disturbances. To use a single IK problem that combines the wheels and other articulated joints of the robot, attention is paid on the proper definition of the task prioritization and tasks references in order to ensure a feasible, stable and reliable behavior of the robot while performing the balancing tasks.

The platform used to validate the proposed method is the wheeled-legged quadruped *Centauro* which is depicted in Fig. 1. *Centauro*'s design provides physical robustness against disturbances, rich proprioception, and additional characteristics [12]. From the stability point of view, the hardware incorporates intrinsic stability capabilities. However, to reject strong disturbances, additional strategies should be analyzed. Considering only wheel locomotion may not be adequate since the stability of the moving platform and the reaction velocity might compromise the stability of the robot [13]. Instead, using strategies that combines the wheels and legged reactions for disturbance rejection, terrain adaptation, and traversing of obstacles can provide further stability capabilities [14], [15], increases the robots performance and balancing reliability under external disturbances.

The implementation of the proposed approach makes use of the whole body inverse kinematics (IK) balancing strategy

in [16] which is a model based stabilizer, and the IK solver OpenSoT [17] using for code integration the XBotCore framework [18].

The balancing strategy in [16], considers a free mass model and was originally implemented in bipedal robots. The simplicity of the model permits to port the control strategy to a quadruped robot. The controller is based on the double integrator model, it generates a rotational motion of the body that absorbs the external disturbances using the whole body of the quadruped. The desired references are mapped to the joint level by means of the OpenSoT library.

Within the IK solver, we introduce a stack of tasks that prioritizes the use of wheels and considers the proper dynamic constraints during the balancing reaction generation. By the proper definition of the tasks, robot constraints and hierarchy of the different selected tasks, a robust balancing recovery behavior that takes advantage of the robot's hardware specific mobility characteristics is obtained and validated.

II. Extended Prediction Self-Adaptive Control

This section provides an overview of the Extended Prediction Self-Adaptive Control (EPSAC) algorithm [19], [20] that has been used in this work. MPC controllers use a model of the system such that the obtained control effort minimizes an objective function over a time horizon given the model dynamics. The EPSAC controller has a simple representation of the system and a noise observer which increases the performance of the closed loop. For a complete description of the method the reader is encouraged to read [21].

The generic model of the process within the EPSAC algorithm is given by

$$y(t) = \Psi(t) + n(t)\,, \tag{1}$$

where $y(t)$ is the measured output of the process, $\Psi(t)$ is the model output and $n(t)$ is the process disturbance at discrete-time t. The noise is described as a filter with transfer function

$$n(t) = \frac{C(q^{-1})}{D(q^{-1})} e(t)\,, \tag{2}$$

where $e(t)$ is uncorrelated (white) noise with zero mean and C, D are monic polynomials in the backward shift operator q^{-1}. Given (1), the predicted values of the output are

$$\mathbf{y}(t+k|t) = \overline{\mathbf{y}}(t+k|t) + \mathbf{y_{opt}}(t+k|t)\,, \tag{3}$$

the contribution of each terms is:

- $\overline{\mathbf{y}}(t+k|t)$ reflects the estimation of the output given the effect of the past inputs $u(t-1), u(t-2)\ldots$, the future base control sequence $\mathbf{u_{base}}(t+k|t)$ and the effect of the predicted disturbance $n(t+k|t)$.
- $\mathbf{y_{opt}}(t+k|t)$ reflects the optimizing control actions $\delta\mathbf{u}(t|t), \ldots, \delta\mathbf{u}(t+N_u-1|t)$, with $\delta\mathbf{u}(t+k|t) = \mathbf{u}(t+k|t) - \mathbf{u_{base}}(t+k|t)$, in a control horizon N_u.

The optimized output $\mathbf{y_{opt}}(k)\,, \forall k \in [1, 2, \ldots, N_2]$ can be expressed as the discrete time convolution of the unit impulse response coefficients $h_1, \ldots, h_{N_2}$ and unit step response coefficients $g_1, \ldots, g_{N_2}$ of the system as

$$\begin{aligned}\mathbf{y_{opt}}(t+k|t) = h_k\delta\mathbf{u}(t|t) + h_{k-1}\delta\mathbf{u}(t+1|t) + \ldots \\ +g_{k-N_u+1}\delta\mathbf{u}(t+N_u-1|t)\,.\end{aligned} \tag{4}$$

Combining (3) and (4) and writing them in vector form, the key EPSAC formulation becomes

$$\mathbf{y} = \overline{\mathbf{y}} + \mathbf{G}\mathbf{u}\,, \tag{5}$$

Then, the control effort, U, is optimized by minimizing the cost function

$$\sum_{k=N_1}^{N_2} \left[\mathbf{r}(t+k|t) - \mathbf{y}(t+k|t)\right]^2\,. \tag{6}$$

The horizons N_1, N_2 and N_u are the design parameters and $\mathbf{r}(t)$ represents the desired trajectory [22].

