Contact Observer for Humanoid Robot Pepper based on Tracking Joint Position Discrepancies

Anastasia Bolotnikova, Sébastien Courtois, Abderrahmane Kheddar

Contact Observer for Humanoid Robot Pepper based on Tracking Joint Position Discrepancies

Anastasia Bolotnikova[1,2], Sébastien Courtois[1], Abderrahmane Kheddar[2]

Abstract— In order to enable efficient control of a human-humanoid in physical contact settings, a real-time solution for a contact observer is required. We propose a novel approach for proprioceptive sensor based contact sensing suitable for affordable personal robots with no force/torque or electric current sensing. We combine robot model knowledge and the output of acceleration resolved quadratic programming whole-body controller to make a prediction of expected position tracking error for computing our proposed contact observer signal. We demonstrate the efficiency of our approach in the experiments of contact detection and estimation of collision direction and intensity on a real humanoid robot Pepper platform controlled by a task-space multi-objective quadratic programming controller.

I. INTRODUCTION

The aim of our work is to enable real-time contact sensing for Pepper robots for use in the robot control for physical Human-Robot Interaction (pHRI). The use of force/torque sensors is not available on current Pepper platforms. Thus the contact sensing shall be addressed by using only the sensors which are available on Pepper, mainly joint encoders.

Pepper robot (Fig. 1), produced by Soft Bank Robotics, is often presented in public, and people, especially children, express great interest to interact with it. So far, meaningful physical interaction was limited to only few tactile sensors. Often, however, people excitedly touch various robot links, while robot can express no reaction as it remains "unaware" of these contacts. Enabling robust whole-body contact observer for Pepper can potentially bring the pHRI experience to a whole new level with meaningful robot reaction to various external contacts and ability to interact with the environment through taking contacts and applying forces, not to mention that this functionality can serve as a safety reflex when human touch can damage robot structure or *vice versa* [1].

The topic of contact sensing is particularly challenging for low-cost personal robots, such as Pepper, where the embedded sensors are limited and the design mechanics and kinematics do not obey high precision requirements. We describe an approach which aims to overcome those limitations and enable whole-body contact sensing for Pepper.

We review the progress made in the proprioceptive sensor based contact sensing in recent years and outline why existing approaches could not be adapted for Pepper platform (Sec. II). We present our proposed methodology (Sec. III, Sec. IV). We demonstrate the performance of the proposed

[1]SoftBank Robotics Europe, Paris, France
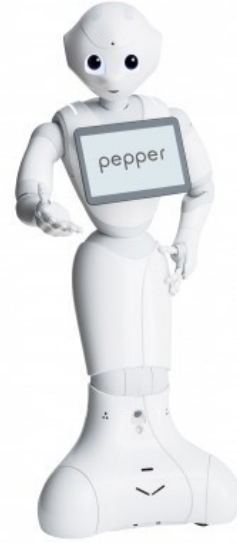[2]University of Montpellier–CNRS LIRMM, Interactive Digital Humans, Montpellier, France

Fig. 1: Our target platform: Pepper humanoid robot that is widely used in costumer service and research spheres.

approach (Sec. V) and conclude the paper with discussion on current limitations and future work (Sec. VI).

Our main contributions are the following:

1) We derive a formula for expected tracking error computation for a DC motor controlled with PD scheme;
2) We describe the process of non-linear system identification for expected tracking error prediction based on the knowledge of desired trajectory and robot model;
3) With the ability to predict which part of the position tracking error is related to the normal collision-free motion, we propose a novel contact observer signal, which incorporates direction and intensity information of the collision event;
4) We perform experiments with a Pepper platform and demonstrate high sensitivity of our proposed contact observer and good performance of contact detection and identification of collision direction and intensity.

II. BACKGROUND

Several methods have been proposed for proprioceptive sensor based contact sensing. The overview of various techniques, namely the direct estimation of the external torque, energy, velocity and momentum observers, is well documented in the survey paper on robot collisions [2], where the collision event pipeline is also introduced. In our

work, we focus on three main phases of the collision event pipeline: (i) detection– *did collision occur*?, (ii) isolation– *where on the robot collision occurred*?, (iii) identification– *what is the direction and intensity of the collision*?

