Online Payload Identification for Quadruped Robots

Guido Tournois[1,2], Michele Focchi[1], Andrea del Prete[3], Romeo Orsolino[1], Darwin G. Caldwell[1], Claudio Semini[1]
[1] Dept. of Advanced Robotics, Istituto Italiano di Tecnologia, Genova, Italy
[2] Delft University of Technology, Delft, Netherlands [3] LAAS-CNRS, Toulose, France.
michele.focchi@iit.it

***Abstract*— The identification of inertial parameters is crucial to achieve high-performance model-based control of legged robots. The inertial parameters of the legs are typically not altered during expeditions and therefore are best identified *offline*. On the other hand, the trunk parameters depend on the modules mounted on the robot, like a motor to provide the hydraulic power, or different sets of cameras for perception. This motivates the use of recursive approaches to identify *online* mass and the position of the Center of Mass (CoM) of the robot trunk, when a payload change occurs. We propose two such approaches and analyze their robustness in simulation. Furthermore, experimental trials on our 80-kg quadruped robot HyQ show the applicability of our strategies during locomotion to cope with large payload changes that would otherwise severely compromise the balance of the robot.**

I. Introduction

Knowledge of the robot's dynamic model parameters is of paramount importance for controlling a robot [1], especially if fast, accurate and dexterous motions are required. Indeed, most advanced locomotion controllers are model based [2]–[4] and rely heavily on a feedforward action to predict the necessary control forces to realize such actions [5]. Due to the inherent delay, feedback control is no longer sufficient for these intrinsically unstable systems The advantage of feedforward action lies therefore in the possibility of specifying the necessary control forces in advance [6].

However, the drawback of these methods is that they rely on the accuracy of the robot dynamic model [1], [6]. Robot dynamic model parameters are usually calculated using computer-aided design (CAD) software [1], [7]–[9]. Although CAD software can calculate these parameters accurately, additional elements like hoses and cables are usually not taken into account [1]. Moreover, the density set for the materials can differ from reality [9], causing significant errors in the dynamic model parameters Inaccurate parameters result in a wrong prediction of the forces, moving the control burden toward the feedback controller and thus increasing tracking errors.

In addition, most stability criteria in legged locomotion are heavily affected by model errors as they depent on an accurate estimation of the robot CoM [10]. A common strategy to increase robustness against CoM uncertainties [11] is to ensure a reasonable *stability margin* inside the support polygon. Ensuring such margin, however, becomes difficult on rough terrains, where only a limited set of potential footsteps is available. There can be situations where the support polygon shape shrinks and degenerates and where ensuring a sufficient margin to account for CoM uncertainty is impossible. Therefore, an accurate model is crucial for legged robots that are meant to traverse unstructured environments. Lastly, proper knowledge of the robot model is needed to accurately simulate the robot dynamics.

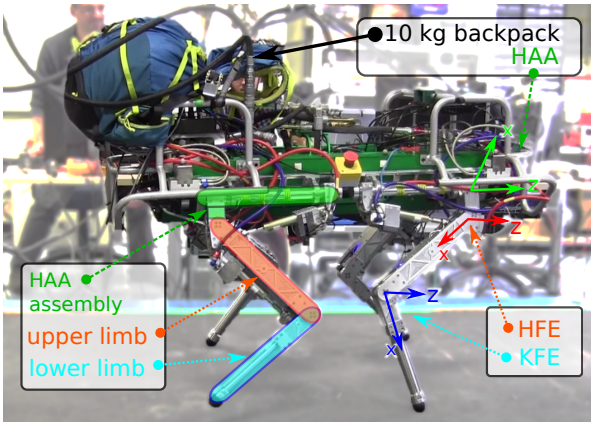

Fig. 1. Picture of IIT's 80 kg HyQ loaded with a 10 kg backpack. The names of the links are shown on the picture itself together with the joints definitions: Hip Adduction-abduction (HAA), Hip Flexion-Extension (HFE), Knee Flexion-Extension (KFE).

Identification of dynamic model parameters or equivalently inertial parameter for legged robots has recently been investigated [1], [6], and is mostly carried out *offline*. *Offline* identification is appropriate for the robot legs, since the mechanical structure (and therefore the model parameters) does not change during locomotion. The trunk of the robot, however, often undertakes inertial changes during locomotion (e.g. due to a payload change). In particular, the trunk parameters depend on the different modules that are mounted on it, like an onboard hydraulic power unit, different sets of perception sensors or mission payloads (e.g. a backpack). Furthermore, arms can be mounted on the trunk for manipulation tasks, creating a centaur [12]. Hence, for trunk inertial parameters, an *online* identification strategy would be beneficial to identify the changes in the trunk parameters (CoM, mass and inertia) and promptly reflect them into the trunk controller that exploits the rigid body model of the robot to compute the joint torques [13].

Examples of *online* identification of the inertial parameters of manipulators can be found in [14], [15] but these works were not extended to floating base systems. More-

over, approaches targeting specifically at the locomotion and planning problems, consider the dynamic model to be too complex to be precisely identified. Tracking performance is improved by learning policies that start off from the inaccurate models and thus do not improve the knowledge of the model [16]. Regarding online estimation approaches for legged robots, recently Stephens [17] improved his state estimator to incorporate CoM offsets.

A. Contributions

This work focuses on the identification of the inertial parameters of the torque-controlled hydraulic quadruped HyQ [18]. HyQ is a robot designed for robust and versatile movements on rough terrain. Considering the characteristics of our robot, we take a hybrid offline-online approach. In the first part of the paper we briefly describe the identification of the leg parameters, which is performed *offline*. In the second part we propose multiple original techniques to identify *online* the parameters of the trunk of the robot, which is our main contribution. The latter approach, together with our trunk controller that we use to ensure locomotion stability, can be seen as adaptive controls where the accuracy of the dynamic model is improved *online* whenever a change in the load is detected. Moreover, our experimental contribution is to show experiments of HyQ walking on flat and rough terrain while we add weight on its trunk. The robot detects the payload change, performs the online identification and continues walking with the updated parameters [1].