The cost function (6) can be represented as

$$(\mathbf{r}-\mathbf{y})^{\mathbf{T}}(\mathbf{r}-\mathbf{y}) = \left[(\mathbf{r}-\overline{\mathbf{y}}) - \mathbf{G}\mathbf{u}\right]^{\mathbf{T}}\left[(\mathbf{r}-\overline{\mathbf{y}}) - \mathbf{G}\mathbf{u}\right],$$

where $\mathbf{r} = \left[r(t+N_1|t)\ldots r(t+N_2|t)\right]^T \in \Re^{N_2}$. That can be transformed into the standard quadratic cost index

$$J(\mathbf{u}) = \mathbf{u^T H u} + \mathbf{2fu} + c\,, \tag{7}$$

with

$$\begin{aligned}\mathbf{H} = \mathbf{G^T G}, \quad \mathbf{f} = -\mathbf{G^T}(\mathbf{r}-\overline{\mathbf{y}}), \\ c = (\mathbf{r}-\overline{\mathbf{y}})^{\mathbf{T}}(\mathbf{r}-\overline{\mathbf{y}})\,,\end{aligned} \tag{8}$$

where $\mathbf{G^T G} \in \Re^{N_u \times N_u}$. Finally, the feedback characteristic of MPC is given by the fact that only the first optimal control input $\mathbf{u}^*(t) = \mathbf{u_{base}}(t|t) + \delta\mathbf{u}(t|t) = \mathbf{u_{base}}(t|t) + \mathbf{U}^*(1)$ is applied to the plant and then the whole procedure is repeated again at the next sampling instant $(t+1)$.

III. Attitude Controller

The proposed control strategy considers the orientation of the robot around the sagittal plane. However, as presented in [16], it can be extended to the lateral as well. The controller allows the robot to absorb the disturbances by a change of the body orientation dissipating the applied energy and recovering the desired body orientation. For this aim, we used the previously introduced EPSAC using as internal model the double integrator, which represents the single-degree-of-freedom rotational motion [23]. The prediction and control horizons and additional control variables were tuned accordingly. For further implementation details, readers are encouraged to read [16]. This controller gives a balancing strategy that only depends on the robot's dynamics and not on the model itself. The final motion behaviour is that of a free rotational body in space which generates rotational compliant position reference to absorb and reject disturbances. The double integrator model is $\ddot{x} = a$ being a the desired acceleration which corresponds to the control effort.

The working flow of the controller is presented in Fig. 2. Having the desired body orientation as reference , the system uses as feedback signals the angular position and velocity

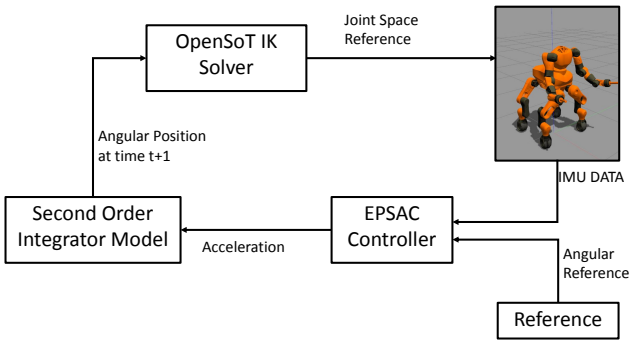

Fig. 2. Closed loop scheme of the attitude controller

which are provided by the IMU sensor of the robot. The feedback measurements and the desired reference are given to the controller generating the control effort which is the necessary acceleration to make the body converge towards the desired position with zero velocity. This control effort is evaluated in the double integrator model to generate the desired angular displacement at the next sample time. Finally, the new reference is fed into the inverse kinematics (IK) solver. In the present work, we are using the OpenSoT library to solve the IK which provides further flexibility allowing to take into consideration different constraints levels and prioritization of the tasks.

This control strategy has been applied in the bipedal robots COMAN [24] and WALK-MAN [25] with successful results. In this work, we port this controller to a wheeled-legged quadruped robot exploring the possibilities of full body IK that integrates the whole articulated body with the reaction responses from the wheel modules to enhance the balancing capabilities of the robot against strong disturbances.

IV. Full Body Inverse Kinematics

In this section we denote with ${}^{a}\boldsymbol{\omega}_{b,c}$ the twist of frame c w.r.t frame b expressed in frame a. The same notation applies for Jacobians. The state variables are the $\dot{\mathbf{q}}$ which contain both the *floating-base* velocities and the joint velocities. The quadrupedal wheeled-legged robot, namely *Centauro*, that we consider, is reported in Fig. 1. In the following parts we introduce the main tasks and constraints we considered for the *whole-body* IK as well as the way we solve it.

