

# UKF-Based Sensor Fusion for Joint-Torque Sensorless Humanoid Robots

Ines Sorrentino<sup>1,2</sup>, Giulio Romualdi<sup>1</sup>, Daniele Pucci<sup>1,2</sup>

**Abstract**—This paper proposes a novel sensor fusion based on Unscented Kalman Filtering for the online estimation of joint-torques of humanoid robots without joint-torque sensors. At the feature level, the proposed approach considers multi-modal measurements (e.g. currents, accelerations, etc.) and non-directly measurable effects, such as external contacts, thus leading to joint torques readily usable in control architectures for human-robot interaction. The proposed sensor fusion can also integrate distributed, non-collocated force/torque sensors, thus being a flexible framework with respect to the underlying robot sensor suit. To validate the approach, we show how the proposed sensor fusion can be integrated into a two-level torque control architecture aiming at task-space torque-control. The performances of the proposed approach are shown through extensive tests on the new humanoid robot ergoCub, currently being developed at Istituto Italiano di Tecnologia. We also compare our strategy with the existing state-of-the-art approach based on the recursive Newton-Euler algorithm. Results demonstrate that our method achieves low root mean square errors in torque tracking, ranging from 0.05 Nm to 2.5 Nm, even in the presence of external contacts.

## I. INTRODUCTION

Humanoid robots have emerged as versatile tools with applications ranging from assistive devices to complex human-robot interactions. Precise control of these robots is crucial for replicating human-like movements and ensuring safe interactions in dynamic environments [1], [2], [3]. Among the control strategies, torque control has gained significant attention for effectively operating in unpredictable situations, providing robust motion tracking, and ensuring compliance with unexpected contacts [4], [5], [6]. Nevertheless, the implementation of this control strategy requires knowledge of the effective joint torques. When joint torque sensors are unavailable, the accurate estimation of these torques becomes a fundamental requirement to achieve high performance in torque tracking and dynamic tasks. This paper proposes a joint torque estimation strategy to enable torque control on sensorless humanoid robots to guarantee compliance to unexpected disturbances.

Many implementations of torque-based whole-body control algorithms have been developed for robots equipped with joint torque sensors [5], [7], [8]. However, integrating these sensors into humanoid robots presents challenges due to space constraints, costs, mechanical complexity, and potential problems like measurement drift and joint elasticity [9],

\*The paper was supported by the Italian National Institute for Insurance against Accidents at Work (INAIL) ergoCub Project

<sup>1</sup> Artificial and Mechanical Intelligence, Italian Institute of Technology, Genoa, Italy, (e-mail: name.surname@iit.it)

<sup>2</sup> School of Computer Science, University of Manchester, Manchester, UK

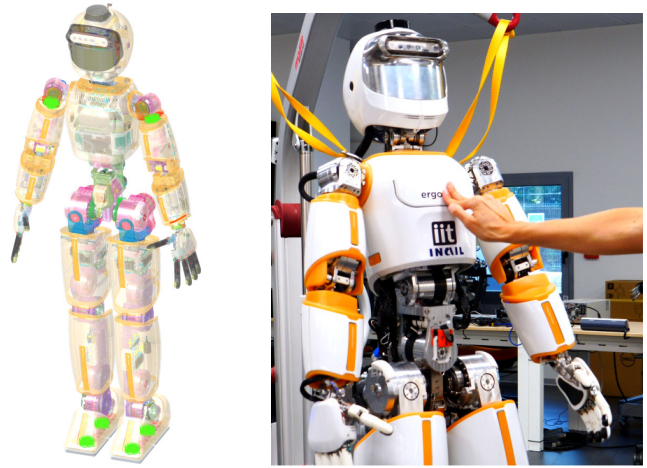


Fig. 1: On the left The ergoCub humanoid robot in transparency with FT sensors highlighted in green. On the right ergoCub being pushed on the torso during a closed-loop torque controller test.

[10]. Consequently, numerous research teams have designed their robot platforms without joint torque sensors. In these scenarios, torque control is often replaced by a stiff position control strategy, and dynamic tasks are satisfied under specific assumptions based on the application, such as flat terrain or absence of obstacles [11], [12], [13]. To overcome these limitations and enable joint torque control on such robots, the challenge is accurately estimating the joint torques.

One of the most common approaches for torque estimation on humanoid robots is based on the classical recursive Newton-Euler algorithm (RNEA). The RNEA approach uses the robot dynamic model to propagate the measured wrenches from the force/torque (FT) sensors along the robot kinematic chain. However, this deterministic estimation method neglects effects like measurement noises, miss-calibrated sensors, or modeling uncertainties. Consequently, torque estimation often does not achieve the desired performance levels, especially when dealing with external contacts, which can only influence the estimate when applied after the FT sensors in the kinematic chain [14], [15], [16].

A more recent approach overcomes the problem of considering external contacts by estimating joint torques from the motor current, motor angular velocity, and motor angular acceleration measured in each joint [17]. This method remains deterministic and relies on prior knowledge of certain coefficients, including the motor torque constant, motor moment of inertia, and the coefficient of friction

for the harmonic drives. These coefficients have a direct impact on the algorithm outcomes, but measuring them can be challenging. Additionally, the deterministic nature of this approach overlooks modeling uncertainties, and the accuracy of joint torque estimation depends entirely on how effectively the model identification procedure is executed.