Among all methods, presented in the survey, the best performing one proved to be the momentum observer [3], as it can effectively address all three main phases of the collision event pipeline. This method has been extended for the application on floating base (humanoid) systems in [4]. The momentum observer has been also augmented to include common non-linear effects, namely large backlash and friction, commonly encountered on low-cost platforms [5]; updated momentum observer was implemented and tested on the Romeo robot arm, which is also produced by Soft Bank Robotics. Another interesting work have addressed reconstruction of the interaction forces in static conditions; it was implemented and tested on a small humanoid robot from Soft Bank Robotics NAO [6].

Initially, the momentum observer was introduced for a single contact isolation, however, it was also used as a base for the multi-contact isolation method in [7]. The momentum observer can be used more efficiently in combination with force sensor measurements, when a force/torque sensor is installed either at the robot base for fixed platforms [8] or when force sensors are present on some of the many robot links, as was done for humanoid Atlas in [9]. Such extensions of the momentum observer, however, are not applicable to low-cost robots due to the lack of force sensing devices on the platform, mainly because of their cost and the logistics they require.

There are several reasons, why the classical momentum observer (that we investigated and tried) cannot be efficiently applied to platforms like Pepper robot:

1) Motor-side friction is significant and will appear as an external torque in the residual vector unless friction compensation is appropriately implemented;
2) Significant motor-link backlash and flexibility in some joints violates the assumption that motor and joint angles coincide and consequently torque from the motor is not always well transmitted to the link-side.
3) Motor torque, τ_m, which is usually estimated from electric current and current to torque constants, cannot (for now) be exploited from Pepper due to current measurements being absolute and down-sampled; if this issue could be resolved in the future, it can be an adding to our presented method.

We could use the method developed in [5] to overcome friction and backlash; yet it requires having two encoders per joint. As for now, we do not have access to Pepper motor-side encoders to measure and account for the backlash.

Given the aforementioned constraints, we address the whole-body contact observer for Pepper by means of monitoring the difference between measured position tracking error and predicted expected position tracking error (i.e. without external torques during free or static motions) given known robot's model and "intentions" (desired setpoint tasks such as given postures). In our work Pepper is

controlled by acceleration resolved quadratic programming controller (QP) [10], [11], which we use to compute desired acceleration (and subsequently velocity and position by numerical integration) and joint torque for a given motion task. We use those quantities to make a prediction of the expected tracking error value.

In the next section we present the detailed developments of our approach for expected tracking error prediction and contact observer signal computation.

III. Proposed contact observer method

We are challenged to use only position tracking error to extract the collision event information: intensity, direction and link. However, we assume the condition of having a compliant (low PD gains with or without feedforward terms) semi-reversible or totally reversible actuators. This assumption holds in case of the Pepper platform, which was designed to be safe (low gains) and semi-reversible, hence inherently compliant.

In static settings, when the tracking error value does not vary significantly and remains small, the contact monitoring based on tracking error is trivial –collision changes the tracking static error and reveals the intensity (i.e. stronger collision causes larger deviation of the static error), direction (positive or negative deviation) and link (last joint in the chain with tracking error exceeding the given threshold) information, assuming the contact direction is such that it causes joint displacement (which doesn't happen if the direction of the contact force is normal to the joint's axis).

When the robot moves, the problem becomes more complex –the tracking error is not constant as in the static case; its dynamics (i.e. the increase or decrease of the tracking error) is not always caused by collision. Indeed, it is due to the fact that because of the dynamics (inertia, Coriolis...) and the posture configuration (w.r.t gravity) each joint might not yet reach desired steady-state position. In this case we need to be able to distinguish when tracking error means collision and when it simply means free joint motion dynamics.