B. Outline

This paper is structured as follows: in Section II we start with the generic theory of identification, specified for quadrupeds; Section III presents a state-of-the-art trunk CoM identification method [13] (which is used as ground truth) and two recursive strategies for *online* identification. Sections IV and V present the data collected from simulation and experiments both for our offline and online identification approaches. Finally, Section VI concludes the paper.

II. Offline Leg Parameter Identification

In this section we briefly summarize state-of-the-art *offline* identification techniques to identify the inertial parameters of the HyQ legs. In offline identification some desired motion is commanded to the joints and the corresponding joint torques and motion are recorded. Then the model parameters are fitted on this data such that the relationship between the applied torques and the observed motion is best described. Significant work has been done on these data driven approaches using a floating-base model [1], [6].

However, using such a floating-base model and thus by including the dynamics of the trunk it is hard to observe the influence of the single leg parameters. Indeed, as the trunk mass is significantly bigger than the link masses, the torques caused by link motion will be hard to distinguish from those due to the trunk motion, making the identification more difficult. In addition, a floating base introduces some difficulties: namely, 1) contact states can change, i.e. a leg can be pulled up, 2) balance requirements might limit the allowable motions [6]. Therefore, we fixate the trunk and identify the parameters of a single leg at a time. As HyQ can be seen as a tree-structured mechanism, each leg is considered as a fixed based manipulator.

The starting point of the offline identification is the generic set of Lagrange equations which maps the motion of a system to output torques or vice versa. Therefore, for a single leg with n joints, the well-known dynamic equations of motion are written as [6]:

$$M(q)\ddot{q} + c(\dot{q}, q) = \tau \tag{1}$$

where $q \in \mathbb{R}^n$ represents the vector of joint angular positions of the leg joints, $M(q) \in \mathbb{R}^{n\times n}$ is the inertia matrix, $c(q, \dot{q}) \in \mathbb{R}^n$ is the vector of centripetal, Coriolis and gravity forces and $\tau \in \mathbb{R}^n$ is the vector of the joint torques. As each HyQ leg has 3 Degrees of Freedom (DoFs), henceforth we will consider $n = 3$. The set of equations (1) are linear in the model parameters [6], [19] and the left hand side of (1) can accordingly be linearized towards a set of inertial model parameters by

$$Y(q, \dot{q}, \ddot{q})\phi = \tau, \tag{2}$$

with $Y(q, \dot{q}, \ddot{q}) \in \mathbb{R}^{n\times(p\times 10)}$ and $\phi = [\phi_1^T \ \phi_2^T \ \dots \ \phi_p^T]^T$ being the sets of inertial parameters for the p rigid bodies. Each rigid body i has 10 inertial parameter: mass, the center of mass and the inertia tensor, given by

$$\begin{aligned}\phi_i = [&m_i \quad m_i c_{xi} \quad m_i c_{yi} \quad m_i c_{zi} \dots \\ &\dots \quad I_{i,xx} \quad I_{i,xy} \quad I_{i,xz} \quad I_{i,yy} \quad I_{i,yz} \quad I_{i,zz}\ ]^T.\end{aligned} \tag{3}$$

HyQ has 3 links per leg, so $p = 3$, resulting in 30 inertial parameters per leg. In order to solve this system of equations (2) for the constant unknown leg parameters, we collect over m data samples and stack the measurements to obtain an over-defined system of equations, given by

$$\overline{\mathbf{Y}}\phi = \overline{\tau}. \tag{4}$$

Here, the barred notation denotes the augmented stack of their respective counterparts from (2). In addition, $\overline{\mathbf{Y}} \in \mathbb{R}^{(m\times n)\times p}$ is called the *regressor* matrix. The *regressor* matrix often is not full rank (i.e. certain parameters that are not observable [1]). Applying a damped pseudoinverse of $\overline{\mathbf{Y}}$ to compute ϕ, would result in zero values for some parameters. Instead, it is useful to minimize the distance from the CAD parameter values, ϕ_{CAD}, for the unobservable subspace (nullspace of $\overline{\mathbf{Y}}_B$) to the inertial parameters as

$$\phi = \overline{\mathbf{Y}}^+\overline{\mathbf{y}} + N_{\overline{\mathbf{Y}}}(\phi_{CAD} - \overline{\mathbf{Y}}^+\overline{\mathbf{y}}) \tag{5}$$

where $\overline{\mathbf{Y}}^+$ is the pseudoinverse of the rank deficient $\overline{\mathbf{Y}}$ and where $N_{\overline{\mathbf{Y}}}$ forms a null space basis for $\overline{\mathbf{Y}}$.

Remark 1: In alternative, a weighted approach could be implemented, but this has the drawback that a wrong choice of the weights would shift the identification too much towards the CAD parameters.

[1] A video of the experiments is available at https://youtu.be/EDsBPYudHik.

Remark 2: If a subset of the parameters has to be identified it is necessary to partition (5) into the contribution of the parameters to be identified and those who are not. The latter still contribute to the torques and have therefore to be moved to the right hand side of (5), to obtain

$$\overline{\mathbf{Y}}_{\text{id}}\phi_{\text{id}} = \overline{\mathbf{y}} - \overline{\mathbf{Y}}_{\text{f}}\phi_{\text{f}}, \tag{6}$$

with ϕ_{id} and ϕ_{f} the subsets of parameters to be identified and fixed parameters and $\overline{\mathbf{Y}}_{\text{id}}$ and $\overline{\mathbf{Y}}_{\text{f}}$ their respective corresponding columns from the *regressor* matrix.

A. Observability

The rank of the *regressor* matrix (i.e. the number of linearly-independent columns) represents the number of parameters that are observable [19]. A method to check whether a specific parameter is observable is to compute the observable subspace of *regressor* matrix, as explained in [20]. Often, only a linear combination of some parameters is observable, rendering these parameters partially observable. By optimizing the joint trajectories [9], the number of observable parameters can be maximized.

III. Online Trunk CoM Identification

In this section we will explain three methods for online identification of the trunk parameters. First in Section **??** we identify the trunk mass. Second in III-A, a *batch* Least-Squares state-of-the-art approach [13] to identify trunk CoM (approach (a)) is introduces which will be used as a base line. In Sections III-B and III-C, we describe two *recursive* strategies for identification of the trunk CoM: one based on the orientation error (approach (b)) and another based on the contact forces (approach (c)). The former only exploits inertial (IMU) measurements but requires a trunk posture controller. The latter, on the other hand, works with any controller, but requires accurate measurement/estimation of the contact forces. Recursive strategies are preferable over batch ones, because they allow to reduce the identification time. Indeed, in batch approaches, to reduce the effect of noise data has to be collected over larger time intervals. Moreover, the identification can be done smoothly, rather than in one go.