A. End-Effector and Floating-Base Control

For our stabilizer, we control the *floating-base* w.r.t. the *world* frame

$$ {}^{w}\mathbf{J}_{w,b}\dot{\mathbf{q}} = {}^{w}\ \boldsymbol{\omega}_{w,b}, \tag{9}$$

where ${}^{w}\mathbf{J}_{w,b}$ is the *floating-base* Jacobian and w denotes the *world* frame while b denotes the *floating-base* frame of our robot. ${}^{w}\boldsymbol{\omega}_{w,b}$ is the desired twist at the *floating-base* and is the output of the second order integrator model in Fig. 2.

The legs are controlled w.r.t. the *floating-base* frame and, in particular, we set to $\mathbf{0}$ the relative motion

$$ {}^{b}\mathbf{J}_{b,ee}\dot{\mathbf{q}} = \mathbf{0}, \tag{10}$$

where ${}^{b}\mathbf{J}_{b,ee}$ is the relative augmented Jacobian of the four legs and ee denotes the *end-effector* frame placed at the center of each wheel.

Finally we control the arms w.r.t. the *floating-base* frame when we do not want use them for the stabilizing action

$$ {}^{b}\mathbf{J}_{b,h}\dot{\mathbf{q}} = \mathbf{0}. \tag{11}$$

Alternatively, we control them w.r.t. the *world* frame

$$ {}^{w}\mathbf{J}_{w,h}\dot{\mathbf{q}} = \mathbf{0}, \tag{12}$$

where in (11) ${}^{b}\mathbf{J}_{b,h}$ is the relative augmented Jacobian, in (12) ${}^{w}\mathbf{J}_{w,h}$ is the augmented *floating-base* Jacobian and h denotes the hand frames.

B. Pure Rolling Condition

We consider all the wheels in full contact with the ground at the point $\mathbf{p}_i$. Lets call $\hat{\mathbf{n}}_i$ the normal vector to the tangent contact plane. We associate a contact frame $P_i = \left(\mathbf{p}_i, \hat{\mathbf{t}}_i, \hat{\mathbf{l}}_i, \hat{\mathbf{n}}_i\right)$ where $\hat{\mathbf{l}}_i$ is parallel to the wheel rotation axis and $\hat{\mathbf{t}}_i = \hat{\mathbf{l}}_i \times \hat{\mathbf{n}}_i$. The contact frame P_i is centered in $\mathbf{p}_i$

$$ \mathbf{p}_i = \mathbf{p}_{ee,i} - r\hat{\mathbf{n}}_i, \tag{13}$$

where $\mathbf{p}_{ee,i}$ is the center of the wheel frame $P_{ee,i}$ and r is the wheel radius, as shown in Fig. 3.

We consider the twist ${}^{P_i}\boldsymbol{\omega}_{w,P_i}$ of the contact point in P_i. The *Pure Rolling Condition* at contact point P_i can be written as

$$ \mathbf{S}\,{}^{P_i}\boldsymbol{\omega}_{w,P_i} = \mathbf{0}, \tag{14}$$

where $\mathbf{S}$ is a selection matrix

$$ \mathbf{S} = \left[\mathbf{I}_{\mathbf{4\times 4}}\ \mathbf{O}_{\mathbf{4\times 2}}\right], \tag{15}$$

The conditions in (14) mean that the contact point P_i does not move w.r.t. the *world* frame and does not rotate about the local x-axis.

The twist of the contact point can be also written as

$$ {}^{P_i}\boldsymbol{\omega}_{w,P_i} = {}^{P_i}\mathbf{J}_{w,P_i}\dot{\mathbf{q}}, \tag{16}$$

where ${}^{P_i}\mathbf{J}_{w,P_i}$ is the *floating-base* Jacobian of the contact point. Therefore our constraint becomes

$$ \mathbf{S}\,{}^{P_i}\mathbf{J}_{w,P_i}\dot{\mathbf{q}} = \mathbf{0}. \tag{17}$$

For our controller we consider four constraints (17), one for each leg/wheel.

C. IK Solver

In our controller we consider three level of priorities:

0) pure rolling conditions for the four wheels
1) *floating-base* + arms control
2) legs control

where 0 is the highest priority. Each level represent a cost function (or task) to minimize without affecting the result obtained at the previous level. Furthermore, these tasks are minimized subject to joint limits and joint velocity limits.

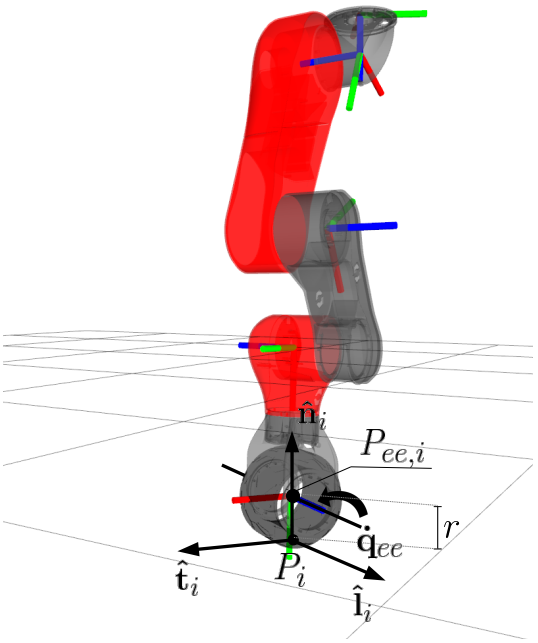

Fig. 3. Contact frame definition

Therefore, for each level, we setup a Quadratic Programming (QP) problem in the form