This paper contributes towards the design of a novel joint torque estimator with the aim of enabling torque control on humanoid robots without joint torque sensors and mounting high-ratio harmonic drives. More precisely, unlike classical deterministic algorithms, our proposed methodology is based on Unscented Kalman Filtering [18] and sensor fusion [19] to combine data acquired from distributed sensors, including joint encoders, FT sensors, inertial sensors, and motor current sensors. This approach is employed to address non-measurable factors, such as external contacts, modeling uncertainties, and measurement noise. The joint torque estimation algorithm is tested in combination with a two-layers torque control architecture to track references in the joint or cartesian spaces. The high-level control law is implemented as a constrained quadratic programming problem and generates the reference joint torques. The low-level torque controller is implemented as a Proportional-Integral (PI) feedback-feedforward control with a friction compensation action to reduce energy losses. Experiments on the humanoid robot ergoCub (see Fig. 1) demonstrate the effectiveness of our approach. A comparison with the state-of-the-art RNEA algorithm for torque estimation shows improvements in terms of estimation accuracy, in particular in the presence of external contacts.

The paper is organized as follows. Sec. II introduces the notation and recalls some concepts of floating-base systems. Sec. III details the joint torque estimation. Sec. IV presents the validation results on the position-controlled humanoid robot ergoCub. Sec. V concludes the paper.

## II. BACKGROUND

### A. Notation

- $I_{m \times n}$  and  $0_{m \times n}$  denote the  $m \times n$  identity and zero matrices respectively; when  $m = n$  we use  $I_n$  and  $0_n$ .
- $\mathcal{I}$  denotes the inertial frame.
- ${}^{\mathcal{I}}p_B$  is a vector connecting the origin of frame  $\mathcal{I}$  and the origin of frame  $\mathcal{B}$  expressed in frame  $\mathcal{I}$ .
- given  ${}^{\mathcal{I}}p_B$  and  ${}^{\mathcal{B}}p_C$ ,  ${}^{\mathcal{I}}p_C = {}^{\mathcal{I}}R_B {}^{\mathcal{B}}p_C + {}^{\mathcal{I}}p_B = {}^{\mathcal{I}}H_B \begin{bmatrix} {}^{\mathcal{B}}p_C \\ 1 \end{bmatrix}$ .  ${}^{\mathcal{I}}H_B$  is the homogeneous transformation and  ${}^{\mathcal{I}}R_B \in SO(3)$  is the rotation matrix.
- ${}^{\mathcal{B}}v_{\mathcal{I},\mathcal{B}} = \begin{bmatrix} {}^{\mathcal{B}}v_{\mathcal{I},\mathcal{B}} \\ {}^{\mathcal{B}}\omega_{\mathcal{I},\mathcal{B}} \end{bmatrix} \in \mathbb{R}^6$  is the velocity of the frame  $\mathcal{B}$  with respect to the frame  $\mathcal{I}$  expressed in the frame  $\mathcal{B}$ , where  ${}^{\mathcal{B}}v_{\mathcal{I},\mathcal{B}}$  denotes the linear velocity and  ${}^{\mathcal{B}}\omega_{\mathcal{I},\mathcal{B}}$  the angular velocity. For the sake of simplicity, in the rest of the paper, we use  $v_B$ .
- $\times$  is the cross product on  $\mathbb{R}^3$ .
- $g_B$  is the gravity vector expressed in  $\mathcal{B}$ .
- $\alpha_S = s \dot{v}_{I,S} - \begin{bmatrix} ({}^S R_{\mathcal{I}}^T v_S) \times s \omega_{\mathcal{I},S} \\ 0_{3 \times 1} \end{bmatrix}$  is the conversion rule between the acceleration of a sensor with frame  $\mathcal{S}$  with

respect to the inertial frame and the sensor acceleration  $\alpha_S$  expressed in the sensor frame.

- $i_m \in \mathbb{R}^n$  is the vector of motor currents with  $n$  the number of motors.
- $f_B^\top = \begin{bmatrix} f_B^\top & \mu_B^\top \end{bmatrix}$  is the wrench acting on a point of a rigid body expressed in the frame  $\mathcal{B}$ .
- the *hat symbol* identifies the variables estimated through the Unscented Kalman Filter.
- For clarity, all the sensor measurements will be expressed in the sensor frames.

### B. Humanoid Robot Model

A humanoid robot is a floating base multi-body system composed of  $n + 1$  links connected by  $n$  joints with one degree of freedom each. The configuration of the robot is defined by considering both the joint positions  $s$  and the homogeneous transformation from the inertial frame to the robot base frame  $\mathcal{B}$ . The configuration is identified by the triplet  $q = ({}^{\mathcal{I}}p_B, {}^{\mathcal{I}}R_B, s) \in \mathbb{R}^3 \times SO(3) \times \mathbb{R}^n$ . The velocity of the system is characterized by the set  $\nu = ({}^{\mathcal{I}}v_B, {}^{\mathcal{I}}\omega_B, \dot{s})$  where  ${}^{\mathcal{I}}\omega_B$  is the angular velocity of the base frame expressed with respect to the inertial frame,  $\dot{s}$  is the time derivative of the joint positions.