We do not identify the trunk inertia tensor because it only comes into play at very dynamic motions. Indeed, after performing some simulations, we assessed the impact of inertial forces on joint torques at different walking speeds. In practice, we evaluated the integral of the norm of the vector of joint torques (during one locomotion cycle) and we compared it with the norm of the vector of torques related to inertial forces. For the range of walking speed that we usually set in our locomotion experiments, [5,10,15,15]cm/s, we found that the inertial forces accounted only for [9.5,11.5,15,22]% of the total joint torques. Therefore, as the influence of the trunk inertia tensor is minor, its identification is out of the scope of this work.

As a prerequisite to the online approaches, the trunk mass has to be estimated. Based on the contact forces, the robot

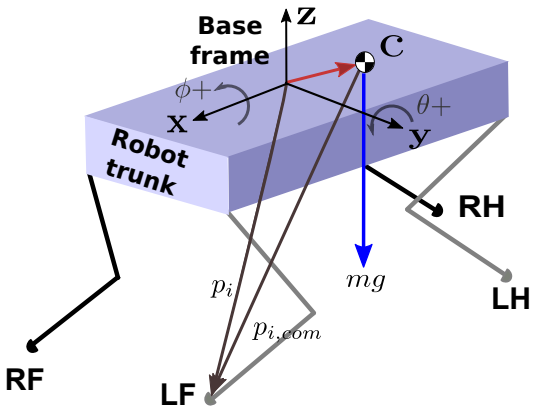

Fig. 2. Robot sketch to define the vectors, the frames and to show the sign convention. The four legs are called Left Front (LF), Right Front (RF), Left Hind (LH), Right Hind (RH).

mass m_{r} is identified in static position as

$$m_{\text{r}} = \sum_{i=1}^{N} f_{i_z}/g \tag{7}$$

with N the number of stance legs, f_{i_z} the vertical contact force per stance leg and g gravity. Under the assumption that the mass of the legs, m_{l} is known or estimated reasonably close to their real values (from Section II), we subtract their contribution to find the trunk mass as $m_{\text{tr}} = m_{\text{r}} - m_{\text{l}}$.

A. Identification of trunk CoM with static poses

This approach is based on the equation of the net moment around the CoM of the entire robot, expressed by

$$\Gamma = \sum_{i=1}^{N} (p_{i,c} \times f_i) = \sum_{i=1}^{N} ((p_i - c) \times f_i) \tag{8}$$

with $f_i \in \mathbb{R}^3$ the contact forces at point i and N is the number of ground contacts. Moreover, $p_{i,c} \in \mathbb{R}^3$ is the lever arm of contact force f_i w.r.t. the whole robot CoM and is rewritten as the difference from the Cartesian position of the contact point $p_i \in \mathbb{R}^3$ and the whole robot CoM $c \in \mathbb{R}^3$ both w.r.t. the base frame origin (see Fig. 2).

For a static robot it follows that the net moment Γ equals zero, hence (8) equals zero, such that by rearranging we get

$$\sum_{i=1}^{N} (c \times f_i) = \sum_{i=1}^{N} (p_i \times f_i)\,. \tag{9}$$

By using the skew symmetric operator to express the cross product, we can further rearrange to

$$-\left(\sum_{i=1}^{N} [f_i]_\times\right) c = \sum_{i=1}^{N} (p_i \times f_i)\,. \tag{10}$$

The whole robot CoM can be partitioned as $c = (m_{\text{tr}}c_{\text{tr}} + m_{\text{l}}c_{\text{l}})/(m_{\text{tr}} + m_{\text{l}})$, where m_{tr} and c_{tr} are the trunk mass and trunk CoM. Moreover, m_{l} and p_{l} are the mass and CoM of the legs. Again, assuming that the mass and CoM of the different leg links are known, or estimated reasonably close to their real values (from Section II), we can subtract

their contribution from the whole robot CoM and find a relationship between the trunk CoM and the contact forces:

$$-\left(\sum_{i=1}^{N}[f_i]_\times\right)c_{\text{tr}} = \sum_{i=1}^{N}\left(\frac{(m_r)p_i - m_{\text{l}}c_{\text{l}}}{m_{\text{tr}}} \times f_i\right). \quad (11)$$

By collecting numerous sets of contact force estimates and positions for different static predefined poses of the trunk, we obtain an over-defined system of equations that can be solved for c_{tr} in a least-square (LS) fashion.

Remark: It is important that the predefined poses render the over-defined system matrix full rank in order to observe the three CoM components. For this it suffices to have the base wrenches not aligned with gravity (e.g. trunk not horizontal).

B. Identification of trunk CoM using orientation error

This approach is built on top of a trunk posture controller [13] that controls the position of the robot CoM and the orientation of the trunk by means of *virtual springs*. The controller also compensates for *gravity*, by applying a force equal and in opposite direction of gravity and passing through the CoM (see Fig. 2). However, if the estimated CoM $\hat{c}$ used in the controller (e.g. obtained by CAD data) is different from the real CoM c, the gravity compensation force creates a tipping moment around the real CoM due to the lever arm $({}_w\hat{c} - {}_w c)$ (see Fig. 3(left)). This will result in either pitching or rolling of the robot (or both) until this tipping moment is equilibrated by the returning torque due to the torsional springs (gains) of the posture controller. At equilibrium, the amount of pitch and roll will be dependent on the CoM error and on the stiffness of the posture controller. This behaviour can be exploited to identify the real CoM coordinates, first to identify the x and y component and subsequently to identify the z component. Indeed considering x and y components, it can be proven (not reported in this work for sake of space) that whenever the joint velocities are zero there is a one-to-one mapping (linear) relationship between the orientation error e_o and the estimation error on the CoM ($c - \hat{c} = f(e_o) = Me_o$), where:

