

MOB-Net: Limb-modularized Uncertainty Torque Learning of Humanoids for Sensorless External Torque Estimation

Journal Title
XX(X):1–21
©The Author(s) 2023
Reprints and permission:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/ToBeAssigned
www.sagepub.com/



Daegyu Lim¹, Myeong-Ju Kim¹, Junhyeok Cha¹ and Jaeheung Park^{1,2,3}

Abstract

Momentum observer (MOB) can estimate external joint torque without requiring additional sensors, such as force/torque or joint torque sensors. However, the estimation performance of MOB deteriorates due to the model uncertainty which encompasses the modeling errors and the joint friction. Moreover, the estimation error is significant when MOB is applied to the high-dimensional floating-base humanoids, which prevents the estimated external joint torque from being used for force control or collision detection in the real humanoid robot. In this paper, the pure external joint torque estimation method named *MOB-Net* is proposed for humanoids. MOB-Net learns the model uncertainty torque and calibrates the estimated signal of MOB. The external joint torque can be estimated in the generalized coordinate including whole-body joints and the virtual joints of the floating-base robot only with internal sensors (an IMU on the pelvis and encoders in the joints). Our method substantially reduces the estimation errors of MOB, and the robust performance of MOB-Net for the unseen data is validated through extensive simulations, real robot experiments, and ablation studies. Finally, various collision handling scenarios are presented using the estimated external joint torque from MOB-Net: contact wrench feedback control for locomotion, collision detection, and collision reaction for safety.

Keywords

Humanoid robot, model uncertainty learning, external joint torque estimation, momentum observer, collision detection, collision reaction, bipedal locomotion, neural network

1 Introduction

Safety is the most essential requirement for the collaborative robot that shares the workspace with humans. To avoid any undesirable contact with obstacles or humans, collision avoidance algorithms have been developed using vision sensors. However, it is difficult to prevent all the possible collisions using the vision sensors because blind spots can exist where the vision sensor can not detect due to occlusions as shown in Figure 1. To detect and cope with unexpected collisions, additional Force/Torque Sensors (FTS) or Joint Torque Sensors (JTS) can be equipped on the end-effectors or every joint of the robot, respectively, or artificial tactile/skin sensors can be attached to the surface of the robot as in Park et al. (2007); Cirillo et al. (2015); Kobayashi et al. (2022). However, incorporating these additional sensors raises the overall cost of the robot system, making it unaffordable. Moreover, attaching these sensors increases the inertia of the robot, system complexity, and probability of system failure, which limits the allowable number of sensors.

To cope with unexpected collisions with a minimum cost increase, external torque estimation methods and collision handling methods using the generalized momentum observer only with proprioceptive sensors have also been actively studied in recent decades De Luca and Mattone (2003, 2005); De Luca et al. (2006, 2007); Haddadin et al. (2008, 2010, 2017). As a result of those studies, the collaborative manipulator can estimate the external joint torque and can reliably detect and react to unexpected collisions without exteroceptive sensors.

In the case of bipedal robots, such as humanoid robots, maintaining the robot's balance is directly related to safety because the robot's base is not fixed, and a fall of the robot could cause severe injury to nearby humans. With this awareness of the problem, many studies have been conducted on the walking controller to enhance the stability and robustness of the humanoid robot in Griffin et al. (2017); Joe and Oh (2018); Jeong et al. (2019); Khadiv et al. (2020); Daneshmand et al. (2021); Mesesan et al. (2021); Kim et al. (2022, 2023). In these studies, the balancing control performances against external pushes are demonstrated in the open space. However, such external pushes by an experimenter or released weight can not represent all collisions in the real-world environment. For example, as shown in Figure 1, collision sources in the real-world would not disappear causing repetitive large impacts to the robot (e.g., heavy box, desk, stair, wall, etc.). Therefore, only with passive balancing controls, the humanoid robot could eventually lose balance in these collisions with heavy

¹Department of Intelligence and Information, Seoul National University, Seoul, Republic of Korea

²Advanced Institutes of Convergence Technology, Suwon, Republic of Korea

³ASRI, RICS, Seoul National University, Seoul, Republic of Korea

Corresponding author:

Jaeheung Park, Dynamic Robotic Systems Lab, Department of Intelligence and Information, Seoul National University, Gwanak-ro 1, Gwanak-gu, Seoul 08826, Republic of Korea.

Email: park73@snu.ac.kr

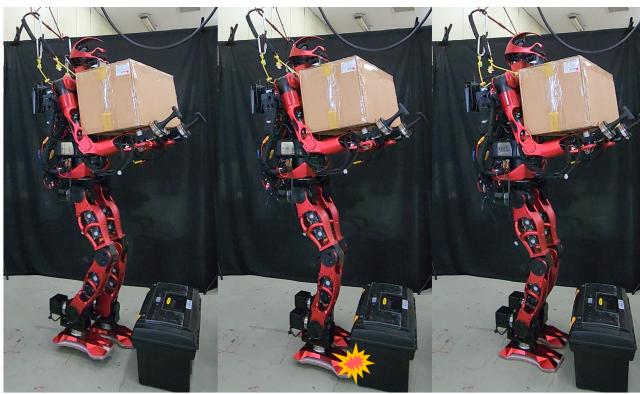


Figure 1. An example of collision detection on the foot of a humanoid robot with a heavy box due to the camera sensor occlusion on the head and collision reaction by back-stepping. See Scenario #3 in Extension 2 for the video.

obstacles. However, if an active collision detection and reaction system is implemented in the humanoid, they would not fall by avoiding further collisions with obstacles.

Nonetheless, studies on collision handling of the legged robot are less spotlighted than the balancing control problem. There were several studies on the collision avoidance of the quadrupedal robot in [Gaertner et al. \(2021\)](#); [Chiu et al. \(2022\)](#) and the collision/contact detection of the quadrupedal robot in [Bledt et al. \(2018\)](#); [Lin et al. \(2021\)](#); [Narukawa et al. \(2017\)](#); [Van Dam et al. \(2022\)](#). For the humanoid robot, there were several studies for external torque estimation for contact detection or contact wrench control using model-based algorithms and FTS in [Kaneko et al. \(2012\)](#); [Flacco et al. \(2016\)](#); [Vorndamme et al. \(2017\)](#); [Benallegue et al. \(2018\)](#); [Lee et al. \(2018\)](#); [Ito et al. \(2019\)](#), but those methods require FTS and are validated only in the simulation or only for the quasi-static conditions. Moreover, to the best of our knowledge, research on the active collision detection and reaction strategy of humanoids during walking has not been conducted yet. Indeed, during DARPA Robotics Challenge in 2015, [Krotkov et al. \(2018\)](#), and ANA Avatar Xprize in 2022, [Ackerman \(2023\)](#), several humanoid robots fell while walking due to a collision with an obstacle even though advanced balancing controllers were implemented, vision sensors were equipped, and the robot was operated remotely. The collision handling method of humanoids can be implemented in parallel with the balancing controller and enhance the safety of the robot in real scenarios.

One of the major reasons for the absence of the collision handling approach (from collision detection to reaction) for humanoids is the low accuracy of the external joint torque estimation. The estimated external joint torque is commonly used as a collision signal for the first step of the collision handling process, i.e., collision detection. Unlike the fixed-base manipulator, it is challenging to estimate the external joint torque of humanoids. The floating-base humanoid suffers from modeling errors in the complex system of high degrees of freedom (DoF), large friction torques due to heavy load on the leg, and the sensing noise that occurs from the repetitive collision on the foot during walking. Such model uncertainty causes large errors in the external joint torque

estimation. Due to its significant errors, sensitive collision detection and reaction are difficult for humanoid robots.