In order to define a tracking error based contact observer signal, we eliminate from the tracking error the part that refers to normal/expected joint motion and leave only the part of the tracking error which is caused by a collision. In order to achieve that, we identify the relationship between our intention in terms of desired trajectory and expected tracking error. In the following one degree of freedom toy example, we show the rational ground that drives our reasoning. Consider a DC motor regulated by a proportional-derivative (PD) controller, with gains K_p and K_d respectively. Subtracting joint position sensor measurements q from desired joint position target q_d gives the tracking error value $\epsilon = q_d - q$. We use ϵ as a feedback in the PD controller to compute desired voltage input u (Eq. 1).

$$u = K_p \epsilon + K_d \dot{\epsilon} \qquad (1)$$

The electric equation of a DC motor is given as Eq. 2

$$u - e = L\frac{di}{dt} + Ri \tag{2}$$

where e is the back-electromotive force that is proportional to the motor speed ω with proportionality constant K_e (Eq. 3).

$$e = K_e\omega \tag{3}$$

The motor torque is proportional to the electrical current i with proportionality constant K_t (Eq. 4).

$$\tau_m = K_t i \tag{4}$$

The dynamic equation of the motor is given as Eq. 5

$$\tau_m = J_m\dot{\omega} + \mu\omega + \tau_l \tag{5}$$

where J_m is the motor inertia, μ is the motor friction constant and τ_l is the load torque that includes motor-link friction and backlash effects (Eq. 6)

$$\tau_l = M(q)\ddot{q} + c(q,\dot{q}) + \tau_f + \tau_B - \tau_{\text{ext}} \tag{6}$$

here M is the load inertia and c combines Coriolis, centrifugal and gravity forces. Following the friction modeling principles, used for the Romeo arm in [5], τ_f depends on the motor-link backlash and is expressed as Eq. 7

$$\tau_f = \begin{cases} K_{\phi v}\dot{\phi} + K_{\phi s}\text{sign}(\dot{\phi}) + \\ \quad K_{qv}\dot{q} + K_{qs}\text{sign}(\dot{q}) & |\phi| < \alpha \\ K_{\theta v}\dot{\theta} + K_{\theta s}\text{sign}(\dot{\theta}) & \text{otherwise} \end{cases} \tag{7}$$

where θ is the motor angle, $\phi = q - \theta$ is the difference between the joint and the motor angles, α is the size of the motor-link backlash gap and $K_{\{x\}v}, K_{\{x\}s}$ are viscous and static friction coefficients respectively.

When $|\phi| < \alpha$ the motor is moving inside the backlash gap, hence, no torque is transfered between the motor and the load in this case (i.e. $\tau_l = 0$). Otherwise, the motor is in contact with one of the borders of the backlash gap and the load is moving together with the motor. The τ_B term in Eq. 6 is a spring-damper regularization term to model the effect due to the motor-link backlash interaction (Eq. 8)

$$\tau_B = \begin{cases} -M(q)\ddot{q} - c(q,\dot{q}) - \tau_f + \tau_{\text{ext}} & |\phi| < \alpha \\ K_{\phi p}(\phi + \alpha) + K_{\phi d}\dot{\phi} & \phi \geq \alpha \\ K_{\phi p}(\phi - \alpha) + K_{\phi d}\dot{\phi} & \phi \leq -\alpha \end{cases} \tag{8}$$

Here, the inside backlash equation (i.e. $|\phi| < \alpha$ case) simply nulls the load torque as seen from the actuator; $K_{\phi p}$ and $K_{\phi d}$ are the stiffness and damping coefficients. Note that both τ_f and τ_B can be thought of as $\xi(q, \dot{q}, \theta, \dot{\theta}, \alpha)$ for simplicity.

And finally, τ_{ext} in the Eq. 6 is any other external torque, especially that caused by a collision on the link.

Substituting (3) and (4) into (2) gives Eq. 9.

$$u = K_e\omega + R\frac{\tau_m}{K_t} \tag{9}$$

We neglect $L\frac{di}{dt}$ term due to its relative insignificance compared to e and Ri.

Finally, substituting (1) and (5) into (9) gives the analytical relation between tracking error ϵ and the external torque applied on the load τ_{ext} (Eq 10).

$$K_p\epsilon + K_d\dot{\epsilon} = \frac{R}{K_t}J_m\dot{\omega} + \left(\frac{R}{K_t}\mu + K_e\right)\omega + \\ + \frac{R}{K_t}\left(M(q)\ddot{q} + c(q,\dot{q}) + \tau_f + \tau_B - \tau_{\text{ext}}\right) \tag{10}$$

Same relation can be derived for other types of control schemes (PID, PI, etc.) in analogous form.