$$c_{x,y} - \hat{c}_{x,y} = \begin{bmatrix}\tilde{c}_x \\ \tilde{c}_y\end{bmatrix} = \underbrace{\begin{bmatrix}0 & -1 \\ 1 & 0\end{bmatrix}\frac{IK_{p_{\text{ang}}}}{mg}}_{M} e_o, \quad (12)$$

with orientation error $e_o^T = \begin{bmatrix}-\phi & -\theta\end{bmatrix}^T$ (we set a zero orientation reference) and with ϕ and θ the measurements of roll and pitch respectively, coming from an inertial sensor (e.g. IMU). $K_{p_{\text{ang}}} \in \mathbb{R}^{2\times 2}$ is the torsional stiffness set in the trunk controller and $I \in \mathbb{R}^{2\times 2}$ is the rotational inertia of the robot in the roll and pitch directions. Furthermore, one can easily see from Fig. 3 (left) that a deviation towards the x direction leads to (negative) pitching, while an error in the y direction leads to rolling. Instead of applying (12) in a single one-shoot correction, to improve robustness to sensor noise and tracking errors, we can take reduced step corrections in a recursive fashion. Therefore, we can write a recursive update

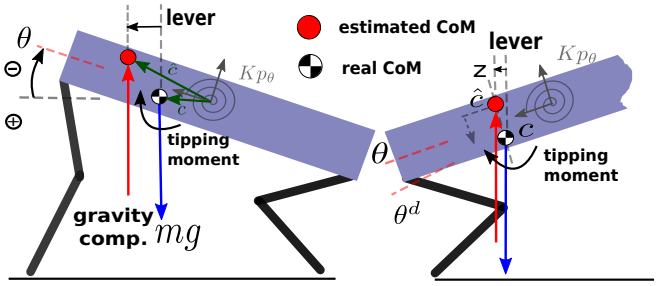

Fig. 3. Lateral view: sketch of the gravity/gravity compensation force in the identification of x,y (left) and of the Z (right) components of c in approach (b).

for each direction to make our estimated CoM $\hat{c}$ converge to the real CoM c:[2]

$$\hat{c}_{x,y}(k+1) = \hat{c}_{x,y}(k) + \beta(c - \hat{c}_{x,y})(k) \quad (13)$$

that, considering the linear mapping in 12, becomes:

$$\begin{bmatrix}\hat{c}_x \\ \hat{c}_y\end{bmatrix}(k+1) = \begin{bmatrix}\hat{c}_x \\ \hat{c}_y\end{bmatrix}(k) + \alpha_{x,y}\begin{bmatrix}-\phi \\ -\theta\end{bmatrix}(k) \quad (14)$$

Rewriting equation (13) we can see that $\hat{c}_{x,y}$ has a first order discrete dynamics $\hat{c}_{x,y} = \beta/(z-(1-\beta))c_{x,y}$ with a settling time $t_a = \frac{-3T_s}{ln(1-\beta)}$, where T_s is the sampling time. Therefore the gains $\alpha_{x,y} = M\beta$ can be tuned to have a trade-off between low convergence time (high α) and filtering capability of high frequency disturbances (low α).

The identification procedure consists of two phases. Using (14) the x and y CoM components can be updated, until the pitch/roll errors converge to (0, 0). Now independent of the z component, the tipping moment is zero and the gravity compensation force is perfectly aligned to the real gravity. Then, to find the z component, the identified x and y CoM components are fixed and a non-zero reference orientation, e.g. (0, θ^d) is set. For an incorrect z component, the robot will pitch either too much or too little with respect to θ^d (see sign conventions in Fig. 2). Using an update rule analogue to (13) given by

$$\hat{c}_z(k+1) = \hat{c}_z(k) - \alpha_z\left(\theta_d - \theta\right)(k), \quad (15)$$

the z component will be moved up or down (in the base frame, see Fig. 3(right)), until convergence $\hat{c}_z = c_z$ is achieved.

Remark 1: Equations (13) and (15) are written for the robot CoM c and not for the trunk CoM c_{tr} as we intended. However, assuming proper identification of the leg inertial parameters *offline*, we can claim that only the modeling error in the trunk CoM is responsible for the modeling error in the robot CoM. In particular, an advantage of using a recursive update (beyond robustness), is that it allows us to replace, without any loss of generality, the robot CoM $\hat{c}$ with the trunk CoM $\hat{c}_{\text{tr}}$, having the update happening only in the trunk parameters. For a similar reason, the update can be performed in the base frame rather than in the world frame.

[2]This is somewhat similar to a gradient descent optimization where β is a scalar representing the step size.

Remark 2: Any other controller (e.g. a joint PD controller) that possibly would fight with the trunk controller should be switched off, because it would create bias body moments that would affect the trunk angular dynamics thus jeopardizing the identification.

C. Identification of trunk CoM using contact forces

The approach presented in the previous Section (III-B) is appealing because it depends only on inertial measurements. However, in practice, since it relies on a whole-body posture controller and (at a lower level) on torque controllers, its performance can be affected by joint torque tracking errors. An alternative approach, only valid to identify the x and y components (in the world frame), is to check the values of the contact forces at the feet. We know that in static conditions, the equilibrium of moments (and forces) should hold:

$$\dot{h} = I_w \dot{\omega} + \dot{I}_w \omega = \sum_{i=1}^{N} ({}_w p_{i,com} \times {}_w f_i) = 0 \tag{16}$$

where the moments are computed about the CoM point. The idea here, is to "*move*" the CoM until the values of the contact forces satisfy (16). Recalling (10) we can rewrite it as

$$-\underbrace{[\sum_{i=1}^{N} ({}_w f_i)]_\times}_{{}_w F} {}_w\hat{c} = \underbrace{\sum_{i=1}^{N} ({}_w p_i \times {}_w f_i)}_{{}_w \mu}. \tag{17}$$