In this paper, therefore, we propose an uncertainty torque learning method for humanoids to estimate the pure external torque of the whole body without additional sensors. First, the model uncertainty torque is learned using the proposed limb-modularized neural network architecture, *MOB-Net*, which combines a model-based momentum observer (MOB) and a data-driven deep learning method. Then, the estimated external torque from MOB is calibrated by the learned uncertainty torque for accurate external torque estimation. The proposed method is validated through extensive simulations, real robot experiments, and ablation studies. Additionally, various collision handling scenarios using the estimated external joint torque are presented.

The major contributions of this work can be summarized as follows:

1. Limb-modularized recurrent neural network, MOB-Net, is proposed for learning the whole-body uncertainty torque of the humanoid, and the external joint torque is accurately estimated in real-time thanks to the effective and efficient limb-modularized structure of MOB-Net. The external joint torque is estimated only with proprioceptive sensors (joint encoders and an Inertia Measurement Unit (IMU)) for a humanoid while FTS on foot is required for data collection, and the estimation performance is significantly improved compared to the model-based methods contributing to the realization of low-cost humanoids.
2. Accurate and general estimation performance of the MOB-Net to the unseen data is validated through simulations and real robot experiments compared to the end-to-end learning method (our previous work in [Lim et al. \(2023\)](#)) and other model-based methods. Additionally, more sensitive and consistent collision detection performance using both the estimated external joint torque and the estimated standard deviation of the uncertainty torque is presented contributing to the safety of humanoids.
3. To the best of our knowledge, we demonstrate the first active collision detection and reaction scenarios for the humanoid robot during walking.

2 Related Work

2.1 Model Uncertainty Handling in External Torque Estimation

[De Luca and Mattone \(2003\)](#) introduced a concept of generalized momentum and proposed residual momentum observer for actuator failure detection. Extending from this method, collision detection to reaction methods are developed for safe human-robot interaction in [De Luca et al. \(2006\)](#); [Haddadin et al. \(2008\)](#); [De Luca and Flacco \(2012\)](#). In [Haddadin et al. \(2017\)](#), a collision handling pipeline is introduced as a sequence process: collision detection, isolation, identification, classification, and reaction. Collision detection, the first step of the collision handling pipeline, should be performed sensitively to cope with unexpected collision, but a model-based

method fundamentally suffers from model uncertainty which deteriorates not only the collision detection performance but also the other collision handling methods following the collision detection.

To enhance the collision detection performance with the existence of the uncertainty torque, a band-pass filter is used in Cho and Song (2013). A band-pass filter suppresses both the high-frequency errors from the sensor noise and the low-frequency errors from the modeling error resulting in more sensitive collision detection. Caldas et al. (2013); Guo et al. (2018) introduced dynamics threshold methods for sensitive collision detection while preventing false positives due to modeling errors. Although such methods improve collision detection sensitivity, they do not fundamentally improve the estimation performance of external joint torque, and these methods are limited to collision detection.

Lee et al. (2015) suggested a friction model that calibrates the residual signal from the momentum observer for more accurate external joint torque estimation, and they show more sensitive collision detection performance using the friction model. Birjandi and Haddadin (2020) proposed an online model adaptation method based on the regressor. The regressor-based method calibrates the inertia parameter adaptively and improves the collision detection performance, but their method requires joint acceleration and joint torque measurement. Lee and Park (2018) also proposed an inertia parameter identification method formulating the problem in the Riemannian manifold. They validate the accuracy and the robustness of the proposed method for a multi-body system, humanoid. However, these methods only cover the joint friction or errors of the inertial parameter, a part of the model uncertainty. Although these methods tried to solve the model uncertainty problem, it is difficult to design an appropriate model to represent the nonlinear friction or an effective algorithm to correct the modeling errors.

On the other side, many data-driven collision detection methods also have been developed to circumvent the model uncertainty problem and directly infer the collision. Using the deep learning technique, end-to-end collision detection methods showed superior detection performance compared to the model-based method in Heo et al. (2019); Park et al. (2020, 2021); Kim et al. (2021). However, the end-to-end collision detection method can not be applied to other tasks such as external force estimation or force control.

Meanwhile, deep learning is also used for an external wrench or external joint torque estimation which can be applied to various problems: collision detection to reaction, contact force control, and human-robot interaction. El Dine et al. (2018) proposed RNNOB which infers the pattern of the FTS during contact-free motion and calibrates the sensor measurement. Although the force during free motion is estimated using only joint encoders and IMU, this method still requires FTS for external wrench estimation after the training. Tran et al. (2020) also trained a Multilayer perceptron (MLP) network to directly estimate the external linear force of the surgical robot, da Vinci Research Kit, with joint torques and velocities. A similar approach is presented for the industrial manipulator in Shan and Pham (2023). They performed the peg-in-hole task to validate the contact wrench estimation performance. Yilmaz et al. (2020) proposed the residual dynamics torque learning during the

free motion for external joint torque estimation on the da Vinci Research Kit, and the estimated external torque is transformed to the external wrench by using Jacobian inverse mapping. We also proposed uncertainty torque learning based on the momentum observer for external torque estimation without additional sensors in Lim et al. (2021). The estimated external torque is used for collision detection in the 2 DoF planar manipulator. However, research on the external torque learning for the legged robot has not been performed except for our other previous work in Lim et al. (2023).

2.2 Collision Handling for Legged Robot

Momentum observer introduced in De Luca et al. (2006) is also applied to the humanoid robots in various works. Flacco et al. (2016) presented a momentum observer framework for floating-base humanoids based on the work in De Luca et al. (2006) and validated the external torque estimation and contact point estimation performance in the simulation. Vorndamme et al. (2017) utilized the momentum observer for multi-contact collision detection, isolation, and identification of the humanoid robot, and the proposed method is validated and analyzed in the simulation. In Benallegue et al. (2018), the extended Kalman filter-based external wrench estimation algorithm is suggested. The proposed method performs better than their previous sensor-based method in Kaneko et al. (2012) for the hand contact wrench estimation through simulation and the real robot experiment. However, all the aforementioned methods require FTS and the estimation task is only validated in the simulation or in the real robot but during quasi-static motion.

For the quadrupedal robot, Hwangbo et al. (2016) proposed a probabilistic foot contact estimation method combined with the dynamics and kinematics of the robot, and they demonstrated faster contact detection performance compared with the momentum observer-based method. Benallegue et al. (2018) proposed a discretized version of the momentum observer for more accurate estimation performance, and it is used for foot contact estimation. Van Dam et al. (2022) also utilized the momentum observer combined with a band-pass filter for collision detection of the legged manipulator. The band-pass filter relieves the modeling error and signal noise at the same time, but the estimation error increases when the legged robot walks. Barasuol et al. (2019) proposed a kinematic-based collision detection method and reaction control method for the collision on the shin link using the implemented trunk controller, but it does not produce additional reaction motion at the planning level. Moreover, all the model-based method for the legged robot does not account for the model uncertainty.

Foot contact detection methods based on deep learning are investigated for the humanoid robot in Piperakis et al. (2022) and the quadrupedal robot in Lin et al. (2021). In Piperakis et al. (2022), the contact states of each foot are estimated directly using both FTS and IMU on the foot. Lin et al. (2021) also proposed a deep learning classification method for the contact state estimation of a quadrupedal robot, and the estimated contact states are used for the SLAM algorithm. But, as mentioned in the previous section, the

application of the end-to-end contact estimation is limited to contact detection.

In summary, there were many works for collision/contact detection and identification problems using model-based methods or data-driven methods. However, none of the studies cover the collision reaction problem against unexpected collisions on the leg for the legged robot system. Additionally, model uncertainty is indirectly handled by threshold, signal filtering, and probabilistic model, or ignored. In our approach, the external torque is estimated accurately in the generalized coordinate considering the model uncertainty, and the entire collision handling scenarios of the humanoid robot are presented from collision detection to reaction without FTS.