In the contact observer context, this relation can be exploited in the following ways. First, it shows that by measuring $q, \dot{q}, \ddot{q}, \omega, \dot{\omega}, \epsilon$ and $\dot{\epsilon}$ and knowing robot model (M, c), motor properties (R, K_t, K_e, μ, J_m) and controller gains (K_p, K_d), as well as all other constants present in Eq. 10, we can directly compute τ_{ext} without necessity to measure motor torque or electric current.

Secondly, assuming the motion of the load free of external collisions, i.e. $\tau_{\text{ext}} = 0$ we can use Eq. 10 in order to compute expected (under free motion assumption) tracking error ϵ_{exp} from the value of desired position, speed and acceleration of the load $q_d, \dot{q}_d, \ddot{q}_d$ (Eq. 11).

$$\epsilon_{\text{exp}} = \frac{RJ_mN}{K_tK_p}\ddot{q}_d + \left(\frac{R}{K_t}\mu + K_e\right)\frac{N}{K_p}\dot{q}_d + \\ + \frac{R}{K_tK_p}\underbrace{\left(M(q_d)\ddot{q}_d + c(q_d,\dot{q}_d) + \tau_f + \tau_B\right)}_{\text{desired load torque } \tau_{ld}} - \frac{K_d}{K_p}\dot{\epsilon}_{\text{exp}} \tag{11}$$

where N is the gear reduction ratio ($\omega = N\dot{q}$). Yet, in the presence of significant backlash, it is more accurate to model ω as a function of $\dot{q}$ that also includes the motor-link backlash effect ($\omega = \xi(\dot{q}, \phi, \alpha)$).

Finally, Eq. 10 tells us that tracking error has direct relation to the external torque and thus can be potentially used to reconstruct external collision forces.

In our work we cannot directly evaluate Eq. 11, because we do not know precisely R, K_t, K_e, J_m, μ and we cannot compute $\dot{\epsilon}_{\text{exp}}$ before computing ϵ_{exp}. Additionally, we do not have the access to the motor side encoder to measure θ, ω and $\dot{\omega}$, which would allow us to handle the backlash appropriately and compute τ_f and τ_B terms.

We, thus, choose to identify a non-linear model to approximate Eq. 11 using a set of available desired motion related variables, namely $\dot{q}_d, \ddot{q}_d$ and τ_{ld}. We select a binary-tree prediction model [12]. The non-smooth activation function of a binary-tree nonlinearity estimator is suitable in our particular case, because it is capable of modeling sudden abrupt changes in the tracking error signal, unlike other nonlinearity estimators with smooth activation function, such ass sigmoid or wavelet networks [13], which we also experimented with.

The final form of the $\tilde{\epsilon}_{\text{exp}}$ expression is Eq. 12

$$\tilde{\epsilon}_{\text{exp}}(t) = \texttt{binary\_tree}(\dot{q}_d, \ddot{q}_d, \tau_{ld}) \tag{12}$$

With the identified model that is capable to predict expected tracking error for every robot joint with sufficient accuracy, we can compute the part of tracking error that is related only to the collision. We do that by subtracting predicted tracking error value from the measured tracking error to compute our contact observer signal r:

$$r = \epsilon - \epsilon_{\text{exp}} \tag{13}$$

In the following section we describe how model identification process is performed and what is the resulting performance of expected tracking error prediction.

IV. Model identification for expected tracking error prediction

In this section we describe the process of model identification for the expected tracking error prediction (Eq. 12).

Consider the left shoulder roll joint (*LSRoll*) for this example. In order to identify a model for expected tracking error prediction for *LSRoll*, we record the data free of external collisions while controlling Pepper via the QP controller with a single posture task in the objective function that generates a sequence of various motions including moving between the joint bound limits with randomly selected small or big offsets and moving the joint to various random setpoints. In the middle of each data acquisition process the configuration of a previous joint (*LSPitch*) and a next joint (*LERoll*) in the chain change to new randomly selected setpoints and the main joint, *LSRoll*, repeats the motion sequence again. With such setup our intention is to identify the model which is "aware" of the change in the configuration of other joints. Note that this only becomes possible due to the desired load torque, τ_{ld}, being one of the inputs to the nonlinearity estimator as it incorporates the robot model knowledge. The joint stiffness value is set to 100% for all joints in our experiments.