The equation of motion of a multibody system is described by applying the *Euler-Poincaré Formalism* [20]

$$M(q)\dot{\nu} + h(q, \nu) = B\tau_j + \sum_k J_k(q)^\top f_{ext,k}, \quad (1)$$

where  $M \in \mathbb{R}^{(6+n) \times (6+n)}$  is the mass matrix,  $h(q, \nu) \in \mathbb{R}^{6+n}$  accounts for the Coriolis, centrifugal and gravitational effects,  $\tau_j \in \mathbb{R}^n$  are the joint torques,  $B \in \mathbb{R}^{(6+n) \times n}$  is the selection matrix,  $f_{ext,k} \in \mathbb{R}^6$  is the  $k$ -th contact wrench expressed in the contact frame,  $J_k(q)$  is the Jacobian associated to the  $k$ -th contact wrench.

### C. Model of the harmonic drive mechanical friction

Humanoid robots are often equipped with harmonic drives having high reduction ratios [21], [22]. The performance of harmonic drives is influenced by significant non-linear effects such as kinematic error, flexibility, hysteresis, and friction [23], [21]. Among these, friction is the predominant factor responsible for dissipating a large part of the torque generated by the input current to the motors. Consequently, it significantly influences the accuracy of joint torque estimation. The effective joint torque can be calculated as

$$\tau_j = r\tau_m - \tau_F, \quad (2)$$

where  $r$  is the gear-ratio,  $\tau_m$  is the motor torque, and  $\tau_F$  is the friction torque. We approximate the friction model through the combination of Coulomb and viscous friction models. To avoid discontinuities at zero velocity, we introduce the hyperbolic tangent function into our model [24]:

$$\tau_F = k_0 \tanh(k_1 \dot{s}) + k_2 \dot{s}. \quad (3)$$

The identification of the coefficients  $k_0$ ,  $k_1$ ,  $k_2$  is discussed in Section IV.

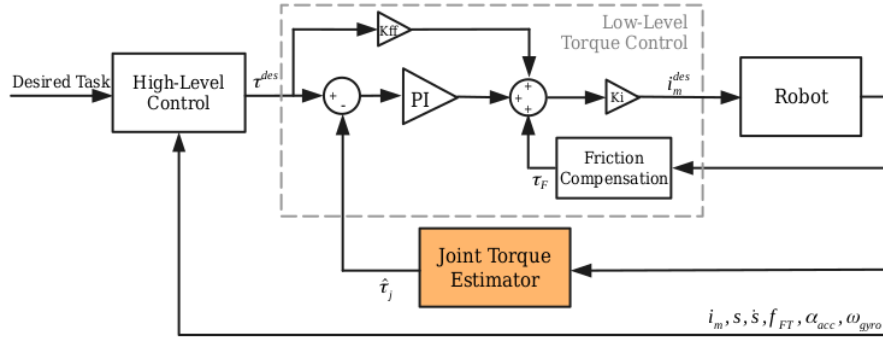


Fig. 2: Block diagram of the controller architecture implemented on the ergoCub humanoid robot.

### III. UKF BASED STOCHASTIC SENSOR FUSION FOR JOINT TORQUES

In this section, we describe the estimation algorithm for the robot joint torques  $\tau_j$ , by combining data from the physical sensors on the robot through sensor fusion. Because the process and measurement models are non-linear, we formulate this problem as an Unscented Kalman Filter [18]. Unlike traditional approaches that require linearization [25], the UKF provides accurate state estimates without requiring this linearization step. Nonetheless, the UKF still depends on statistical information regarding the noise affecting both the state and measurements, specifically their covariance.

#### A. Unscented Kalman Filter definition

We adopt a Kalman Filter to fuse the sensor data and estimate the system state. The discrete-time nonlinear dynamic system is written as

$$\begin{aligned} x_{k+1} &= F(x_k, u_k, v_k) \\ y_k &= G(x_k, u_k, w_k) \end{aligned} \quad (4)$$

where

$$\begin{aligned} x_k &= \left[ \hat{s}^\top \quad \hat{\tau}_m^\top \quad \hat{\tau}_F^\top \quad \hat{f}_{FT}^\top \quad \hat{f}_{ext}^\top \right]_k^\top \\ y_k &= \left[ \dot{s}^\top \quad i_m^\top \quad f_{FT}^\top \quad \alpha_{acc}^\top \quad \omega_{gyro}^\top \right]_k^\top \\ u_k &= \left[ H_B^\top \quad v_B^\top \quad \dot{v}_B^\top \quad s^\top \right]_k^\top. \end{aligned} \quad (5)$$

where  $\hat{s}$ ,  $\hat{\tau}_m$ ,  $\hat{\tau}_F$ ,  $\dot{s}$ ,  $i_m \in \mathbb{R}^n$ , with  $n$  the number of joints.  $\hat{f}_{FT}$ ,  $f_{FT} \in \mathbb{R}^{6m}$  where  $m$  is the number of FT sensors.  $\hat{f}_{ext} \in \mathbb{R}^{6l}$  where  $l$  is the number of external contacts.  $\alpha_{acc} \in \mathbb{R}^{3p}$  is the vector of the linear accelerations measured by the accelerometers, being  $p$  the number of accelerometers.  $\omega_{gyro} \in \mathbb{R}^{3h}$  is the vector of all the angular velocities measured by the gyroscopes, being  $h$  the number of gyroscopes. The process noise  $v_k$  drives the dynamical system, while  $w_k$  is the observation noise. It is worth noting that,  $F$  and  $G$  are non-linear equations.