2.3 Comparison to our previous work

This paper is expanded from our previous work in Lim et al. (2021) as follows. 1) We expand the external torque estimation method validated in the 2 DoF fixed-based test bed to the whole body of the floating-base humanoid robot with 37 DoF except head and hand joints. 2) The architecture of the neural network is advanced to train and infer the network efficiently for humanoids. Specifically, a limb-modularized network structure is proposed in this paper using different networks, using different input/output features, and introducing a different data collection method compared to the method in Lim et al. (2021). The general performance of the proposed method for the unseen data is extensively validated through simulations, experiments, and ablation studies. 3) We also applied the estimated pure external torque of the humanoid robot to the collision detection task and various collision reaction scenarios. Previously in Lim et al. (2021), only the collision detection was performed using the estimated external joint torque in the 2 DoF test bed.

3 Problem Formulation

3.1 Rigid Body Dynamics of Floating-base Robot

A humanoid can be described as a floating-base multi-body system that has $n + 1$ rigid bodies and n joints. In the case of a floating-base robot, the robot's dynamics can be expressed by linking six virtual joints to the base frame. Thus, the fundamental dynamics of the rigid body for an $n + 6$ DoF system with six virtual joints are outlined as follows

$$\mathbf{M}(\mathbf{q}_v)\ddot{\mathbf{q}}_v + \mathbf{C}(\mathbf{q}_v, \dot{\mathbf{q}}_v)\dot{\mathbf{q}}_v + \mathbf{g}(\mathbf{q}_v) = \boldsymbol{\tau}_v + \boldsymbol{\tau}_e, \quad (1)$$

$$\boldsymbol{\tau}_e = \sum_{i=1}^k \mathbf{J}_{c,i}^T \mathbf{F}_{e,i}, \quad (2)$$

where $\mathbf{M}(\mathbf{q}_v), \mathbf{C}(\mathbf{q}_v, \dot{\mathbf{q}}_v) \in \mathbb{R}^{(n+6) \times (n+6)}$, and $\mathbf{g}(\mathbf{q}_v) \in \mathbb{R}^{n+6}$ are the inertia matrix, the Coriolis and centrifugal matrix, and the gravity vector, respectively. $\mathbf{q}_v, \dot{\mathbf{q}}_v, \ddot{\mathbf{q}}_v \in \mathbb{R}^{n+6}$ are the generalized position, velocity, and acceleration vectors including virtual joints, respectively. $\boldsymbol{\tau}_v \in \mathbb{R}^{n+6}$ is the control torque and $\boldsymbol{\tau}_e \in \mathbb{R}^{n+6}$ is the external torque. The external torque occurs by the external wrench in the operational space as represented in (2) where k , $\mathbf{J}_{c,i} \in \mathbb{R}^{6 \times (n+6)}$, and $\mathbf{F}_{e,i} \in \mathbb{R}^6$ are the total number of contacts,

the i -th contact Jacobian matrix, and the i -th external wrench, respectively. The generalized coordinate vector consists of six virtual joints, and n motor angles: $\mathbf{q}_v = [\mathbf{x}_{fb}^T \ \boldsymbol{\theta}_{fb}^T \ \mathbf{q}^T]^T$, $\dot{\mathbf{q}}_v = [\mathbf{v}_{fb}^T \ \boldsymbol{\omega}_{fb}^T \ \dot{\mathbf{q}}^T]^T$, $\ddot{\mathbf{q}}_v = [\ddot{\mathbf{v}}_{fb}^T \ \ddot{\boldsymbol{\omega}}_{fb}^T \ \ddot{\mathbf{q}}^T]^T$. $\mathbf{x}_{fb} \in \mathbb{R}^3$ and $\boldsymbol{\theta}_{fb} \in \mathbb{R}^3$ are the position vector and the Euler angles of the floating-base. The orientation of the floating-base can be also expressed with the orientation matrix $\mathbf{R}_{fb}^T \in SO(3)$. $\mathbf{v}_{fb} \in \mathbb{R}^3$ and $\boldsymbol{\omega}_{fb} \in \mathbb{R}^3$ are the linear and angular velocity of the floating-base. The underactuated floating-base robot has zero input torque for the virtual joints, and motor torque $\boldsymbol{\tau}_m \in \mathbb{R}^n$ for the actuating joints as $\boldsymbol{\tau}_v = [\mathbf{0}^T \ \boldsymbol{\tau}_m^T]^T$.

3.2 Momentum Observer (MOB)

The external torque on the robot can be obtained by mapping the contact force $F_{e,c}$ measured by FTS to the joint torque as described in (2), or it can be estimated by measuring the link-side joint torque from JTS and using the dynamics of the robot in (1). However, for the robot in the absence of such sensors (FTS or JTS), the external torque can be estimated from only internal sensors including the joint encoder, motor current sensor, and IMU.

Using the robot's dynamics in (1) and the nominal model, external torque can be calculated by rearranging the equation as below

$$\boldsymbol{\tau}_e = \overline{\mathbf{M}}(\mathbf{q}_v)\ddot{\mathbf{q}}_v + \overline{\mathbf{C}}(\mathbf{q}_v, \dot{\mathbf{q}}_v)\dot{\mathbf{q}}_v + \overline{\mathbf{g}}(\mathbf{q}_v) - \boldsymbol{\tau}_v \quad (3)$$

where $\overline{(\cdot)}$ indicates the nominal model obtained from the estimated model parameters. Assuming ideal system identification, $\boldsymbol{\tau}_e$ can be obtained only with internal sensor information $\mathbf{q}_v, \dot{\mathbf{q}}_v, \ddot{\mathbf{q}}_v$ and $\boldsymbol{\tau}_m$.

However, the joint acceleration $\ddot{\mathbf{q}}$ in (3) is highly noisy because it is commonly calculated by the numerical differentiation. To circumvent this problem, a conventional model-based method, momentum observer was introduced in De Luca and Mattone (2003) which does not use joint acceleration for external torque estimation. According to De Luca and Mattone (2003), the generalized momentum of the floating-base robot is defined as

$$\mathbf{p} = \overline{\mathbf{M}}(\mathbf{q}_v)\dot{\mathbf{q}}_v. \quad (4)$$

The derivative of the \mathbf{p} can be expressed by using the floating-base dynamics (1) with a nominal model and well known relation $\dot{\overline{\mathbf{M}}} = \overline{\mathbf{C}} + \overline{\mathbf{C}}^T$ as

$$\dot{\mathbf{p}} = \boldsymbol{\tau}_v + \boldsymbol{\tau}_e + \overline{\mathbf{C}}^T(\mathbf{q}_v, \dot{\mathbf{q}}_v)\dot{\mathbf{q}}_v - \overline{\mathbf{g}}(\mathbf{q}_v) \quad (5)$$

$$= \boldsymbol{\tau}_v + \boldsymbol{\tau}_e + \overline{\beta}(\mathbf{q}_v, \dot{\mathbf{q}}_v), \quad (6)$$

where $\overline{\beta}(\mathbf{q}_v, \dot{\mathbf{q}}_v) = \overline{\mathbf{C}}^T(\mathbf{q}_v, \dot{\mathbf{q}}_v)\dot{\mathbf{q}}_v - \overline{\mathbf{g}}(\mathbf{q}_v)$ is defined for the readability. To estimate external torque, a residual vector $\mathbf{r} \in \mathbb{R}^{n+6}$ and its dynamics are defined as follows:

$$\dot{\mathbf{r}} = \mathbf{K}_0(\dot{\mathbf{p}} - \dot{\hat{\mathbf{p}}}) \quad (7)$$

$$\dot{\hat{\mathbf{p}}} = \boldsymbol{\tau}_v + \overline{\beta}(\mathbf{q}_v, \dot{\mathbf{q}}_v) + \mathbf{r}, \quad (8)$$

where $\mathbf{K}_0 = diag\{k_{0,i}\}$ is the positive diagonal matrix and $\dot{\hat{\mathbf{p}}}$ is the derivative of the estimated momentum. Integrating (7) results in

$$\mathbf{r} = \mathbf{K}_0 \left\{ \mathbf{p}(t) - \mathbf{p}(0) - \int_0^t (\boldsymbol{\tau}_v + \overline{\beta}(\mathbf{q}_v, \dot{\mathbf{q}}_v) + \mathbf{r}) dt \right\}. \quad (9)$$