For the estimation dataset, we record 7 different sequences of *LSRoll* joint motion with various QP posture task stiffness values of the QP posture task. Posture task stiffness varies from 2 to 5 in estimation dataset recordings. This data is used to identify the parameters of a binary tree and to evaluate accuracy of tracking error reconstruction on estimation dataset. Average resulting accuracy of tracking error prediction over 7 data sequence recordings used in the estimation process is 77.29%.

Now that the satisfactory performance on the estimation dataset is achieved, we evaluate performance of this model on "unseen" test data. We execute the same QP controller but with new randomly selected offset and setpoint parameters and increase significantly the stiffness of the QP posture task, setting it to 9, in order to trigger motion with higher speed (and thus larger tracking error). Total accuracy of tracking error prediction on test data set is 81.11%. The plot of a segment of the joints trajectories from this experiment (test dataset) and error reconstruction plot are shown in Fig. 2.

From results in Fig. 2 it is evident that the identified prediction model generalizes well to unseen data and accurately predicts the value of the position tracking error. We see that
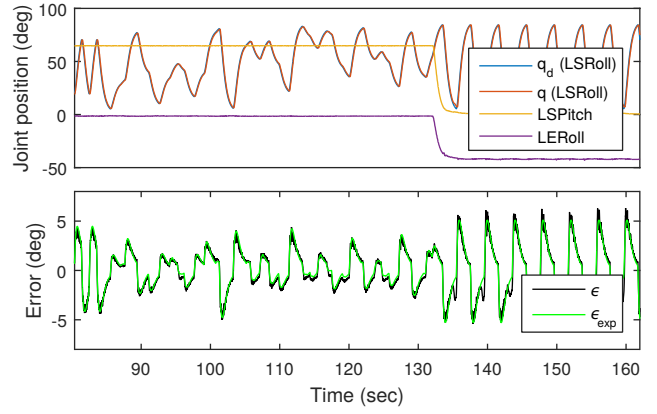

Fig. 2: Evaluation of the identified binary tree model for the expected tracking error prediction on test data: joint trajectories (top); expected tracking error prediction for *LSRoll* joint (bottom). Overall tracking error prediction accuracy: 81.11%.

the model is robust to changes of the QP task stiffness and changes of the configuration of other joints in the chain.

For the best performance, model identification has to be performed for every motor separately and possibly repeatedly, as the motors and the gear system wear out with time. However, we noticed that left body side joint's model can perform equally well for the right side, as the same motor/load types are used and they wear out approximately equally. Even more so, we observed that the same motor types can "share" a model. Pepper has 17 joints and uses 5 different types of motors in total. Thus, in general, it is sufficient to perform the model identification only 5 times.

In the next section we show how our system performs when applied to the data sequence with external collisions.

V. Experimental Results

We now show that our proposed contact observer signal r is suitable for the contact detection and identification of contact direction and intensity.

We apply a median filter over 11 latest samples of r to reduce noise and eliminate occasional spikes in the signal. We set a fixed threshold $\delta = 2.5°$ for the contact detection. Whenever $|r| > \delta$ we consider that a collision/contact occurred. The threshold δ can also be interpreted as an external torque sensitivity threshold, meaning that any external force, which results in such a τ_{ext} at the joint that causes the displacement beyond δ, can be detected. We show that using our proposed method, even light collisions ($> 2.5°$) can be detected. The sign and magnitude of r reveal the direction and collision intensity information respectively.