We want to find an estimate ${}_w\hat{c}$ of ${}_w c$ for which (17) is valid. This happens only if ${}_w\hat{c} = {}_w c$. If this is not the case (e.g. we have a modeling error), we can exploit (17) to create a recursive update for ${}_w\hat{c}$ given by

$${}_w\hat{c}(k+1) = {}_w\hat{c}(k) + \alpha(-[{}_w F(k)]^+_\times {}_w\mu(k) - {}_w\hat{c}(k)) \tag{18}$$

where $[.]^+$ is a damped pseudo-inversion and $\alpha << 1$ is a tunable parameter. This update rule will make ${}_w\hat{c}$ converge to the real robot CoM position. This is equivalent to make a Recursive Least Square fitting. The difference of (17) from 10 is that here in (17) we are considering only one sample to do the estimation. Note that this is not possible in one go, as $[{}_w F]_\times$ is not full rank (the cross product eliminates the component parallel to $\sum_{i=1}^{N} f_i$ which in quasi-static conditions is mg). This means it is not possible to estimate all the 3 components of ${}_w c$. In particular, if the trunk is horizontal we cannot appropriately reconstruct the ${}_w c_z$ component from the vector ${}_w F$, as it is aligned to gravity. We therefore perform the update only on the x and y components. The advantage of this approach is that it is independent from the type of controller used to regulate the orientation (e.g. it can also be a stiff joint space position controller). The drawback is that, the identification accuracy depends on the accuracy of the contact force estimation (e.g. $f = -J_c^+(S^T\tau - h(q,\dot{q}))$); thus it is sensitive to torque offsets and modeling errors. Similarly to what previously highlighted, without loss of generality, the approach is valid also if we apply the update rule (18) only to the trunk CoM $\hat{c}_{\text{tr}}$ and in the base frame. Furthermore, rewriting the update rule in the base frame, does not take into account that some directions could be unobservable (e.g. along gravity). Therefore, to avoid errors it is important not to perform the update along that direction, thus a selection matrix E_{xy} for the x and y components is used to obtain

$$c_{\text{tr}} = c_{\text{tr}} + \alpha_w R_b^T E_{xyw} R_b(-[F]^+_\times - \hat{c}), \tag{19}$$

where ${}_w R_b$ is a rotation matrix from base to world frame.

IV. Results: Offline Identification

As introduced in Section II we performed the offline identification of the leg parameters by swinging one leg in the air (without contact) while keeping the trunk of the robot attached to a frame. To be able to identify masses, CoM and inertias, we designed rich *identification* trajectories that have enough accelerations.[3] We designed sinusoidal signals as reference joint positions. High velocities and accelerations could be achieved either by increasing the amplitude or the frequency. We preferred to increase the amplitude over frequency to avoid jerky movements[4] that can excite the elasticity of the mechanical structure, that we do not model. To better explore the workspace we shifted the origin of the sinusoidal joint trajectories in different positions that we selected to maximize the gravity torque shift. Finally, to have richer data, we randomly sampled the frequency, amplitude and phase shift of the 3 joint sinusoids.

A. Simulation Results

To generate a *Simulation* dataset useful to assess the convergence of the *offline* identification algorithm, we modified the original CAD parameters, adding (Gaussian) zero-mean parametric noise on the links CoM positions and inertia tensor, with standard deviations 0.02m and 0.02 kgm^2, respectively. Then, we moved the right hind (RH) leg according to the designed *identification* trajectory, and recorded joint torques and link accelerations, velocities and positions. For the *identification* trajectory, the frequency ranged between 0.3 and 0.9Hz, the phase shift between 0 and π and the amplitudes between $(0, 0, 0)$ and $(0.25, 0.20, 0.4)$rad for the 3 joints, respectively. After collecting a 180 second dataset, we estimated all the leg parameters in one shot using (5). The regressor matrix Y at each time step was computed using the iDynTree software library [8].

Table I compares the identified parameters for the RH leg to the simulated CAD parameters. [5] For the RH leg the observable and partially observable parameters converge to the simulated values with a 1mm accuracy. For the unobservable ones (red) the noiseless CAD values are maintained. At most a rank of 12 was obtained, meaning 12 out of 30 link parameters were observable.

[3]Without acceleration (e.g. using just static poses) we would be able to identify only the first moment of inertia $c \times m$ for each link and we could not discriminate the mass.

[4]The jerk of a sinusoid scales linearly with its amplitude, but cubically with its frequency.

[5]The inertia tensors are expressed in the link frame because only in that frame the dynamic equations are linear.

TABLE I

OFFLINE IDENTIFICATION OF RH LEG IN SIMULATION. COLORS REPRESENT PARAMETER OBSERVABILITY: RED-UNOBSERVABLE, GRAY-PARTIALLY OBSERVABLE, GREEN-OBSERVABLE.

Hip assembly	**mass**	**CoM x**	**CoM y**	**CoM z**
Simulated	2.9300	0.0025	-0.006	0.1228
Identified	2.9300	0.0033	-0.0079	0.1693
Upper limb				
Simulated	2.6380	0.1409	0.0510	-0.0246
Identified	2.6363	0.1436	0.0512	-0.0088
Lower limb				
Simulated	0.8810	0.1361	-0.0131	0.0212
Identified	0.8485	0.1368	-0.0127	-0.0215

Hip assemb.	$\mathbf{I_x}$	$\mathbf{I_{xy}}$	$\mathbf{I_{xz}}$	$\mathbf{I_y}$	$\mathbf{I_{yz}}$	$\mathbf{I_z}$
Simulated	0.1423	0.0188	-0.0651	0.1313	-0.0141	-0.0093
Identified	0.1347	0.0	-0.0227	0.1442	-0.0001	0.0059
Upper limb						
Simulated	0.0019	0.0230	-0.0009	0.1117	-0.0139	0.0900
Identified	-0.0121	0.0231	-0.0139	0.0363	0.0134	0.0933
Lower limb						
Simulated	-0.0152	0.0117	-0.0050	0.0360	0.0133	0.0246
Identified	-0.0145	0.0121	-0.0052	0.0996	-0.0138	0.0254

Remark: We were not able to achieve proper observability of the masses for the 3 links. As the CoM is computed from the product $c{\cdot}m$ as seen in (3), this poses significant problems in the estimation of the link CoM. More investigation should be done in this direction.