Substituting (6) and (8) to (7) derives the relation between τ_e and r as

$$\dot{r} = \mathbf{K}_0(\tau_e - r). \quad (10)$$

Once \mathbf{K}_0 is a diagonal matrix, the residual dynamics (10) in each joint is independent and Laplace transforming of it for each joint leads to

$$\frac{r_i(s)}{\tau_{e,i}(s)} = \frac{k_{0,i}}{s + k_{0,i}} \quad (i = 1, 2, \dots, n+6). \quad (11)$$

This means that the MOB residual vector, r , is the first-order low-pass filtered signal of the pure external torque, $\tilde{\tau}_e$.

3.3 Uncertainty Torque

Despite the theoretical guarantee of exponential convergence in (10), the residual torque is prone to have large errors in practice. The estimation error of the momentum observer results from the unmodeled dynamics torques which are not considered in the nominal dynamics or sensor noise. We formally define the error of the nominal model in the torque level as an uncertainty torque.

Definition 3.1. (*Uncertainty Torque*):

Given a nominal model and its dynamics as (3), the difference between the dynamics torque calculated using the nominal model ($\bar{\mathbf{M}}(\mathbf{q}_v)\ddot{\mathbf{q}}_v + \bar{\mathbf{C}}(\mathbf{q}_v, \dot{\mathbf{q}}_v)\dot{\mathbf{q}}_v + \bar{\mathbf{g}}(\mathbf{q}_v)$) and the sum of the control torque and the external torque ($\tau_v + \tau_e$) is uncertainty torque (τ_u).

$$\tau_u = \bar{\mathbf{M}}(\mathbf{q}_v)\ddot{\mathbf{q}}_v + \bar{\mathbf{C}}(\mathbf{q}_v, \dot{\mathbf{q}}_v)\dot{\mathbf{q}}_v + \bar{\mathbf{g}}(\mathbf{q}_v) - \tau_v - \tau_e. \quad (12)$$

Then, the major sources of the uncertainty torque are categorized into three factors in this paper; modeling error, friction torque, and sensor noise.

In practice, it is difficult to obtain the accurate dynamics model parameters of the robot $\mathbf{M}(\mathbf{q}_v)$, $\mathbf{C}(\mathbf{q}_v, \dot{\mathbf{q}}_v)$, and $\mathbf{g}(\mathbf{q}_v)$ in (1). Instead, the nominal model is commonly obtained from a CAD model or estimated by a system identification procedure. Then, the dynamics of the floating-base robot with nominal model parameters is expressed below

$$\bar{\mathbf{M}}(\mathbf{q}_v)\ddot{\mathbf{q}}_v + \bar{\mathbf{C}}(\mathbf{q}_v, \dot{\mathbf{q}}_v)\dot{\mathbf{q}}_v + \bar{\mathbf{g}}(\mathbf{q}_v) = \tau_v + \tau_e + \tau_p. \quad (13)$$

In (13), $\tau_p = (\bar{\mathbf{M}}(\mathbf{q}_v)\ddot{\mathbf{q}}_v + \bar{\mathbf{C}}(\mathbf{q}_v, \dot{\mathbf{q}}_v)\dot{\mathbf{q}}_v + \bar{\mathbf{g}}(\mathbf{q}_v)) - (\mathbf{M}(\mathbf{q}_v)\ddot{\mathbf{q}}_v + \mathbf{C}(\mathbf{q}_v, \dot{\mathbf{q}}_v)\dot{\mathbf{q}}_v + \mathbf{g}(\mathbf{q}_v)) \in \mathbb{R}^{n+6}$ is the modeling error torque induced by the error of the parameters between the nominal model and the real robot. Therefore, because the nominal model is used for the momentum observer in (9), the residual torque contains modeling error torque as well as the external torque.

Various kinds of joint friction models have been introduced to compensate for the friction including coulomb, static, viscous, and load-dependent frictions as introduced in Xiao et al. (2018); Lee et al. (2015); Abeykoon and Chinthaka (2014); Ma et al. (2018); Quiroga et al. (2021). Moreover, it is reported that the friction torque shows hysteresis in Kircanski and Goldenberg (1997); Ruderman and Iwasaki (2015). In this study, we consider the joint friction torque as a function of joint position, joint velocity, and joint torque, $\tau_f(\mathbf{q}_v, \dot{\mathbf{q}}_v, \tau_v)$.

While all kinds of sensors have noise in their measurement, such noise can be modeled and interpreted using a

probabilistic model and one of the most common models for sensor noise is the Gaussian model. Therefore, a lumped disturbance torque induced by the sensor noise, τ_n , can be added to the disturbance. $\tau_n(\mathbf{q}_v, \dot{\mathbf{q}}_v, \ddot{\mathbf{q}}_v) \sim P(\tau_n | \mathbf{q}_v^m, \dot{\mathbf{q}}_v^m, \ddot{\mathbf{q}}_v^m, \sigma_s^2)$ follows the conditional probability distribution and σ_s^2 is the variance of all the sensor noise.

By summing up the three components of the uncertainty torque, the uncertainty torque in (12) can be expressed as a stochastic function of joint position, velocity, acceleration, and joint torque:

$$\begin{aligned} \tau_u(\mathbf{q}_v, \dot{\mathbf{q}}_v, \ddot{\mathbf{q}}_v, \tau_v) = & \tau_p(\mathbf{q}_v, \dot{\mathbf{q}}_v, \ddot{\mathbf{q}}_v) + \tau_f(\mathbf{q}_v, \dot{\mathbf{q}}_v, \tau_v) \\ & + \tau_n(\mathbf{q}_v, \dot{\mathbf{q}}_v, \ddot{\mathbf{q}}_v). \end{aligned} \quad (14)$$

3.4 Problem Definition

In this paper, the fundamental goal is to estimate the pure external joint torques of the floating-base humanoid robot using only internal sensors for the realization of both safe and low-cost humanoids. To achieve this goal, the momentum observer is utilized as a baseline algorithm. However, as mentioned in the previous section, the estimated residual vector of the MOB contains not only the low-pass filtered external torque but also the low-pass filtered uncertainty torque:

$$r = \tilde{\tau}_e + \tilde{\tau}_u \quad (15)$$

Therefore, the core problem is defined as an estimation of the delayed uncertainty torque ($\hat{\tau}_u$). After that, the delayed pure external torque ($\tilde{\tau}_e$) can be obtained by subtracting $\hat{\tau}_u$ from the residual r in (15) as $\tilde{\tau}_e = r - \hat{\tau}_u$. According to the previous section, the uncertainty torque is a stochastic function conditioned by the internal states \mathbf{X}_p . An assumption is made to simplify the complex problem that the conditional probability of the uncertainty torque follows the Gaussian distribution independently for each joint as $\tau_{u,j}(\mathbf{X}_p) \sim \mathcal{N}(\hat{\tau}_{u,j}, \hat{\sigma}_{u,j}^2 | \mathbf{X}_p)$. Finally, the core problem can be formulated as obtaining the optimal normal distribution of the uncertainty torque from proprioceptive sensing.