#### B. Process Model

The list of dynamic equations composing the process model of the UKF is defined as follows.

- *Joint velocity dynamics*: we split the robot model at the FT sensors shown in green in Fig. 1. Then, considering

Eqs. (1) and (2) we define the dynamic equation of each submodel  $i$

$$\begin{aligned} \dot{\nu}^{(i)} &= M(q)^{-1(i)} \left( r\hat{\tau}_m^{(i)} - \hat{\tau}_F^{(i)} - h(q, \nu)^{(i)} \right. \\ &\quad \left. + \sum_k J_{FT,k}^\top \hat{f}_{FT,k} + J_{ext}^\top \hat{f}_{ext} \right), \end{aligned} \quad (6)$$

This allows us to treat FT measurements as external contacts for the sub-models. From  $\dot{\nu}^{(i)}$  we select the joint velocities  $\dot{s}^{(i)}$  of the submodel  $i$ .

- *Friction torque dynamics*: the dynamics of the friction torque is computed as the time derivative of equation (3)

$$\dot{\hat{\tau}}_F = (k_0 k_1 \operatorname{sech}^2(k_1 \dot{s}) + k_2) \dot{s} \quad (7)$$

where  $\dot{s}$  is computed from the forward dynamics (6).

- *Motor torque, FT sensor, and contact wrench dynamics*: the dynamics of the motor torques, FT sensors, and contact wrenches are unknown, and we consider a constant dynamic model  $\dot{\hat{\tau}}_m = 0_{n \times 1}$ ,  $\dot{\hat{f}}_{FT} = 0_{6m \times 1}$ , and  $\dot{\hat{f}}_{ext} = 0_{6l \times 1}$ .

#### C. Measurement Model

The measurement model accounts for all the sensors mounted on the robot, implementing a sensor fusion approach.

- *Joint velocity measurement*:  $\dot{s} = \dot{\hat{s}}$ .
- *Motor current measurement*: the current sensor measurement is related to the state vector as  $i_m = k_\tau^{-1} \hat{\tau}_m$  where  $k_\tau$  is the motor torque constant.
- *FT sensor measurement*:  $f_{FT} = \hat{f}_{FT}$ .
- *Accelerometer sensor measurement*: the accelerometer measures the linear accelerations expressed in the sensor frame. The measurement model of an accelerometer is defined as:

$$\begin{aligned} \alpha_{acc} &= (\dot{J}\nu + J\dot{\nu})_{3 \times 1} - {}^S R_A A g \\ &\quad - ({}^S R_A A \dot{o}_S) \times {}^S \omega_{A,S}. \end{aligned} \quad (8)$$

where all the variables on the right depend on the state  $\dot{\hat{s}}$  and the input  $s$ . The  $3 \times 1$  subscript means that we select the first three elements of the resulting vector.

- *Gyroscope sensor measurement*: the gyroscope measures the angular velocities expressed in the sensor

frame, and the measurement model is defined by selecting the angular part  $\omega_{gyro}$  of the equation  $v_{gyro} = J\nu$  where  $J$  and  $\nu$  depend on the state  $\hat{s}$  and the input  $s$ .

#### D. Covariance tuning

We have performed the following trial-and-error procedure to fine-tune the covariances of the UKF observer. We have generated joint torque trajectories using the high-level controller described in Section IV-A.1 which have been used as references for the ergoCub robot simulated in Gazebo [26]. The joint positions measured during the simulation are adopted as reference trajectories for the joints of the real robot. While executing the trajectory, we log data from all sensors, creating offline datasets to be used as input and measurements for the UKF. This process allows us to tune the process and measurement covariances, ensuring that the UKF accurately estimates joint torques, matching the joint torques coming from the simulation.

### IV. EXPERIMENTS

This section presents the validation results of the joint torque estimation method introduced in Sec. III. We also describe the controller architecture used for validation and the friction identification procedure. The code is available online [27]. We test our algorithm on the ergoCub humanoid robot (Fig. 1) equipped with various sensors including 19 joint encoders, 19 motor current sensors, 6 FT sensors placed 2 under each foot and 1 on each arm respectively, and 7 accelerometers and gyroscopes placed 2 under each foot, 1 on each arm, 1 on the pelvis. The joint torque estimator is seamlessly integrated into the robot's computer, activating upon startup. The results reported in the paper are based on data collected at various time points post-activation, demonstrating robust performance with low tracking errors, even after prolonged usage.

Validation experiments involve placing the robot on a pole to simplify the high-level control architecture, as the necessity to maintain balance is irrelevant to the study. Three experiments are conducted. In two of them, the proposed strategy is compared with a state-of-the-art joint torque estimation based on RNEA [16] to analyze reactions to external disturbances. In the last experiment, we analyze the performance of our method with a more complex task in the cartesian space. All the tests are shown in the supplementary video.

#### A. Controller architecture

The control architecture is shown in Figure 2 and is detailed below.

1) *High-Level Control*: Based on the robot dynamics (1), the high-level control law computes the desired joint torques  $\tau_{des}$  solving a constrained quadratic programming problem [9]. The tasks are handled as cost functions and constraints. This section introduces the list of tasks considered in the control problem.

- Cartesian task: We account for desired link position and orientation with respect to the inertial frame  ${}^A H_L^{des} =$

TABLE I: Friction parameters for ergoCub leg joints.