$$\begin{aligned} \underset{\dot{\mathbf{q}}}{\operatorname{argmin}} \; & \|\mathbf{J}_i\dot{\mathbf{q}}_i - \boldsymbol{\omega}_i\|^2 + \lambda\|\dot{\mathbf{q}}_i\|^2 \\ s.t. \; & \underline{\mathbf{u}} \leq \dot{\mathbf{q}}_i \leq \overline{\mathbf{u}} \\ & \mathbf{J}_{i\text{-}1}\dot{\mathbf{q}}_{i\text{-}1} = \mathbf{J}_{i\text{-}1}\dot{\mathbf{q}}_i \\ & \vdots \\ & \mathbf{J}_0\dot{\mathbf{q}}_0 = \mathbf{J}_0\dot{\mathbf{q}}_i \end{aligned} \tag{18}$$

where the second term in the cost function is a regularization, $[\underline{\mathbf{u}} \; \overline{\mathbf{u}}]$ are the joint limits and joint velocity limits constraints and all the equality constraints represent the priorities between the tasks. This technique is also called *inequality Hierarchical QP* [26].

D. Software Architecture

The XBotCore (*Cross-Bot-Core*) [18], a light-weight, hard Real-Time (RT) software platform robot control was used as a middle-ware to implement the plugin that runs the stabilizer and the full body inverse kinematics. The framework provides a set of open-source plugins for the Gazebo simulator [1] which allows for a transparent code portability from simulation to the real robotic hardware. We carried out the experiments described in the next section implementing an RT XBotCore plugin for the high-level control of the *Centauro* inside the Gazebo environment.

The *Plugin Handler* is the main component of the architecture: it is a RT thread responsible to start all the loaded plugins, to execute them sequentially and to close them before unloading them. In Fig. 4 we show that the *Plugin Handler* is running the IK Stabilizer RT plugin that

[1] https://github.com/ADVRHumanoids/GazeboXBotPlugin

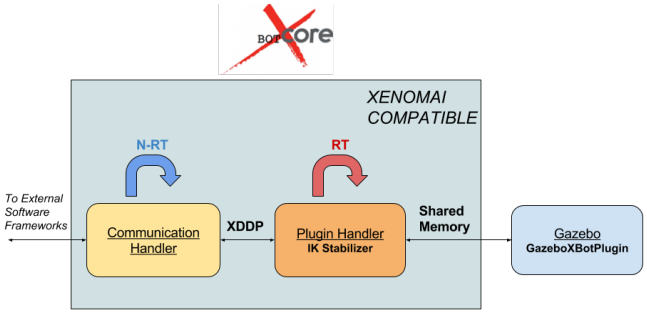

Fig. 4. XBotCore threads and communication mechanisms: *Communication Handler*, *Plugin Handler* and *GazeboXBotPlugin*.

communicates using a shared memory mechanism with the *GazeboXBotPlugin*. The *Communication Handler* non-RT thread is used to start/stop and check the status of the plugin during the simulation.

V. Validation Results

In this work, we are providing only simulation results since the real platform is not yet available for experimental testing. However, the used model and physical constraints are equivalent to those in the real robot.

The proposed method was validated by applying on the robot strong pushing disturbances along different directions in the sagittal plane. The initial posture of the *Centauro* was defined to provide a good intrinsic stability along the sagittal and lateral planes, meaning that the CoM is inside the support-polygon provided by the four legs on the ground. The wheels are align in the direction of the applied disturbance allowing us to evaluate the performance of the proposed IK stack.

Given the hardware characteristics of *Centauro*, it is desired that the robot uses the wheels not only to move but to perform wheeled-legged locomotion. In this case, the wheels absorb the impact as much as possible, taking advantage of the intrinsic stability properties of quadrupeds; this way, a linear motion in the wheels direction absorbs part of the impact as long as it does not compromise the robot stability. i.e. if the robot is closed to an edge the wheels might absorb part of the applied energy and then should be blocked to avoid a possible fall. To increase the stability margin of the robot and avoid tip-overs, full body control strategies as the one presented in this work are required. This kind of strategies will be useful when the robot has a limit range of motion or when hard interactions are given and the wheels action is not able to compensate them or the terrain roughness prohibits the use of the wheels action.

Another useful scenario would be when the wheels configuration are not align in the direction of the disturbances, therefore only body balancing strategies are available to compensate the disturbances.