Problem 3.1. *Model Uncertainty Torque Estimation:*

$$\hat{\tau}_u^*(\mathbf{X}_p), \hat{\sigma}_{u*}^2(\mathbf{X}_p) = \arg \max_{\hat{\tau}_u, \hat{\sigma}_u^2} P(\tau_u | \hat{\tau}_u, \hat{\sigma}_u, \mathbf{X}_p). \quad (16)$$

Find the optimal mean and variance functions of the normal distribution maximizing the conditional probability of the true uncertainty torque given the observation \mathbf{X}_p . \mathbf{X}_p is a set of internal states of the robot obtained from the proprioceptive sensors.

4 Model Uncertainty Torque Learning for Pure External Torque Estimation

In this section, a data-driven method called *MOB-Net* is presented which is the deep learning method for model uncertainty torque estimation to solve **Problem 3.1** and, finally, for pure external torque estimation. Deep learning is powerful for nonlinear regression problems if appropriate input and output features and training data can be obtained. Based on the momentum observer in Section 3.2, dynamics of the humanoid robot, and analysis of the uncertainty torque in Section 3.3, neural networks are designed to estimate the

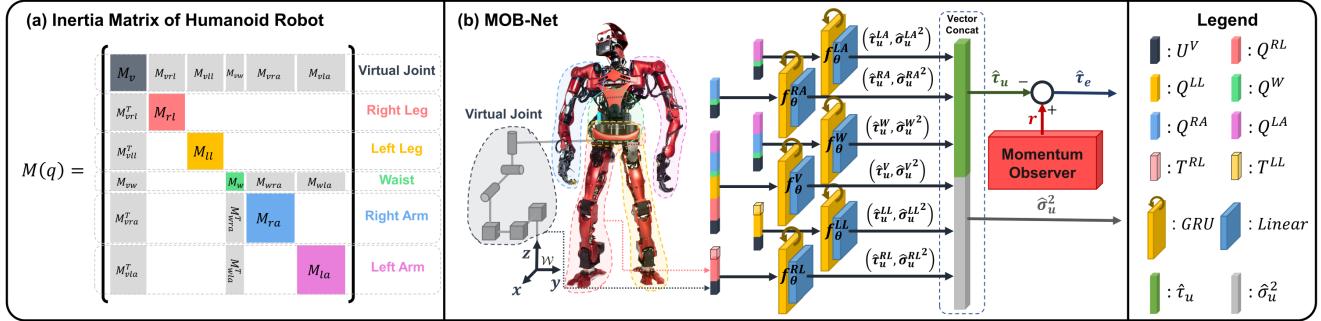


Figure 2. Diagram of the proposed method. (a) Structure of the inertia matrix of a humanoid robot. The empty region has zero elements. (b) The architecture of the proposed MOB-Net. Each color bar represents the input data from the specific limb.

uncertainty torque only with internal information. Details on the network architecture, data collection, and training method are introduced.

4.1 MOB-Net Architecture

In order to design the architecture of MOB-Net, the dynamics of the humanoid robot is analyzed. According to [Featherstone \(2014\)](#), the tree structure of the humanoid robot causes sparsity in the robot's dynamics. Therefore, when the virtual joints are attached to the pelvis link of the humanoid robot, the inertia matrix of the robot has a sparse structure as shown in Figure 2 (a). In the inertia matrix, the blank white area has zero components and gray blocks represent the coupled dynamics. The inertia matrix is only presented for visualizing the dynamics because the inertia matrix is core for analysis of dynamics and $C(q, \dot{q})$ matrix has the same structure as the inertia matrix. As the inertia matrix represents, the robot's dynamics can be divided into 6 groups: *Virtual Joint (V)*, *Right Leg (RL)*, *Left Leg (LL)*, *Waist (W)*, *Right Arm (RA)*, *Left Arm (LA)*. Note that the neck joints are ignored for simplification in this work. When it comes to the dynamics of the left leg, for example, only the states of the virtual joint and the left leg's joint are involved in the dynamics of the left leg. This relationship is utilized for uncertainty torque learning. Therefore, only relevant states are provided to the network for training the uncertainty torque of the specific limb, which enhances the learning performance. This dynamics analysis is leveraged as an inductive bias for an efficient and effective learning framework resulting in six independent networks.

As shown in Figure 2 (b), MOB-Net consists of a momentum observer and six Gated Recurrent Unit (GRU)-Linear networks ($f_\theta^g(\mathbf{X}_p)$, $g \in \{V, RL, LL, W, RA, LA\}$) that estimate uncertainty torque for each limb group. Each GRU-Linear network receives a relevant input vector from the robot's proprioceptive sensors for each limb's uncertainty torque estimation and infers the uncertainty torque and its variance. To ensure that the estimated variance is always positive, the softplus activation function is applied only for the output vectors of the variance. After that, the output vectors are concatenated to a single whole-body uncertainty torque and its variance. Finally, the residual vector of the momentum observer (r) is subtracted from the estimated uncertainty torque ($\hat{\tau}_u$) leaving pure external torque ($\hat{\tau}_e$).

The reason for using the Recurrent Neural Network (RNN) structure with a GRU module is that RNN is

specialized for handling time-sequence data inherently. The friction torque shows hysteresis requiring history information in addition to the current states for uncertainty torque estimation and RNN is suitable. Moreover, RNN enables to avoidance of using joint acceleration which is noisy, and more details about input features are discussed in Section 4.2. The superiority of GRU compared with the other kinds of networks is presented in Section 6.

Therefore, limb-group modularized GRU is adopted as a key architecture of MOB-Net. Thanks to its compact size, not only the high estimation performance but also the real-time calculation of 6 networks on the embedded computer is possible within 1 ms using a single core. A CPU (Intel Core i7-10700K) is used in the real robot.

4.2 Input Feature and Target Value

The selection of the input feature (\mathbf{X}_p) and the target value of the uncertainty torque (τ_u) affect the way of data collection, the architecture of the network, and most importantly the learning performance. Therefore, the input feature and the target value should be determined carefully considering various aspects. We design the input feature and the target value of the whole body of the humanoid robot by categorizing the limb groups of a humanoid robot into legs (RL, LL), upper body (W, RA, LA), and virtual joint (V).

The input vector is defined as a sequence of the selected proprioceptive states as below.

$$\mathbf{X}_p^g = [\mathbf{x}^g(k-h+1), \dots, \mathbf{x}^g(k-1), \mathbf{x}^g(k)],$$

where k is the current discrete control time, and h is the length of the data sequence horizon. $(\cdot)^g$ indicates the partial vector of the joints in the group g . The proprioceptive input states, $\mathbf{x}^g(k)$, are composed of a combination of three sensor vectors of each limb group g :

$$\begin{aligned} \mathbf{Q}^g(k) &= [\mathbf{q}^g(k), \dot{\mathbf{q}}^g(k)] \\ \mathbf{T}^g(k) &= [\boldsymbol{\tau}_d^g(k)] \\ \mathbf{U}^V(k) &= [{}^p\mathbf{R}_{fb,1:6}(k), {}^p\boldsymbol{\omega}_{fb}(k), {}^p\dot{\mathbf{v}}_{fb}(k)] \end{aligned} \quad (17)$$

where $\mathbf{Q}^g(k)$ is the concatenation of joint states, $\mathbf{T}^g(k)$ is the desired torque vector, and $\mathbf{U}^V(k)$ is the vector of IMU measurement. $\boldsymbol{\tau}_d^g(k)$ is the desired joint torque belonging to the limb group g . ${}^p\mathbf{R}_{fb,1:6}(k) \in \mathbb{R}^6$ is the first two columns of the rotation matrix of the pelvis. ${}^p\boldsymbol{\omega}_{fb}(k)$ and ${}^p\dot{\mathbf{v}}_{fb}(k)$ are

Table 1. Summary of the network architecture of MOB-Net.