	$k_0$ (Nm)	$k_1$ (1/rad/sec)	$k_2$ (Nm/rad/sec)
hip pitch	4.9	4.0	0.6
hip roll	4.0	4.7	0.3
hip yaw	2.5	2.6	0.5
knee	2.3	2.7	0.1
ankle pitch	2.3	2.3	0.3
ankle roll	1.3	2.0	0.3

$(p_L^{des}, {}^A R_L^{des}) \in \mathbb{R}^3 \times SO(3)$ . The task is defined at the acceleration level as  $\Psi_{LSE(3)} = [\dot{v}_L^{*\top} \quad {}^A \dot{\omega}_L^{*\top}]^\top - J_L \dot{\nu} - \dot{J}_L \nu$  where  $\dot{v}_L^*$  and  ${}^A \dot{\omega}_L^*$  are set to guarantee almost global convergence of the measured pose to the desired trajectory.

- Joint position regularization task: This task prevents huge variations of the desired joint accelerations. It is described by  $\Psi_s = \ddot{s}^* - [0_{n \times 6} \quad I_n] \dot{\nu}$ .  $\ddot{s}^*$  is defined as  $\ddot{s}^* = \ddot{s}^{des} + k_s^d (\dot{s}^{des} - \dot{s}) + k_s^p (s^{des} - s)$  where  $s^{des}$  is the desired joint trajectory, and  $k_s^d$  and  $k_s^p$  are two positive-defined diagonal matrices.

2) *Low-Level Torque Control*: The joint torques  $\tau^{des}$  generated by the high-level controller described in Section IV-A.1, are sent to a decentralized low-level joint torque controller. The torque controller, shown in Figure 2, is implemented as a feedback-feedforward control and includes the compensation of the mechanical friction introduced in Section II to mitigate energy losses [28]. The control law generates the reference currents to drive the motors.

#### B. Friction parameters identification

The identification of the friction models showed the presence of substantial static and viscous friction between the motor and the load. We have generated reference motor currents to drive the motors and identify the friction model associated with each harmonic drive. Under conditions of no external disturbances, the residual between the torque applied to the motors and the torque at the joint, as calculated in (2), corresponds to the friction forces outlined in equation (3). The friction parameters for the leg joints are reported in Table I. From experimental analysis, we assume negligible friction affecting the torso joints.

#### C. Comparison with the state-of-the-art

We conducted a comparative analysis between our approach and the state-of-the-art method, which relies on RNEA. The setup of our ergoCub robot differs from that of the iCub robot used in the studies proposed in the state-of-the-art [14] only in terms of the location of the FT sensors. Indeed, the FT sensors in the upper part of the legs are absent in the ergoCub robot. To ensure a fair comparison of the performance of the two algorithms under the same working conditions, we focused also on the torso chain which is cut at the FT sensor of the upper arms and the FT sensors of the feet. However, given that the robot is mounted on a pole, the legs do not influence the estimation. The comparison is conducted through two separate experiments, detailed in the

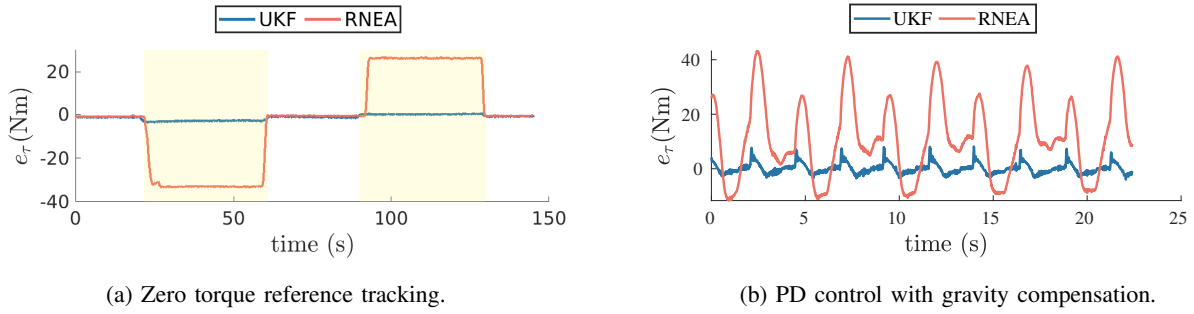


Fig. 3: (a)-(b) Comparison of the joint torque tracking errors obtained with the RNEA and UKF algorithms.