A. Front Push Disturbance

Firstly, we applied a -500 N disturbance in the *Centauro's* pelvis during 0.3 s which is equivalent to a 150 Ns momen-

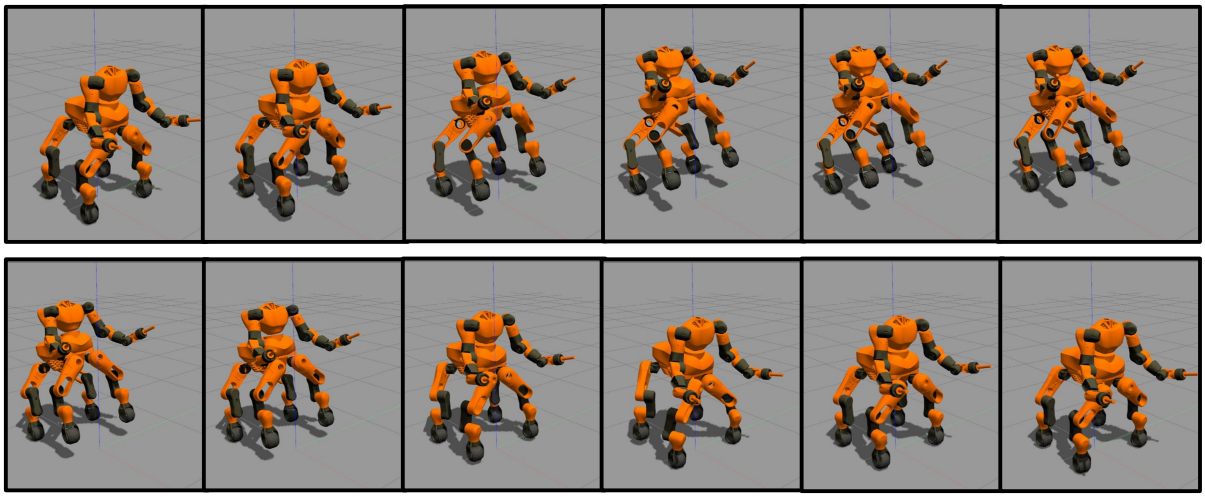

Fig. 5. Snapshots of the simulation when a frontal push is applied to *Centauro*.

tum resulting to a velocity disturbance of 1.6 m/s in a robot that weights 90 Kg. As it is seen in Fig. 5 the robot is pushed back and the front part of the body rises. To compensate the disturbance, the robot changes its Cartesian position using the wheels and the whole body dynamics allowing the robot to recover the original position.

This is seen from the IMU data obtained during simulation. As shown in Fig. 6, the applied disturbance makes the robot falls if no recovery action is performed. When the proposed controller and IK stack are used, the robot absorb the impact and recover the desired position in 2 s. Given the force of the impact, once the frontal disturbance is compensated with a maximum deviation of 25°, the body tilts forward and a second balancing action is taken to avoid the robot to fall forward with an angular deviation of 7. As

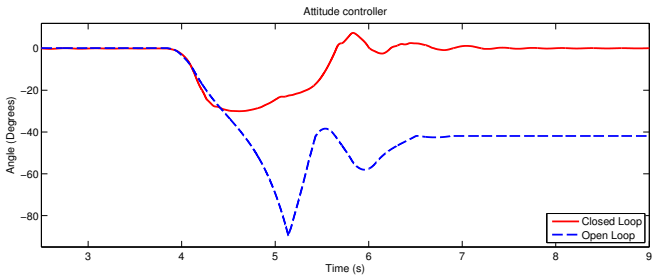

Fig. 6. Robot's angular deviation due to a -500 N (0.3 s in duration) push at the back of the robot.

it can be observed in Fig. 7, When the impact is applied the robot's wheels are moved backward by 25.5 cm, changing the robots initial position and absorbing the impact. Once the disturbance is compensated, the robot moves back to its initial location and the gait is recovered.

As mention previously, the whole body and not only the wheels are involved in the stabilization and posture recovery of the robot. In Fig. 8, the joints' angular position of the

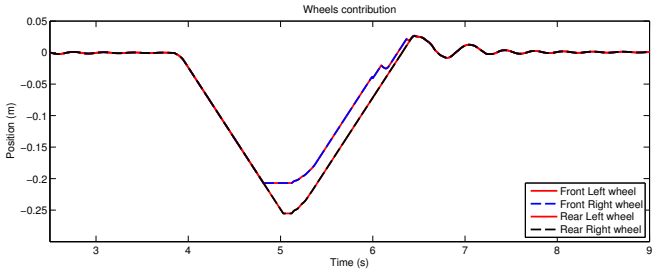

Fig. 7. Wheels contribution when rejecting a -500 N (0.3 s in duration) push at the back of the robot.

front left leg are shown. As it can be seen, the fifth joints move during the impact to compensate it.