Network	Input State (x^g)	Target Value (τ_u)	Network Architecture	Network Size	Time Horizon of TBPTT (h)
f_θ^{RL}	(Q^{RL}, T^{RL}, U^V)	$r^{RL} - \tau_e^{RL}$	GRU[30, 150]-Linear[150, 12]	83712	50 steps
f_θ^{LL}	(Q^{LL}, T^{LL}, U^V)	$r^{LL} - \tau_e^{LL}$	GRU[30, 150]-Linear[150, 12]	83712	50 steps
f_θ^{RA}	(Q^{RL}, Q^W, U^V)	r^{RA}	GRU[34, 200]-Linear[200, 16]	144816	50 steps
f_θ^{LA}	(Q^{LA}, Q^W, U^V)	r^{LA}	GRU[34, 200]-Linear[200, 16]	144816	50 steps
f_θ^W	$(Q^{RA}, Q^{LA}, Q^W, U^V)$	r^W	GRU[50, 200]-Linear[200, 6]	152406	50 steps
f_θ^V	$(Q^{RL}, Q^{LL}, Q^{RA}, Q^{LA}, Q^W, U^V)$	$r^V - \tau_e^V$	GRU[74, 200]-Linear[200, 12]	168012	100 steps

the pelvis angular velocity and the pelvis linear acceleration in the pelvis local frame. The specific input state and the target value for each network f_θ^g is summarized in Table 1. The reason for the selection of the input state and the target value of MOB-Net is addressed in the three considerations: the quality of the available information, characteristics of the potential target task, and the cost of the data collection.

4.2.1 Quality of Information Although all the required input features ($q_v, \dot{q}_v, \ddot{q}_v, \tau_m$) of uncertainty torque derived in (14) can be measured or estimated from the internal sensors (encoder, IMU, and motor current sensor), joint acceleration is too noisy to use from the numerical differentiation. Also, measuring the motor torque requires a current sensor and it is a time-delayed signal compared to the command torque. Therefore, not all input features from the dynamics analysis are provided to the network but only the sequence of the partial information is used as input data based on several assumptions, which is also used in our previous work, Lim et al. (2023):

- Joint acceleration is redundant and can be estimated from the time sequence of joint velocity.
- Angular acceleration and linear velocity of the pelvis can be estimated with the time sequence of joint position, joint velocity, and IMU data (orientation of pelvis, pelvis angular velocity, and pelvis linear acceleration), Rotella et al. (2014).
- Motor torque measured from the current sensor τ_m can be replaced with the desired torque τ_d because it has a reasonably small error to the measured torque.

Therefore, only three sensor vectors in (17) are used as input features.

4.2.2 Potential Target Task The expected target tasks for the legs of the humanoid robot are to support the entire weight of the robot with one or two legs and walk to the target destination making repetitive impacts with the ground. During these tasks, the legs are loaded with large forces (> 1000 N for our robot) making the load-dependent dynamics dominant. On the other hand, the upper body, especially the arm's typical target task is manipulation whose maximum expected load is below 5 kg (< 50 N), which makes the load-dependent dynamics minor. Therefore, load-dependent friction is considered for the legs requiring the desired joint torque $T^g(k)$ as an input feature in addition to the $Q^g(K)$ and $U^V(K)$. For the upper body, however, load-dependent friction is ignored and the friction torque becomes a function of joint velocity ($\tau_f(\dot{q})$) only requiring $Q^g(K)$ and $U^V(K)$ as input features. Virtual joints do not consider friction and require only $Q^g(K)$ of all joints and $U^V(K)$ to infer the modeling error torque.

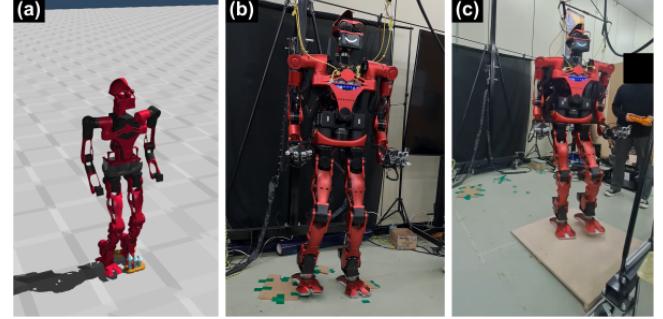


Figure 3. Random walking data collection environment for the simulation and the real robot experiment. (a) Simulation environment with uneven terrain of check pattern. (b) Uneven terrain setting with three wooden plates of roughly 10mm in thickness. (c) Slope terrain setting of roughly 3 degrees.

4.2.3 Cost of Data Collection For the target value of the uncertainty torque, the residual vector is basically used which can be calculated with only internal states. If it can be assumed that any external force is not applied, the residual vector during contact-free motion itself becomes the target uncertainty torque, which costs nothing more than the residual calculation. Thus, contact-free motion is used to get the training data for the upper body. For the legs, on the other hand, walking data is utilized without additional devices for the data collection. If the walking data is used, external force should be measured to obtain the model uncertainty torque. In our robot, the FTS under the feet is used to measure the contact wrench, and the measured contact wrenches are converted to the external torque as (2). Otherwise, the free motion of the legs can be collected by hanging up the robot in the air ignoring the load-dependent friction and possibly elasticity of the joint as the upper body joints. In this case, however, additional experimental devices are needed to hang up the robot in the air, to measure the lifting force for calculation of the external force in the virtual joints, and to measure the linear velocity of the base link for calculation of the residual of MOB. For our humanoid robot with a high gear ratio reducer (100:1), the load-dependent dynamics could not be ignored because of the large estimation error, and such experimental settings for hanging up the humanoid robot cost more than the FTS on the feet. Therefore, the residual minus the measured external torque $r - \tau_e^m$ from the walking data is used as the target value of the uncertainty torque learning for the networks of the legs, and only the residual r from the free motion data is used as the output for the upper body networks. In the virtual joints, all the external forces are observed, thus $r - \tau_e^m$ is used for the target value.

Table 2. Random walking parameters and their range

Walking Parameters	Range
Step Length X-dir	[-0.15, 0.20] m
Step Length Y-dir	[-0.10, 0.10] m
Turning Angle	[-20, 20]°
Step Duration	{0.6, 0.7, 0.8, 0.9} s
Maximum foot height	{4.0, 5.0, 6.0, 7.0} cm

4.3 Data Collection

Two sets of data are collected using our humanoid robot, TOCABI, and the implemented controllers. The first data set (*Random Walking Data*) is collected for the legs and the second set (*Random Motion Data*) is collected for the upper body and the virtual joint, see [Extension 1](#). Across two data sets, the data is collected with a 1000 Hz sampling frequency which is half of the control frequency, 2000 Hz. The data contains input and output features of all the joints including the virtual joints (39 DoF). The collected data is divided into training data and validation data with a 9:1 ratio.

Random Walking Data is collected using the implemented walking controller in our robot proposed in [Kim et al. \(2023\)](#). The operator can control the robot manually using a joystick with various walking patterns. The walking command consists of 2D step lengths and a 1D turning angle. Our robot can walk omnidirectionally given an arbitrary combination of walking commands within the predefined limitations. In addition, the robot's step duration, and the maximum foot height of the swing foot were also changed among several predefined values. The range of the random walking parameters is summarized in Table 2. The data collection was performed in the uneven terrain environment as shown in Figure 3. Figure 3 (a) shows the check-patterned uneven terrain with a depth difference of 15 mm in the simulation. Figure 3 (b) and (c) show the uneven terrain and slope terrain settings for data collection for the real robot. Three wooden plates of 10 mm thickness are placed on the floor in Figure 3 (b), and a wooden slope of 3 degrees is placed next to the uneven terrain settings. The robot walks over uneven terrain to produce largely distributed and uncorrelated data. If the data is only collected from the flat terrain, the uncertainty torque can be correlated with the foot height. Additionally, an experimenter applied external disturbances to the upper body and feet of the robot for largely distributed data. While the humanoid robot is walking, the other joints in the upper body are fixed. The total amount of the collected data is 1 hour resulting in about 3.6M samples.