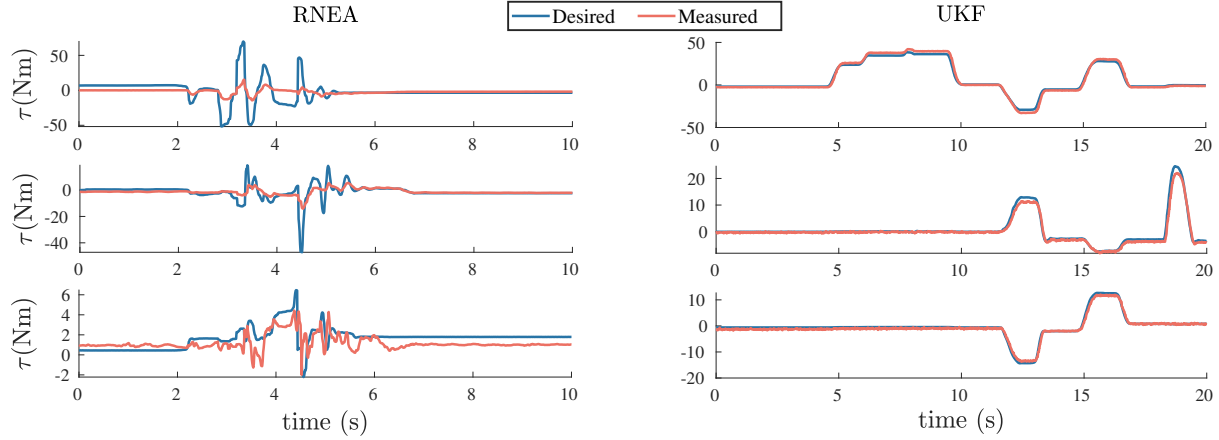


Fig. 4: Comparison of the joint torque tracking for the PD control with gravity compensation experiment on torso joints. On the left are the results obtained with the RNEA algorithm, and on the right results given by the UKF method.

following subsections. Both the RNEA and UKF approaches utilize identical high-level and low-level controller gains.

1) *Zero Torque Tracking*: In this experiment, we focus on controlling one joint in the robot leg to track a zero-torque reference, while other leg joints are position-controlled. An external force, generated by contact with an operator, perturbs the leg links and causes a change in the joint configuration. The external force is not measurable by the FT sensors being located lower in the kinematic chain. Consequently, the classical RNEA is unable to account for this contact in the estimation process. In Figure 3a, we compare the RNEA algorithm (in red) and our UKF method (in blue) in tracking the desired torque ( $\tau^{des} = 0$  Nm). The yellow areas represent the contact phases. RNEA estimates the joint torque only based on leg configuration, resulting in a significant error and causing the controller to react oppositely to the external force direction. Our approach uses motor current measurements to accurately estimate the joint torque with small error (approximately 2.5 Nm), improving control performance and allowing for smoother movement through different joint configurations. Similar enhancements are observed in other leg joints.

2) *PD control with Gravity Compensation*: For this validation, we conducted two separate experiments. In Experiment 1, the controller has the task of tracking a sinusoidal joint trajectory in one of the robot leg joints. In Experiment 2, we control all three joints of the ergoCub torso with the aim

of maintaining their initial configuration. The control method involves PD control with gravity compensation, implemented through the high-level controller described in Section IV-A, by adding a joint position regularization task.

In Experiment 1 the high-level controller generates the necessary torques to achieve the desired joint position, velocity, and acceleration. Figure 3b illustrates the torque tracking error for a single joint in the robot leg. In this experiment, an obstacle is placed in front of the leg, preventing it from fully executing the trajectory. The contact point occurs higher than the FT sensor location in the kinematic chain. As in the previous test, the RNEA algorithm cannot measure the external contact, resulting in a large torque tracking error (Root Mean Square Error (RMSE) = 18.3 Nm). In contrast, our UKF-based approach, integrating current measurements that vary due to external contact, achieves significantly smaller tracking errors (RMSE = 1.96 Nm) than the state-of-the-art solution.

In Experiment 2 we employ the same high-level controller to maintain all three joints of the ergoCub torso in a constant configuration. Random external disturbances are applied, pushing the robot from various contact points. The objective is to prove that the state-of-the-art controller offers a solution that stiffens the robot when contacts occur away from FT sensors. In contrast, our solution ensures compliance with external forces and guarantees the tracking of the desired task once the contact ceases. It is evident from Figure 4

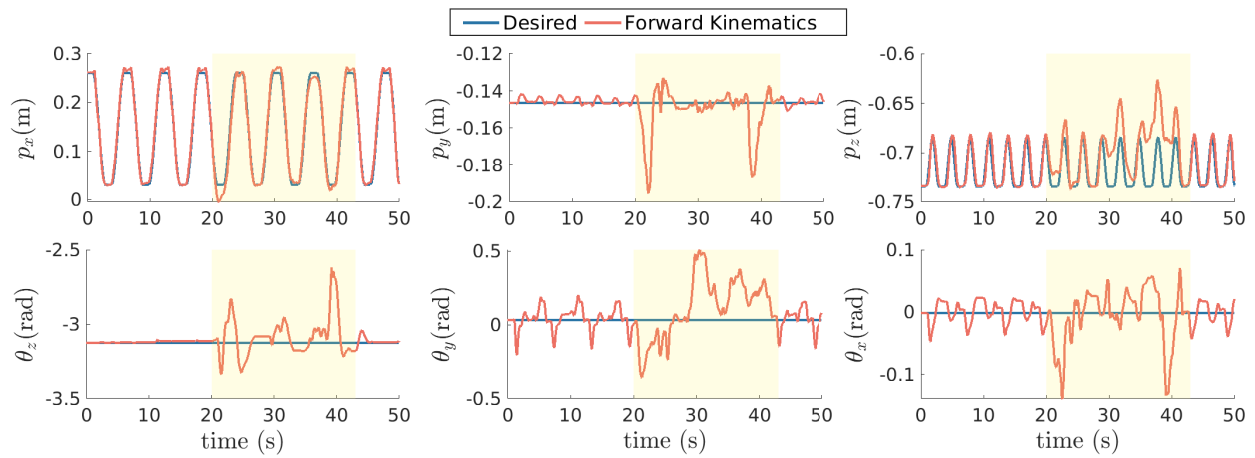


Fig. 5: Tracking of a cartesian task defined for the right sole of ergoCub when the leg is subject to external contacts.