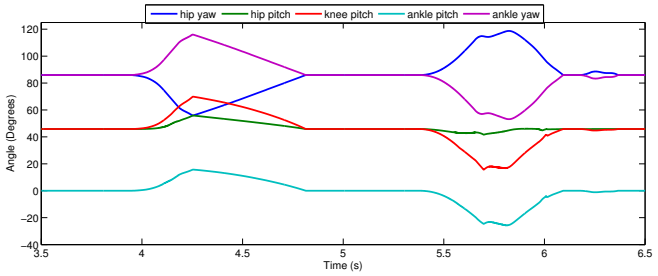

Fig. 8. Front left leg joint position when -500 N push at the pelvis of the robot is applied.

B. Back Push Disturbance

In the second simulation, we applied a 300 N disturbance to the pelvis of the *Centauro* in the rear part of the body in the positive sagittal direction for 0.3 s which is equivalent to a 90 Ns momentum or 1 m/s disturbance in a robot that weights 90 Kg. Snapshots of the simulation are given in Fig. 9. As it is seen, after the impact the body is moved forward and the whole-body is involved in the stabilization action.

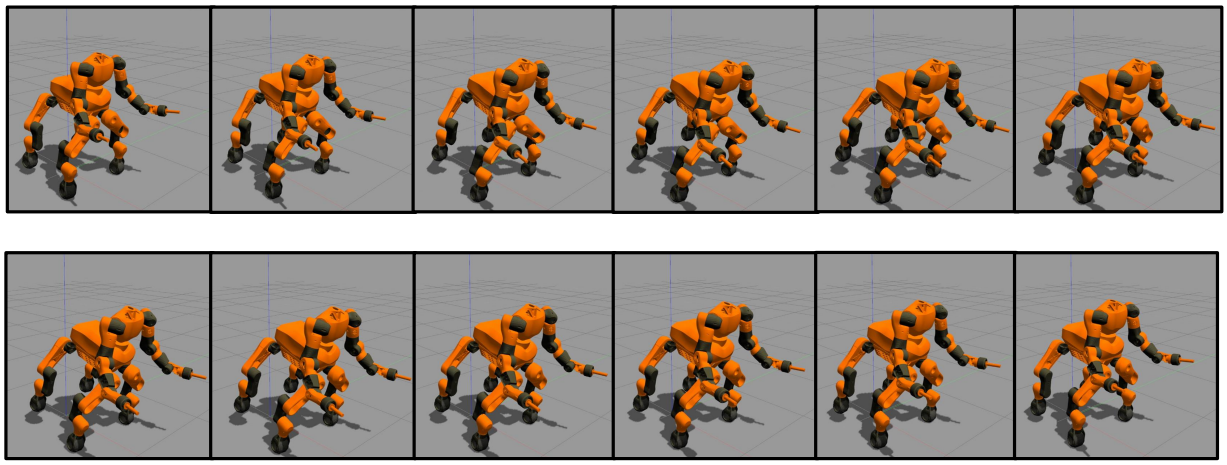

Fig. 9. Snapshots of the simulation when a back push is applied to *Centauro*.

Even though the rear wheels loose contact with the ground, the rear legs react to modify the whole body dynamics and recover the initial position.

As it can be seen in Fig. 10 the open loop response of the robot diverges showing that the robot is not able to keep its balance on its own when a disturbance of this kind is applied. On the other hand, when the attitude strategy is applied, the robot is able to recover the initial position in 1.5 s. The obtained deviation was 12° and the disturbance was rejected using the whole robot including the wheels' reactions.

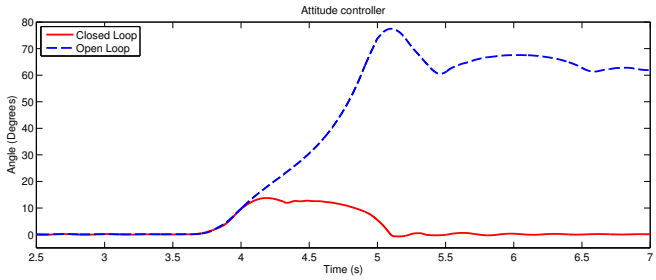

Fig. 10. Robot's angular deviation due to a 300 N push at the back of the robot.

As it is depicted in Fig. 11, the robot uses rear and front wheels to move forward by 12 cm. Following that, the front wheels continue modifying the body position at a lower rate. Once the energy of the disturbance is absorbed, the robot recovers the desired orientation and the initial position.

Additional simulations where carried out using different homing configuration, this permits us to evaluate the performance of the attitude controller and stack of task under different working conditions. As it is introduced in Fig. 12 the desired configuration favors the sagittal motion of the knees and hip joints of the *Centauro* robot. When the push is applied, the robot moves back and looses contact with the ground. To recover the gait, he robot changes the legs position and rolls back using the wheels. This action modifies the center of mass of the robot and allows it to keep balance

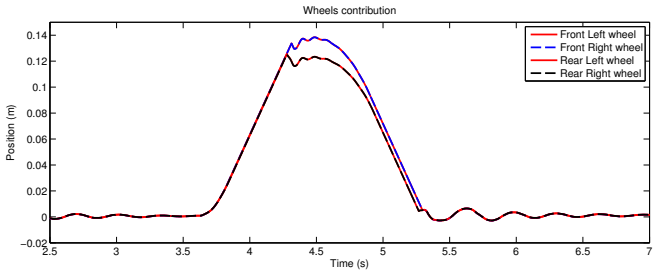

Fig. 11. Wheels contribution when rejecting a 300 N push at the back of the robot.

and recover the desired posture.