Furthermore, we used a dataset for *identification* and a second one for *validation*. With the identified parameters ϕ_{id} from the first set, we predict the torques for the *validation* set by $\tau_{\text{id}} = \bar{Y}\phi_{\text{id}}$. Moreover, the error between predicted torques and measured torques is computed as $e_{id} = \tau_{\text{meas}} - \tau_{\text{id}}$. For comparison the torque is also predicted using the CAD data $\tau_{\text{CAD}} = \bar{Y}\phi_{\text{CAD}}$. Again prediction error $e_{\text{CAD}} = \tau_{\text{meas}} - \tau_{\text{CAD}}$ is computed and shown in Fig. 4. The improvement in torque prediction for the HAA joint given by the identified parameters with respect to the CAD parameters is clearly observed.

B. Experimental results

We appropriately processed the experimental data before using it for the identification. In particular, all variables were zero-delay low-pass filtered and data samples where velocity of the three links was smaller than 0.01rad/s were removed to mitigate the effect of stiction. Results are shown in Fig. 5. Since we do not have a base line (i.e. we do not know what the real parameters are), the effectiveness of the approach is demonstrated by how well the torque τ_{id} of the HAA joint, computed with the identified parameters, matches the measured torque τ_{meas}. For comparison we also report the prediction error of the CAD parameters.

The prediction error for the HAA joint is below 1Nm in the case of the identified model, while for the CAD model it is 3 times as large (e.g. below 3Nm). If one would implement an inverse dynamics controller based on the CAD model, this error would be obtained. Despite the torque is predicted quite well for the HAA and HFE there is some discrepancy for the KFE joint. One possible reason is the low mass and inertias of the lower leg link (0.88kg). As a result the torque in the KFE joint (around $3/4$Nm) is very small with respect to the measuring range of the sensor (150Nm) that also has quite small resolution (0.5% of the range). This significantly reduces the signal to noise ratio.

V. RESULTS: ONLINE IDENTIFICATION

First the trunk mass was estimated from contact forces using (7). For sake of brevity, numerical results are omitted. However, we found that even small trunk accelerations have a large effect on the estimated mass, so we excluded data where velocity and acceleration where non-zero. Moreover,

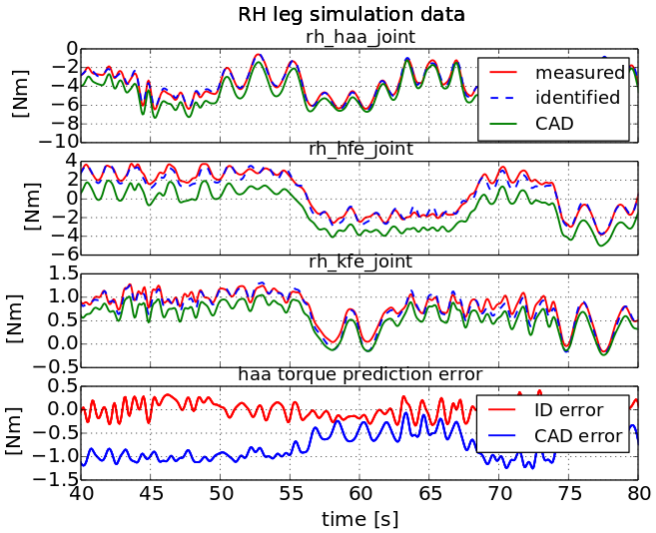

Fig. 4. Simulation data. RH leg. The first 3 plots are for the torques for the 3 leg joints: HAA, HFE, KFE. Red: Measured joint torques τ_{meas}, blue: predicted torques calculated with the identified parameters τ_{id}, green: predicted joint torques calculated with the CAD parameters τ_{CAD}, Fourth plot compares the two prediction errors: e_{id} and e_{CAD} for the HAA joint.

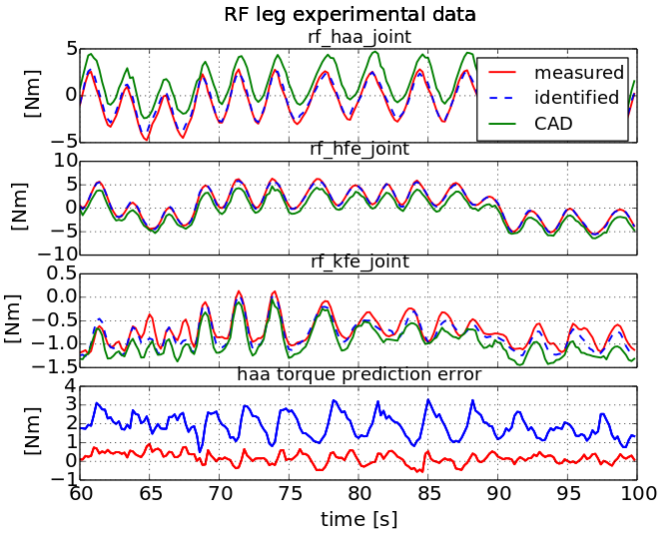

Fig. 5. Experimental data. RF leg. The first 3 plots are for the torques for the 3 leg joints: HAA, HFE, KFE. Red: Measured joint torques τ_{meas}, blue: predicted torques calculated with the identified parameters τ_{id}, green: predicted joint torques τ_{CAD} calculated with the CAD parameters. Fourth plot compares the two prediction errors: e_{id} and e_{CAD} for the HAA joint.

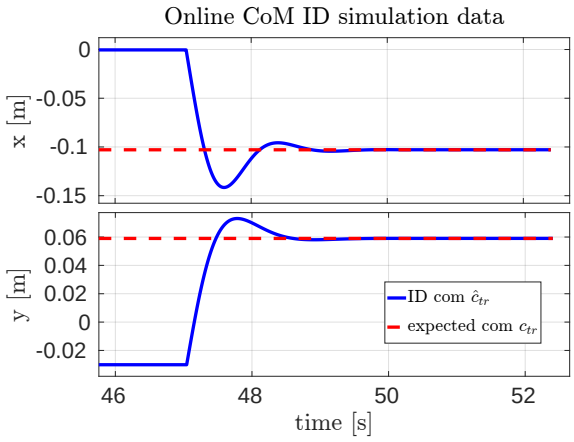

Fig. 6. Simulation of online identification with approach (c) after a virtual load of 100N was applied at $(-0.75, 0.4, 0.0)$m w.r.t. the base frame.

as HyQ has no contact force sensors at the feet, the ground reaction forces are estimated from joint torques through the equation $f = -J_c^+(S^T\tau - g)$. Since J_c depends on the leg kinematics, also small kinematic errors are significantly affecting mass estimation. In the following results only the x and y component of the CoM are considered as for quasi-static locomotion these are most important. Moreover, the shift in z component is insignificant for the payload we use.