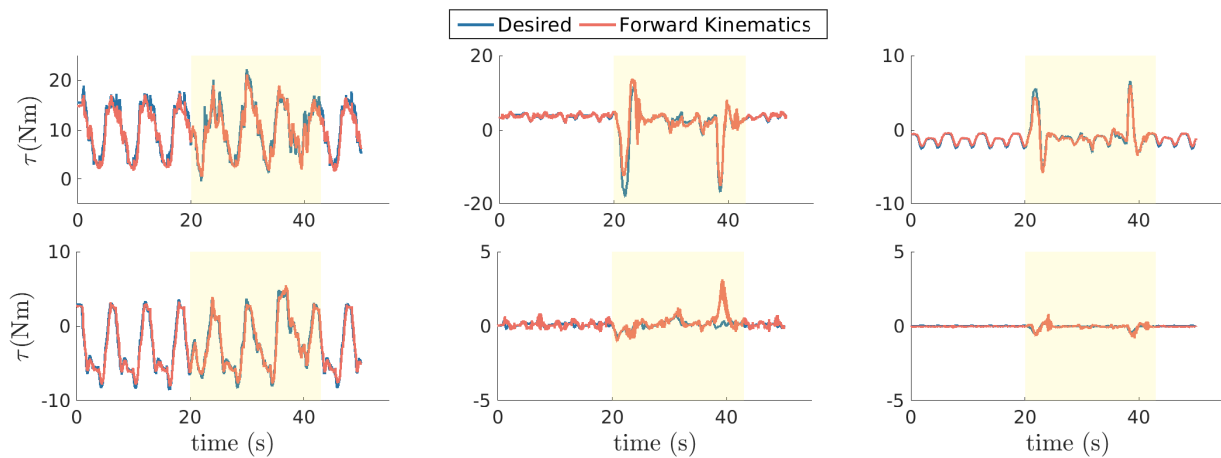


Fig. 6: Tracking of the joint torques while performing a cartesian task on the right leg of the ergoCub robot while subject to external contacts.

that in the case of RNEA, the controller becomes unstable due to significant torque tracking errors at the joints. Conversely, with the UKF-based approach, torques are accurately tracked. Indeed, the RMSE for the three joints are 2.03 Nm, 0.8 Nm, and 0.6 Nm. The multimedia material attached to this paper shows the torso behavior with both algorithms.

#### D. Tracking of a cartesian task

In this last validation test, the controller aims to track a specific link pose. The cartesian trajectory is run on a single leg of the robot ergoCub. External forces are applied to perturb the leg to assess the stability of our torque estimation and control. Figure 5 shows the desired position and orientation tracking of the controlled frame. The highlighted yellow intervals correspond to time-slots when external disturbances are exerted. We can observe a delay in orientation tracking which can be attributed to static friction in the harmonic drive units which is not modeled [24]. As the torque tracking error grows, the torque controller progressively requests higher motor currents overcoming the static friction effect. Figure 6 illustrates the desired torque tracking. The RMSE of the torque tracking for the leg joints are 1.31 Nm, 1.41 Nm, 0.31

Nm, 0.64 Nm, 0.38 Nm, and 0.08 Nm.

#### V. CONCLUSIONS

This paper contributes to the design of a joint torque estimator for humanoid robots equipped with harmonic drives and without joint torque sensors. We develop a UKF-based sensor fusion algorithm, robust against external disturbances, sensor measurement noises, and modeling errors. We validate our estimation approach on the position-controlled humanoid robot ergoCub, by comparing the proposed algorithm with the state-of-the-art deterministic approach based on RNEA. The validation results show that the novel method ensures effective tracking of the desired torques and high-level tasks, even in the presence of external disturbances.

As a future development, we aim to enhance friction model identification by incorporating more advanced models [24], as well as integrating tactile sensors to improve the estimation of external contact points [15]. We also plan to compare our approach with other methods that utilize motor currents and joint and motor encoder measurements for the estimation [17]. Furthermore, we plan to evaluate the performances of our approach in handling more complex whole-body tasks, such as balancing on the ground or walking.