VI. Conclusions

The present work implemented a new attitude stabilizer for the hybrid wheeled-legged quadruped robot *Centauro*. The proposed stabilizer takes advantage of the full body articulated mobility mixed with the wheeled functionality to increase the stability margin of the robot when rejecting external disturbances. The obtained behavior shows that the resulted mixed articulated-wheeled control strategy uses the wheels of the robot as well as the leg joints providing whole-body compensation strategy to dissipate the applied energy and recover the initial posture after strong disturbances.

Using the OpenSoT library it is possible to consider the full body of the robot including wheels during the optimization of the IK. To produce the expected behavior a stack of tasks that prioritizes the use of wheels and considers the proper dynamic constraints during the motion were presented. By the proper definition of the tasks, robot constraints and hierarchy of the different selected tasks, a robust behavior that takes advantage of the robots' hardware specific mobility characteristics was obtained and whole-body wheeled-legged balancing stabilization was obtained.

The performance of the proposed method demonstrates the directions towards robust wheeled-legged balancing and

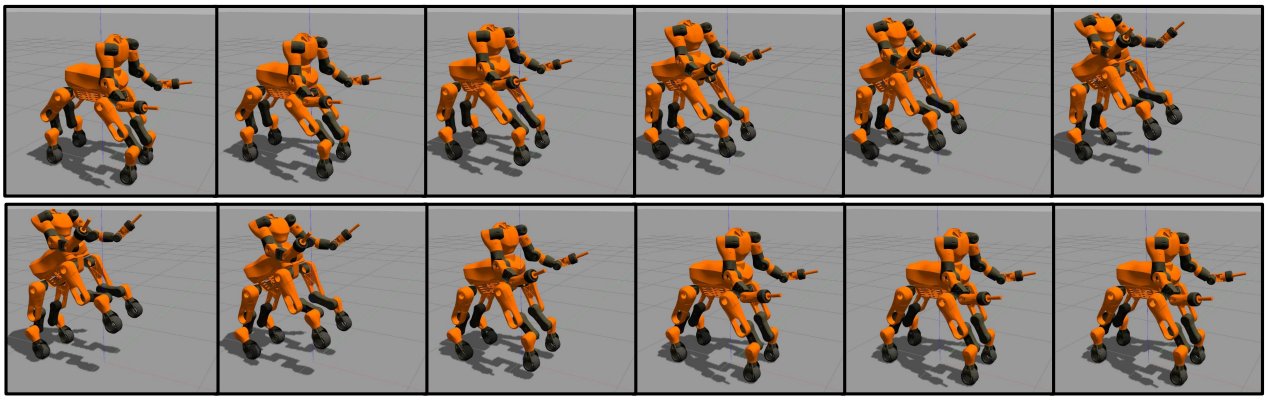

Fig. 12. Snapshots of *Centauro* recovering from a frontal push i a different homing configuration.

locomotion gaits which will permit the robot to perform securely in complex scenarios on unstructured terrains, providing the performance required in real scenarios.

REFERENCES

[1] R. Volpe, J. Balaram, T. Ohm, and R. Ivlev, "Rocky 7: a next generation mars rover prototype," *Advanced Robotics*, vol. 11, no. 4, pp. 341–358, 1996.

[2] E. Rollins, J. Luntz, A. Foessel, B. Shamah, and W. Whittaker, "Nomad: a demonstration of the transforming chassis," in *Proceedings. IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*, vol. 1, 1998, pp. 611–617 vol.1.

[3] K. Iagnemma and S. Dubowsky, "Traction control of wheeled robotic vehicles in rough terrain with application to planetary rovers," *The International Journal of Robotics Research*, vol. 23, no. 10-11, pp. 1029–1040, 2004.

[4] J. Buchli, M. Kalakrishnan, M. Mistry, P. Pastor, and S. Schaal, "Compliant quadruped locomotion over rough terrain," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 814–820.

[5] B. Huang, M. Li, and P. Wang, "A parent-children type robot system for scout: Mechanics, control, simulation and test," in *International Conference on Mechatronics and Automation*, 2006, pp. 406–410.

[6] C. Grand, F. Benamar, and F. Plumet, "Motion kinematics analysis of wheeledlegged rover over 3d surface with posture adaptation," *Mechanism and Machine Theory*, vol. 45, no. 3, pp. 477 – 495, 2010.