We use same QP controller to generate sequence of left arm joints motion with new randomly selected parameters for offsets and setpoints. We set QP posture task stiffness to 8. During the execution of the motion several external collisions are triggered by touching the robot's left arm. The plot in Fig. 3a shows a $\sim$20 second segment of the results from this experiment. We repeat a similar experiment for the right

(a) *LSRoll experiment segment*
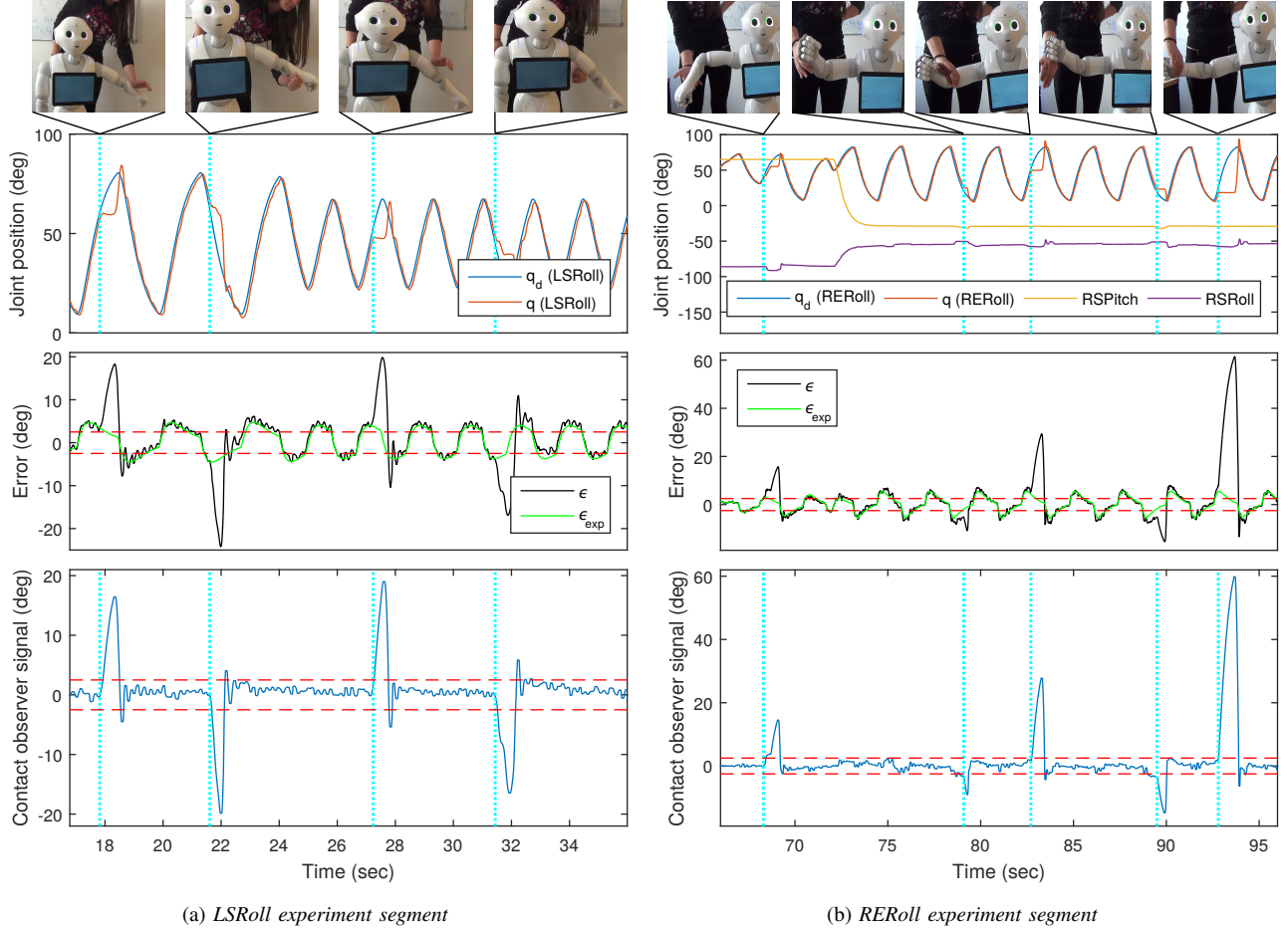
(b) *RERoll experiment segment*

Fig. 3: Model evaluation: joint trajectories (top); expected tracking error prediction (middle); contact observer signal r (bottom). Dashed blue lines show the start of the contacts. Dashed red lines indicate the threshold for contact detection.

arm elbow roll joint *RERoll* with posture task stiffness set to 12. A segment of *RERoll* experiment results is presented in Fig. 3b. The extended presentation of these results can be seen in the video accompanying this paper.