A. Simulation results

We performed several simulations: in the first one we added a virtual mass to the trunk at $(-0.75, 0.4, 0.0)$m w.r.t. the base frame, which caused the trunk CoM to shift. In simulation we compared the online identification approaches (b) (Section III-B) and (c) (Section III-C) with the static one (a) (Section III-A). We repeated the experiment for 3 different loading condition: 0, 50, 100N. In all we deliberately started with a large error in the CoM estimation $\hat{c}_{\text{tr}}$ used in the controllers. As an error measure we computed the Euclidean distance between the identified and real CoM as $||\delta|| = \sqrt{(c_x - \hat{c}_x)^2 + (c_y - \hat{c}_y)^2}$. In table II we show that all the 3 approaches converge coherently to the same ground truth values for the x and y direction, with a 0.5cm accuracy. Furthermore, Fig. 6 shows that, with the approach (c), the estimated CoM $\hat{c}_{\text{tr}}$ converged to the real trunk CoM c_{tr} within 2 seconds with an accuracy less than 2mm. For approach (a) and (b) we also performed the identification of the z component (not reported in the table) finding an error from the ground truth of 2cm and 1cm, respectively.

Since in real world scenario the terrain is not always flat, we investigated the robustness of approach (c) by letting the robot stand on different ramp inclinations (0, 10, 15) degrees and added different loading conditions (0, 50, 100)N. Again the Euclidean distance in x and y dimension is used as error measure. Results in Table III demonstrate accurate performance, i.e. $<$1cm error.

B. Sensitivity analysis

The goal of this section is to evaluate the sensitivity (and thus the robustness) of the proposed (b) and (c) approaches when real world uncertainties are present, such as kinematic errors and torque offsets. We set Gaussian parametric noise to the joint torque measurements and feet positions. For different noise characteristics, namely mean(μ) and standard deviation (σ), convergence error is reported in Table IV We see that approach (b) is mostly affected by torque offsets while approach (c) suffers more from kinematic errors.

C. Experimental results

We implemented an online *load change* detection strategy to detect possible payload changes. Each time velocity and acceleration approach zero (static equilibrium), the robot mass is verified. We performed several experiments with the real robot by placing a 50 and 100N mass at a frame attached to the left-hind corner torso, while the robot was walking at 10cm/s. The walking framework is a statically stable crawl framework that we use for rough terrain locomotion. The core module is a state machine (see [13] for details) that switches between two temporized/event-driven locomotion phases: a swing phase, and a body motion phase. During the *body motion phase* the robot CoM is shifted onto the future support triangle, which is opposite to the next swing leg, in accordance to a user-defined foot sequence. [6] The walk without identification failed at the first step because the CoM, shifted by the payload, went out of the support polygon. Subsequently, thanks to the online identification, the robot was able to continue walking steadily. In detail, after the detection of load change, the robot stopped to start the online identification procedure. After the new trunk CoM c_{tr} was identified (e.g. when the $\Delta c = c - \hat{c}$ was getting below a certain tolerance which we set to 0.0005m with $\alpha = 0.002$) the values were set both in the trunk controller and in the planner for the generation of the future trajectories. As predicted in Simulation (see Section V-B) approach (b) happened to be more sensitive to torque tracking errors. Therefore, due to tracking inaccuracies, it was taking quite

[6] A video of the experiments is available at https://youtu.be/2Pnj5hJVeJM

TABLE II

COMPARISON OF ONLINE COM IDENTIFICATION ERRORS FOR APPRAOCHES (A), (B) AND (C) WITH DIFFERENT LOADS

		Loads		
ID apprach		**0 N**	**50 N**	**100 N**
approach (a)	$\|\|\delta\|\|$ [cm]:	0.010	0.357	0.513
approach (b)	$\|\|\delta\|\|$ [cm]	0.020	0.049	0.452
approach (c)	$\|\|\delta\|\|$ [cm]:	0.006	0.125	0.586

TABLE III

ONLINE COM IDENTIFICATION ERRORS WITH APPROACH (C) FOR DIFFERENT SLOPES AND LOADS

		Loads		
Slope		**0 N**	**50 N**	**100 N**
0°	$\|\|\delta\|\|$ [cm]:	0.082	0.104	0.864
10°	$\|\|\delta\|\|$ [cm]:	0.294	0.621	0.904
15°	$\|\|\delta\|\|$ [cm]:	0.429	0.528	0.773

TABLE IV

SENSITIVITY OF APPROACHES (B) AND (C) TO NOISE ON TORQUE OFFSETS AND FOOT POSITION MODELING ERROR.

Torque offset [Nm]	**Noise char.**	μ=0, σ=1	μ=10, σ=1	μ=20, σ=1
approach(b)	$\|\|\delta\|\|$ [cm]:	0.410	1.114	2.098
approach(c)	$\|\|\delta\|\|$ [cm]:	0.073	0.190	0.255
Foot pos. [cm]	**Noise char.**	μ=0, σ=1	μ=0, σ = 2	μ=0, σ=3
approach(b)	$\|\|\delta\|\|$ [cm]:	0.001	0.060	0.202
approach(c)	$\|\|\delta\|\|$ [cm]:	0.308	2.661	5.074

TABLE V

FOR 3 LOAD CONDITIONS THE IDENTIFIED x AND y COMPONENT OF THE TRUNK COM ARE COMPARED FOR APPROACHES (C) AND (A)

Payload	**appr. (c)** [m]	**appr. (a)** [m]	$\|\|\delta\|\|$ [cm]
0 N	[0.007, 0.015]	[0.005, 0.011]	0.45
50 N	[-0.024, 0.017]	[-0.024, 0.015]	0.20
100 N	[-0.047, 0.023]	[-0.048, 0.020]	0.32

some time to converge to a stable value. Conversely, we were able to successfully perform the identification with approach (c). Because we did not have a ground truth we compare the convergence results to the batch least-square approach (a). Using (11) the new CoM was estimated for the three payloads (0, 50 and 100N). The results are reported in Table V. In the accompanying video we also show successful experiments of a walk on moderately rough terrain (e.g. stepping stones). Furthermore, we performed separate experiments to estimate the z component exploiting (15) but this turned out to be too sensitive to pitch orientation noise. Indeed, the update is done for $\theta_{des} - \theta$ which can be very small. Further work should be done to improve this.