[7] J. Pan, "Measurement and control of attitudes of quadruped robots," in *Intelligent Robots and Systems '94. 'Advanced Robotic Systems and the Real World', IROS '94. Proceedings of the IEEE/RSJ/GI International Conference on*, vol. 2, Sep 1994, pp. 1332–1337 vol.2.

[8] T. Thomson, I. Sharf, and B. Beckman, "Kinematic control and posture optimization of a redundantly actuated quadruped robot," in *IEEE International Conference on Robotics and Automation*, 2012, pp. 1895–1900.

[9] G. Besseron, C. Grand, F. B. Amar, and P. Bidaud, "Decoupled control of the high mobility robot hylos based on a dynamic stability margin," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sept 2008, pp. 2435–2440.

[10] V.-G. Loc, I. M. Koo, D. T. Tran, S. Park, H. Moon, and H. R. Choi, "Improving traversability of quadruped walking robots using body movement in 3d rough terrains," *Robotics and Autonomous Systems*, vol. 59, no. 12, pp. 1036 – 1048, 2011.

[11] X. Wu, X. Shao, and W. Wang, "Stable quadruped walking with the adjustment of the center of gravity," in *IEEE International Conference on Mechatronics and Automation*, 2013, pp. 1123–1128.

[12] L. Baccelliere, N. Kashiri, L. Muratore, A. Laurenzi, M. Kamedula, A. Margan, S. Cordasco, J. Malzahn, and N. G. Tsagarakis, "Development of a human size and strength compliant bi-manual platform for realistic heavy manipulation tasks," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017, pp. 5594–5601.

[13] A. Suzumura and Y. Fujimoto, "Real-time motion generation and control systems for high wheel-legged robot mobility," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 7, pp. 3648–3659, July 2014.

[14] K. Turker, I. Sharf, and M. Trentini, "Step negotiation with wheel traction: a strategy for a wheel-legged robot," in *2012 IEEE International Conference on Robotics and Automation*, May 2012, pp. 1168–1174.

[15] G. Besseron, C. Grand, F. B. Amar, F. Plumet, and P. Bidaud, *Locomotion Modes of an Hybrid Wheel-Legged Robot*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 825–833.

[16] J. A. Castano, C. Zhou, Z. Li, and N. Tsagarakis, "Robust model predictive control for humanoids standing balancing," in *International Conference on Advanced Robotics and Mechatronics (ICARM)*, 2016, pp. 147–152.

[17] E. Mingo Hoffman, A. Rocchi, A. Laurenzi, and N. G. Tsagarakis, "Robot control for dummies: Insights and examples using opensot," in *17th IEEE-RAS International Conference on Humanoid Robots, Humanoids*, 2017, pp. 736–741.

[18] L. Muratore, A. Laurenzi, E. M. Hoffman, A. Rocchi, D. G. Caldwell, and N. G. Tsagarakis, "Xbotcore: A real-time cross-robot software platform," in *2017 First IEEE International Conference on Robotic Computing (IRC)*, April 2017, pp. 77–80.

[19] R. De Keyser, *Model Based Predictive Control*, ser. UNESCO Encyclopaedia of Life Support Systems. EOLSS Publishers, 2003.

[20] R. De Keyser and C. Ionescu, "The disturbance model in model based predictive control," in *Control Applications. CCA. Proceedings of IEEE Conference on*, vol. 1, 2003, pp. 446–451 vol.1.

[21] J. A. Castano, A. Hernandez, Z. Li, N. G. Tsagarakis, D. G. Caldwell, and R. D. Keyser, "Enhancing the robustness of the {EPSAC} predictive control using a singular value decomposition approach," *Robotics and Autonomous Systems*, pp. –, 2015.

[22] J. Sánchez and J. Rodellar, *Adaptive Predictive Control: From the Concepts to Plant Optimization*, ser. Prentice-Hall International Series in Systems and Control Engineering. Prentice Hall PTR, 1996.

[23] V. Rao and D. Bernstein, "Naive control of the double integrator: a comparison of a dozen diverse controllers under off-nominal conditions," in *Proceedings of the American Control Conference.*, vol. 2, Jun 1999, pp. 1477–1481 vol.2.

[24] N. Tsagarakis, S. Morfey, G. Medrano-Cerda, Z. Li, and D. Caldwell, "Compliant humanoid coman: Optimal joint stiffness tuning for modal frequency control," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2013, pp. 665–670.

[25] N. G. Tsagarakis, D. G. Caldwell, F. Negrello, W. Choi, L. Baccelliere, V. Loc, J. Noorden, L. Muratore, A. Margan, A. Cardellino *et al.*, "Walk-man: A high-performance humanoid platform for realistic environments," *Journal of Field Robotics*, 2016.

[26] O. Kanoun, F. Lamiraux, and P.-B. Wieber, "Kinematic control of redundant manipulators: Generalizing the task-priority framework to inequality task," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 785–792, 2011.