The results indicate, yet again, that our proposed method is capable of making precise prediction of expected tracking error and, thus, produce a contact observer signal r which remains below threshold δ when there is no collision, i.e. when the joint moves freely. When collision occurs (indicated in the plots by dashed vertical blue lines), r exceeds the fixed threshold. Moreover, the direction and the intensity information about the collision event is correctly represented via the sign and magnitude of r.

The middle plot of the Fig. 4 demonstrates the segment of experiment where the binary tree model trained using recordings of *LSRoll* joint's data is applied to predict expected tracking error for the right shoulder roll joint (*RSRoll*). This result confirms that models identified on left side body joints generalize well for the right side body joints, eliminating the necessity to train separate models for every robot joint.

The Tab. I reports the total amount of false positive #FP ($r > \delta$ without contact), false negative #FN ($r \leq \delta$ with contact) and true positive #TP ($r > \delta$ with contact) contact detections across three experiments.

Experiment name	#FP	#FN	#TP
LSRoll experiment	2	2	18
RERoll experiment	0	3	19
RSRoll (with LSRoll model)	0	3	18
Total:	**2**	**8**	**55**

TABLE I: Number of false positive, false negative and true positive contact detections across three experiments

Note that usually after strong external collision, when joint quickly returns to its desired position, r exceeds the threshold for a brief amount of time due to the impact. Such cases of exceeding the threshold are not considered as false positives in Tab. I. We also note that most of the false negative cases reported in Tab. I occurred due to the large flexibility of the hip-roll joint, which enables the upper body of Pepper to move when an arm is pushed/pulled. This reduces the amount of the position tracking error in the arm joints and can lead to a false negative contact detection. Of course, this issue would be resolved when the whole body (including the floating wheeled part) is considered.
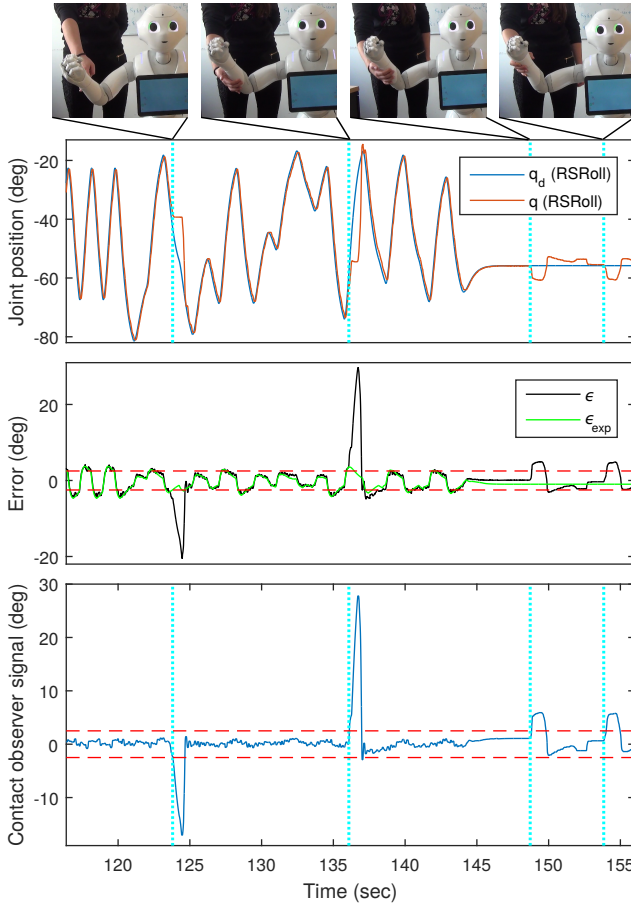
Fig. 4: The model identified using *LSRoll* joint sample data applied to the test data of *RSRoll* joint. The models of the left body side generalize well for the right body side joints.