Random Motion Data is collected using the teleoperation control framework developed in our group ([Lim et al. \(2022\)](#)) to move the upper body of the humanoid robot randomly. A person equips the motion trackers on the body and the human's motion data is recorded. The recorded motion data is mapped to the robot to mimic the human's motion according to the motion mapping and control framework in [Lim et al. \(2022\)](#). During the data collection procedure, none of the external forces are applied to the upper body of the robot, but only ground reaction forces are exerted on the feet and the contact wrenches are measured by FTS on the foot. To ensure that any self-collision does not occur, a learning-based self-collision avoidance algorithm in [Koptev](#)

[et al. \(2021\)](#) is implemented in the IK solver, and the random motion is tested in the simulation before the data collection on the real robot. During the random motion of the upper body, the robot's leg was walking or standing with about a 1:1 ratio. The robot's leg controls the center of mass of the robot to be at the center of the support polygon while the robot is standing. We collected about 15 minutes of the random motions of one person and augmented the data 4 times by modifying the speed of the motion and adding offsets to the target positions of two hands. Consequently, The total amount of the collected data is 1 hour resulting in about 3.6M samples which is the same as the *Random Walking Data*.

To obtain decent generalization capability from deep learning, largely distributed data is essential which covers the potential target tasks. This is because the data-driven method inherently shows superior generalization ability for interpolation problems, but it shows poor performance for extrapolation problems. From this perspective, random actions are used in both data sets to generate the large distribution of the position states such as random walking commands and random motions within feasible constraints. For the velocity and desired torque states, however, the distribution of the training data can not cover all the collision data because the training data only contains collision-free data except for ground reactions on the feet. When a collision occurs on the knee link, for example, the accuracy of the estimation decreases because the velocity and torque of the collision-engaged joints abruptly surge in the swing phase, which is not observed in the training data.

To mitigate this problem, the *Random Torque Exploration* (RTE) method is introduced. Random torques are added to the desired torques in every joint except joints of the support leg to explore more distributed data during data collection. The random torque is designed as a step function with random timing and random magnitude. The duration of the on/off period is sampled from the uniform distribution $\mathcal{U}(0.1, 0.5)$ s, and the joint torque is also sampled from the uniform distribution $\mathcal{U}(-\bar{\tau}_{rnd}, \bar{\tau}_{rnd})$ Nm. This random torque sampling is performed independently for each joint. The maximum magnitude of the random torque ($\bar{\tau}_{rnd} = -\underline{\tau}_{rnd}$) was determined as large as possible until the robot maintains its stability; $\bar{\tau}_{rnd,j}$ for leg and waist joints is 50 Nm, and $\bar{\tau}_{rnd,j} = [3, 15, 15, 10, 10, 5, 5, 5]$ Nm for the arm joints. Although this random torque can not cover all the possible collisions with the limited maximum value, it is observed that applying random torque enhances the training results especially when the collision occurs. Validation for this can be found in Section 6.2.

4.4 Training

The network is trained in a supervised learning manner. The parameters of the GRU network are trained using Truncated Back Propagation Through Time (TBPTT) with a specific time horizon as specified in the last column of Table 1. The training epoch and the batch size are 200 and 64, respectively. The Adam optimizer was used for the parameter update with the default betas (0.9, 0.999) because it shows the best performance among the three optimization algorithms [SGD, RMSprop, Adam]. The linear learning rate scheduler is used to decrease the learning rate from the initial

value of 0.05 to 0.0005 during the first 100 epochs. It is observed that the linear learning rate scheduling shows more stable and better learning results than the constant learning rate.

Recall the Problem 3.1 where the optimal mean and variance functions of the normal distribution should be found to estimate the true uncertainty torque. Let's interpret this problem within a deep learning framework where the optimal function of the normal distribution is a neural network $f_{\theta}(\mathbf{X}_p)$:

$$[\hat{\tau}_{u,\theta^*}(\mathbf{X}_p), \hat{\sigma}_{u,\theta^*}^2(\mathbf{X}_p)] = f_{\theta^*}(\mathbf{X}_p) \quad (18)$$

$$\theta^* \approx \arg \max_{\theta} \sum_{i=1}^{n_T} \sum_{j=1}^{n_j} \mathcal{N}(\tau_{u,j}^i | \mathbf{X}_p^i, \theta) \quad (19)$$

where n_T is the number of training data samples and n_j is the number of joints for the uncertainty torque. The conditional probability of the normal distribution is expressed in the exponential function as

$$\mathcal{N}(\tau_{u,j} | \mathbf{X}_p, \theta) = \frac{1}{\sqrt{2\pi\hat{\sigma}_{u,j,\theta}^2}} \exp\left(\frac{-(\hat{\tau}_{u,j,\theta} - \tau_{u,j})^2}{2\hat{\sigma}_{u,j,\theta}^2}\right) \quad (20)$$

So, if the natural log is taken on the right side of (19), the Gaussian negative log-likelihood loss function L is obtained as described in Nix and Weigend (1994):

$$\theta^* \approx \arg \min_{\theta} \sum_{i=1}^{n_T} \sum_{j=1}^{n_j} -2 \ln (\mathcal{N}(\tau_{u,j}^i | \mathbf{X}_p^i, \theta)) \quad (21)$$

$$L = -2 \ln (\mathcal{N}(\tau_{u,j}^i | \mathbf{X}_p^i, \theta)) = \ln(\hat{\sigma}_{u,j,\theta}^2) + \frac{(\hat{\tau}_{u,j,\theta} - \tau_{u,j})^2}{\hat{\sigma}_{u,j,\theta}^2} \quad (22)$$

To prevent dividing by zero, the estimated standard deviation $\hat{\sigma}_u$ is clipped with the minimum value (1e-6) for the numerically stable loss function calculation.

5 Experiment

The experiments are conducted using the real robot and the simulation environment. Throughout all the experiments, a torque-controllable full-size humanoid robot, TOCABI, is used, and only the results of one side of the leg and arm limbs are demonstrated because the other side limb shows similar results. The MuJoCo simulator in Todorov et al. (2012) is used for the simulations. For the robot's hardware specifications, TOCABI has 39 DoF (16 in both arms, 12 in both legs, 3 in the waist, 2 in the neck, and 6 in the virtual joint). The joint names are displayed in Figure 4. The height is about 1.8m and the weight is around 100 kg. The robot has FTS on each foot, an IMU sensor on the pelvis, and a motor side encoder in each joint. Each actuator consists of an electric brushless motor, a harmonic gear with 100:1 gear reduction, and an Elmo motor controller which controls the current of the motor. For the robot's software framework, ROS and real-time Linux kernel are used. The torque control frequency is 2 kHz in both the simulation and real robot experiment while the data is collected in 1 kHz.

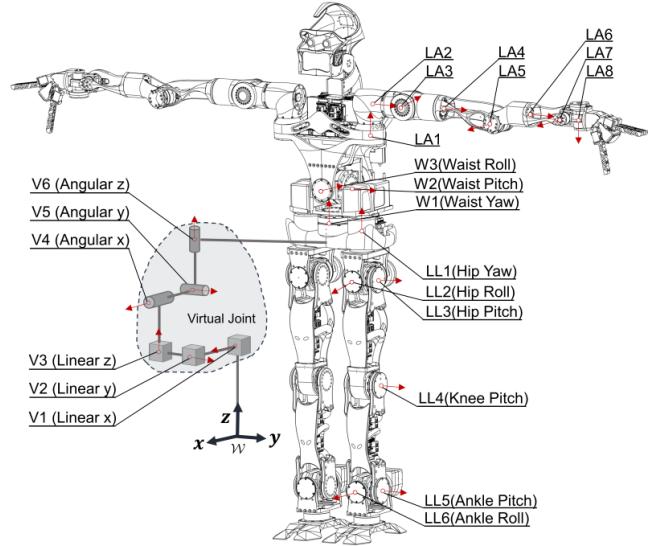


Figure 4. A humanoid robot, TOCABI, used in the experiment and its joint names. Only the joint names in the left limbs are displayed due to their symmetry.