## REFERENCES

- [1] A. Kaplish and K. Yamane, "Motion retargeting and control for teleoperated physical human-robot interaction," in *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*, 2019, pp. 723–730.
- [2] G. Romualdi, S. Daffarra, Y. Hu, P. Ramadoss, F. J. A. Chavez, S. Traversaro, and D. Pucci, "A benchmarking of dcm-based architectures for position, velocity and torque-controlled humanoid robots," *International Journal of Humanoid Robotics*, vol. 17, no. 01, p. 1950034, 2020.
- [3] S. Daffarra, G. Romualdi, and D. Pucci, "Dynamic complementarity conditions and whole-body trajectory optimization for humanoid robot locomotion," *IEEE Transactions on Robotics*, vol. 38, no. 6, pp. 3414–3433, 2022.
- [4] G. Mesesan, J. Engelsberger, G. Garofalo, C. Ott, and A. Albu-Schäffer, "Dynamic walking on compliant and uneven terrain using dcm and passivity-based whole-body control," in *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2019, pp. 25–32.
- [5] N. Ramuzat, G. Buondonno, S. Boria, and O. Stasse, "Comparison of position and torque whole body control schemes on the humanoid robot talos," in *20th International Conference on Advanced Robotics (ICAR)*, 2021.
- [6] A. De Luca and R. Mattone, "Sensorless robot collision detection and hybrid force/motion control," in *Proceedings of the 2005 IEEE international conference on robotics and automation*. IEEE, 2005, pp. 999–1004.
- [7] B. Henze, M. A. Roa, and C. Ott, "Passivity-based whole-body balancing for torque-controlled humanoid robots in multi-contact scenarios," *The International Journal of Robotics Research*, vol. 35, no. 12, pp. 1522–1543, 2016.
- [8] P. Ferrari, L. Rossini, F. Ruscelli, A. Laurenzi, G. Oriolo, N. G. Tsagarakis, and E. M. Hoffman, "Multi-contact planning and control for humanoid robots: Design and validation of a complete framework," *Robotics and Autonomous Systems*, vol. 166, p. 104448, 2023.
- [9] G. Romualdi, N. A. Villa, S. Daffarra, D. Pucci, and O. Stasse, "Whole-body control and estimation of humanoid robots with link flexibility," in *2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids)*. IEEE, 2022, pp. 104–111.
- [10] J.-K. Min, K.-H. Ahn, H.-C. Park, and J.-B. Song, "A novel reactive-type joint torque sensor with high torsional stiffness for robot applications," *Mechatronics*, vol. 63, p. 102265, 2019.
- [11] G. Romualdi, S. Daffarra, Y. Hu, and D. Pucci, "A benchmarking of dcm based architectures for position and velocity controlled walking of humanoid robots," in *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2018, pp. 1–9.
- [12] G. Romualdi, S. Daffarra, G. L'Erario, I. Sorrentino, S. Traversaro, and D. Pucci, "Online non-linear centroidal mpc for humanoid robot locomotion with step adjustment," *arXiv e-prints*, pp. arXiv–2203, 2022.
- [13] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka, "The development of honda humanoid robot," in *Proceedings. 1998 IEEE international conference on robotics and automation (Cat. No. 98CH36146)*, vol. 2. IEEE, 1998, pp. 1321–1326.
- [14] A. Del Prete, N. Mansard, O. E. Ramos, O. Stasse, and F. Nori, "Implementing torque control with high-ratio gear boxes and without joint-torque sensors," *International Journal of Humanoid Robotics*, vol. 13, no. 01, p. 1550044, 2016.
- [15] M. Fumagalli, S. Ivaldi, M. Randazzo, L. Natale, G. Metta, G. Sandini, and F. Nori, "Force feedback exploiting tactile and proximal force/torque sensing," *Autonomous Robots*, vol. 33, no. 4, pp. 381–398, 2012.
- [16] S. Traversaro, A. Del Prete, S. Ivaldi, and F. Nori, "Inertial parameters identification and joint torques estimation with proximal force/torque sensing," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 2105–2110.
- [17] Y. Nagamatsu, T. Shirai, H. Suzuki, Y. Kakiuchi, K. Okada, and M. Inaba, "Distributed torque estimation toward low-latency variable stiffness control for gear-driven torque sensorless humanoid," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 5239–5244.
- [18] E. A. Wan and R. Van Der Merwe, "The unscented kalman filter," *Kalman filtering and neural networks*, pp. 221–280, 2001.
- [19] J. K. Hackett and M. Shah, "Multi-sensor fusion: a perspective," in *Proceedings., IEEE International Conference on Robotics and Automation*. IEEE, 1990, pp. 1324–1330.
- [20] R. Featherstone, *Rigid body dynamics algorithms*. Springer, 2014.
- [21] P. S. Gandhi, F. H. Ghorbel, and J. Dabney, "Modeling, identification, and compensation of friction in harmonic drives," in *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, vol. 1. IEEE, 2002, pp. 160–166.
- [22] M. A. Ismail, J. Windelberg, and G. Liu, "Simplified sensorless torque estimation method for harmonic drive based electro-mechanical actuator," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 835–840, 2021.
- [23] R. Cisneros, M. Benallegue, R. Kikuuwe, M. Morisawa, and F. Kanehiro, "Reliable chattering-free simulation of friction torque in joints presenting high stiction," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 6318–6325.
- [24] K. Johansson and C. Canudas-De-Wit, "Revisiting the lugre friction model," *IEEE Control systems magazine*, vol. 28, no. 6, pp. 101–114, 2008.
- [25] Z. Shi and G. Liu, "Torque estimation of robot joint with harmonic drive transmission using a redundant adaptive robust extended kalman filter," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 6382–6387.
- [26] "Gazebo — gazebosim.org," <https://gazebosim.org/home>, [Accessed 12-09-2023].
- [27] I. Sorrentino, "paper\_sorrentino\_2024\_icra\_robot-dynamics-estimation," [https://github.com/ami-iit/paper\\_sorrentino\\_2024\\_icra\\_robot-dynamics-estimation](https://github.com/ami-iit/paper_sorrentino_2024_icra_robot-dynamics-estimation).
- [28] B. Siciliano, L. Sciacivco, L. Villani, and G. Oriolo, *Robotics: modelling, planning and control*. Springer Science & Business Media, 2010.