VI. CONCLUSION AND FUTURE WORK

We have derived a simplified expression for computing the expected value of the position tracking error of a DC motor controlled with PD scheme given the knowledge of desired joint trajectory and desired load torque. This expression revealed that, under some conditions of compliance (low PD gains with feedforward terms and/or reversibility) the expected tracking error prediction does not require knowledge of neither the motor current nor the motor torque.

We described the process of non-linear model identification and presented the results of expected tracking error prediction, which show good accuracy and generalization properties of the identified models. We demonstrated how prediction of expected tracking error can be used for computing the contact observer signal, which incorporates intensity and direction information of the collision event.

In its current form, our proposed approach still exhibits some false positive contacts detections. In our next development stages, we intend to reduce the amount of false positive detections by making the thresholding process more robust by using, for instance, some type of adaptive thresholding technique as a second step after contact observer signal computation.

In continuation of this work we will adapt the proposed method to the case of complex and arbitrary whole-body motions including the floating base (i.e. considering the robot with all its degrees of freedom). In doing so, we will also determine if any specific handling of hip roll and hand joints flexibility and friction of brushless DC motors in the Pepper's leg is required and can be implemented.

We will also investigate thoroughly the relation between our contact observer signal r and the value of the external torque τ_{ext}. We intend to approach it by comparing, first in static and later also in dynamic conditions, the values of estimated τ_{ext} (computed either from the difference in electric current measurements or estimated on the test-bench that includes motor torque sensors) with the values of contact observer signal r. Once this is done, it would become possible to reconstruct the force which is causing the collision from r without ever measuring or estimating motor torque or the motor electric current.

And finally, our ultimate goal is to develop and test our proposed approach for its integration in the feedback signal of an adaptive control for pHRI in motion assistance scenarios and active compliance to touch.

REFERENCES

[1] A. Kumar Pandey, R. Gelin, R. Alami, R. Viry, A. Buendia, R. Meertens, M. Chetouani, L. Devilliers, M. Tahon, D. Filliat, Y. Grenier, M. Maazaoui, A. Kheddar, F. Lerasle, and L. F. Duval, "Romeo2 project: Humanoid robot assistant and companion for everyday life: I. situation assessment for social intelligence," in *International Workshop on Artificial Intelligence and Cognition*, pp. 140–147, 2014.

[2] S. Haddadin, A. D. Luca, and A. Albu-Schäffer, "Robot collisions: A survey on detection, isolation, and identification," *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1292–1312, 2017.

[3] A. D. Luca and R. Mattone, "Sensorless robot collision detection and hybrid force/motion control," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 999–1004, 2005.

[4] F. Flacco, A. Paolillo, and A. Kheddar, "Residual-based contacts estimation for humanoid robots," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pp. 409–415, 2016.

[5] F. Flacco and A. Kheddar, "Contact detection and physical interaction for low cost personal robots," in *IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, 2017.

[6] T. Mattioli and M. Vendittelli, "Interaction force reconstruction for humanoid robots," *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 282–289, 2017.

[7] L. Manuelli and R. Tedrake, "Localizing external contact using proprioceptive sensors: The contact particle filter," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5062–5069, 2016.

[8] G. Buondonno and A. De Luca, "Combining real and virtual sensors for measuring interaction forces and moments acting on a robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 794–800, 2016.

[9] J. Vorndamme, M. Schappler, and S. Haddadin, "Collision detection, isolation and identification for humanoids," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4754–4761, 2017.

[10] K. Bouyarmane and A. Kheddar, "Using a multi-objective controller to synthesize simulated humanoid robot motion with changing contact configurations," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4414–4419, 2011.

[11] K. Bouyarmane, J. Vaillant, K. Chappellet, and A. Kheddar, "Multi-robot and task-space force control with quadratic programming," *IEEE Transactions on Robotics*, "submitted".

[12] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and regression trees*. Chapman and Hall/CRC, 1984.

[13] Q. Zhang and A. Benveniste, "Wavelet networks," *IEEE transactions on Neural Networks*, vol. 3, no. 6, pp. 889–898, 1992.