VI. CONCLUSIONS

We presented a whole identification pipeline for a quadruped robot. The identification is performed offline for the leg parameters (mass, CoM and inertial tensors) and online for the trunk, which is subject to more frequent payload changes. After the *offline* leg parameters identification, experimental data showed that the predicted torques were matching quite well with the measured ones. For the *online* trunk parameter identification, we proposed two distinct recursive approaches, which where compared to a batch least squares approach. Both of them produced similar results in simulation by converging with good accuracy to the ground truth value of CoM even in presence of rough terrains and slopes. The approach (b) (Section III-B) resulted to be more sensitive to the quality of the torque tracking. Conversely, approach (c) (Section III-C) resulted more flexible because it was independent from the type of controller used, but more sensitive to estimation errors in the ground reaction forces. We also analyzed the sensitivity of the proposed approaches to some parametric uncertainties. Finally, in experimental trials, the online identification was crucial to have the robot successfully walk even when suffering big payload changes (obtained by adding a 10 kg mass on the trunk).

In the future, we plan to solve the joint mass observability issue, by exploiting the whole robot floating-base equations for the estimation of the leg parameters. Also, we will use appropriate *scaling* to address the issue of the different order of magnitude between the inertia of the trunk and of leg links. Last, future extensions will involve the possibility to perform the online identification without stopping the robot at all, by including velocities and accelerations into the framework.

REFERENCES

[1] K. Ayusawa, G. Venture, and Y. Nakamura, "Identifiability and identification of inertial parameters using the underactuated base-link dynamics for legged multibody systems," *The International Journal of Robotics Research*, vol. 33, no. 3, pp. 446–468, 2013.

[2] A. Herzog, L. Righetti, F. Grimminger, P. Pastor, and S. Schaal, "Experiments with a hierarchical inverse dynamics controller on a torque-controlled humanoid," in *IEEE International Conference on Robotics and Automation (submitted to)*, 2014.

[3] S. Feng, X. Xinjilefu, W. Huang, and C. G. Atkeson, "3D Walking Based on Online Optimization," *Humanoid Robots, 2013 13th IEEE-RAS International Conference on*, 2013.

[4] S. Kuindersma, F. Permenter, R. Tedrake, and A. Bu, "An Efficiently Solvable Quadratic Program for Stabilizing Dynamic Locomotion," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2589–2594, 2014.

[5] J. Buchli, M. Kalakrishnan, P. Pastor, and S. Schaal, "Compliant quadruped locomotion over rough terrain," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009.

[6] M. Mistry, S. Schaal, and K. Yamane, "Inertial parameter estimation of floating base humanoid systems using partial force sensing," in *9th IEEE-RAS International Conference on Humanoid Robots*, pp. 492–497, 2009.

[7] C. D. Sousa and R. Cortesão, "Physical feasibility of robot base inertial parameter identification: A linear matrix inequality approach," *The International Journal of Robotics Research*, vol. 33, no. 6, pp. 931–944, 2014.

[8] F. Nori, S. Traversaro, J. Eljaik, F. Romano, A. Del Prete, and D. Pucci, "iCub Whole-Body Control through Force Regulation on Rigid Non-Coplanar Contacts," *Frontiers in Robotics and AI*, vol. 2, no. March, pp. 1–18, 2015.

[9] J. Wu, J. Wang, and Z. You, "An overview of dynamic parameter identification of robots," *Robotics and Computer-Integrated Manufacturing*, vol. 26, no. 5, pp. 414–419, 2010.

[10] P.-B. Wieber, R. Tedrake, and S. Kuindersma, "Modeling and Control of Legged Robots," *Springer Handbook of Robotics*, pp. 1–61, 2015.

[11] P. de Santos, E. Garcia, and J. Estremera, *Quadrupedal locomotion*. springer ed., 2006.

[12] B. U. Rehman, M. Focchi, M. Frigerio, J. Goldsmith, G. C. Darwin, and C. Semini, "Design of a Hydraulically Actuated Arm for a Quadruped Robot," in *Proceedings of the International Conference on Climbing and Walking Robots (CLAWAR)*, 2015.

[13] M. Focchi, A. Del Prete, I. Havoutis, R. Featherstone, D. G. Caldwell, and C. Semini, "High-slope terrain locomotion for torque-controlled quadruped robots," *Autonomous Robots*, pp. 1–14, 2016.

[14] J. J. E. Slotine and L. Weiping, "On the adaptive control of robot manipulators," *International Journal of Robotics Research*, vol. 6, no. Copyright 1988, IEE, pp. 49–59, 1987.

[15] H. Berghuis, H. Roebbers, and H. Nijmeijer, "Experimental comparison of parameter estimation methods in adaptive robot control," *Automatica*, vol. 31, no. 9, pp. 1275–1285, 1995.

[16] J. Z. Kolter, P. Abbeel, and A. Y. Ng, "Hierarchical apprenticeship learning with application to quadruped locomotion," in *Neural Information Processing Systems*, 2008.

[17] B. J. Stephens, "State estimation for force-controlled humanoid balance using simple models in the presence of modeling error," *IEEE International Conference on Robotics and Automation*, pp. 3994–3999, 2011.

[18] C. Semini, N. G. Tsagarakis, E. Guglielmino, M. Focchi, F. Cannella, and D. G. Caldwell, "Design of {HyQ} – a hydraulically and electrically actuated quadruped robot," *Journal of Systems and Control Engineering*, 2011.

[19] C. G. Atkeson, C. H. An, and J. M. Hollerbach, "Estimation of Inertial Parameters of Manipulator Loads and Links," 1986.

[20] M. Gautier, "Numerical calculation of the base inertial parameters of robots," *Journal of robotic systems*, vol. 8, no. 4, pp. 1020–1025, 1991.