Table 3. The sensor noise used in the simulation. The noise is Gaussian noise with zero mean and each variance.

Variable	Joint position	Joint velocity	Base linear acceleration	Base angular velocity
Noise variance	1e-7	2e-3	1e-4	5e-3

Table 4. The joint friction parameter used in the *All uncertainty* simulation.

Parameter	f_c	f_s	v_s	k_{vf}	k_{lf}	$\bar{\tau}_{loss}$
Value	5.0	2.0	1.51	4.0 (legs) 3.0 (otherwise)	0.002	10.0 Nm (legs) 8.0 Nm (otherwise)

5.1 Comparison of External Joint Torque Estimation Performance in Simulation

In this section, the estimation performance of MOB-Net is validated in the simulation environment where the ground truth of the estimation can be obtained easily. Three different levels of uncertainty are organized to measure the estimation performance of MOB, the backbone of MOB-Net, according to the uncertainty components, and to compare with the learning performance of MOB-Net. The three uncertainty levels are listed below.

- *Ideal*: The uncertainty torque is zero. All the inertia parameters of the robot are perfectly identified and the actuator produces the desired torque accurately without friction.
- *Sensor Noise*: Sensor noise is artificially added to the sensor measurements. The uncertainty is only from the sensor noise and this uncertainty is the baseline of the uncertainty torque learning. All the sensor noises are Gaussian noise with zero means, and the noise variances of each variable are summarized in Table 3.
- *All Uncertainty*: In this environment, three uncertainty components are intentionally added to the ideal simulation: sensor noise, modeling error, and joint friction. The modeling error is induced by decreasing the mass and inertia parameters of each link of the nominal model by 10 %. The nonlinear Stribeck joint

Table 5. Simulation result of the forward walking on uneven terrain. RMSE of the estimated external joint torque is summarized according to the uncertainty level and the estimation method. The dimension of all the values is [Nm].

Left Leg Joint							
Uncertainty level	Method	LL1	LL2	LL3	LL4	LL5	LL6
Ideal	MOB	0.00	0.01	0.01	0.01	0.01	0.01
Sensor noise	MOB	0.44	1.69	1.49	1.21	1.36	1.16
	MOB	7.73	24.61	18.67	33.60	17.63	13.54
							19.30 (+18.08)
All uncertainty	MOB-Net	1.45	1.05	1.53	1.31	1.35	1.16
	FTS-e2e	1.34	1.41	2.47	1.34	1.50	1.00
							1.51 (+0.29)

friction torques, τ_{sf} , are applied for each joint and the load-dependent joint friction torques, τ_{lf} , are only applied to the leg joints.

$$\begin{aligned}\tau_{f,j} &= \tau_{sf,j} + \tau_{lf,j} \\ \tau_{sf,j} &= -\text{sgn}(\dot{q}_j)(f_s + (f_s - f_c) \exp(-|\dot{q}_j/v_s|)) \\ &\quad - k_{vf} \dot{q}_j \\ \tau_{lf,j} &= -\text{sgn}(\dot{q}_j)(k_{lf} |\tau_{m,j}|^2)\end{aligned}$$

Additionally, the friction loss option in the MuJoCo simulator is activated which ignores the joint torques whose magnitude is below the threshold torque, $\bar{\tau}_{loss}$. All the parameters for the joint friction in the simulation are summarized in Table 4.

Each MOB-Net for each limb was trained using the corresponding training data that is collected in the simulation environment as shown in Figure 3 (a). For comparisons, two different methods are implemented below.

- *MOB*: a discretized version of momentum observer introduced in Bledt et al. (2018). $K_0 = 100$ is used for MOB. This MOB is also used for the uncertainty learning of MOB-Net.
- *FTS-e2e*: our previous method in Lim et al. (2023) uses supervised learning in an end-to-end manner to directly learn the external joint torques of legs which are measured by FTS on each foot and transformed to the joint torques. FTS-e2e network is trained using the same training data and the same network structure as MOB-Net. FTS-e2e infers the external joint torques and also the variance.

The testing data for the leg joints contains the forward walking data for one minute on uneven terrain with a step duration of 0.7 s and a maximum foot height of 5.5 cm. In Table 5, the root-mean-square error (RMSE) of the external joint torque estimation is summarized according to the joint, uncertainty level, and the estimation method. In the *Ideal* environment, the model-based method, MOB, can estimate the external joint torque accurately with roughly 0.01 Nm errors for each joint. When the sensor noise is added to the sensor measurements, the estimation errors increase by 1.31 Nm on average. However, in *All Uncertainty* environment, the average estimation error of MOB increases by 19.30 Nm due to the additional modeling error and the joint frictions. In the same uncertainty level, MOB-Net learns the uncertainty torque and cancels out the uncertainty

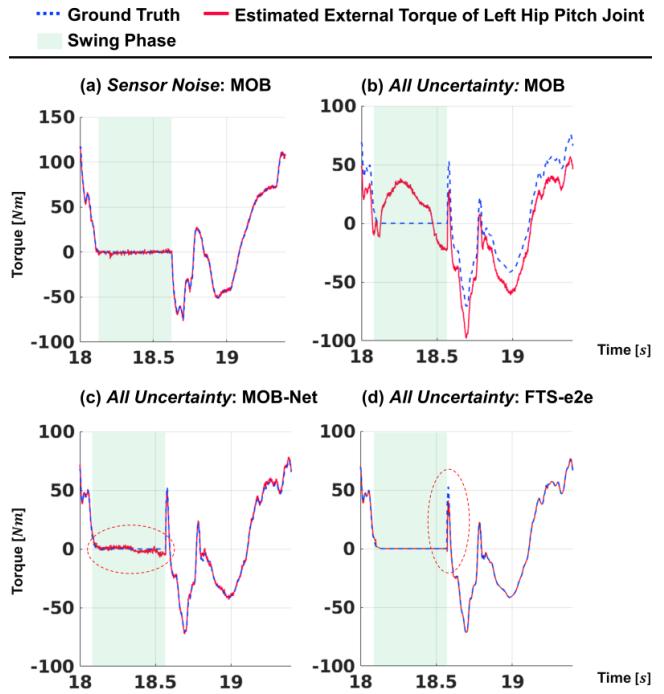


Figure 5. Simulation results of the forward walking on uneven terrain. The estimated external joint torque of the LL3 (left hip pitch) joint is plotted according to the uncertainty level and the estimation method. The robot walks forward on uneven terrain with a step duration of 0.7 s and the plots show one cycle of the locomotion.

torque from MOB resulting in significant improvement in the estimation by reducing the estimation error to 1.31 Nm on average which is just 0.09 Nm (6.7%) larger than the MOB error in the *Sensor Noise* environment. The MOB errors in the *Sensor Noise* environment are regarded as the minimum error that MOB-Net or FTS-e2e can achieve in *All Uncertainty* because, unlike the other uncertainty components, the random sensor noise can not be inferred correctly. FTS-e2e has similar estimation errors to MOB-Net for all joints except for the LL3 joint.

Figure 5 shows the ground truth and the estimated external joint torque of the hip pitch joint (LL3) for one cycle of the walking period in the test data which includes the swing phase and the supporting phase. The swing phase is displayed with the green area on the plots. Figure 5 (a) shows the MOB residual signals in the *Sensor Noise* environment. The residual signal is noisy but estimate the true external joint torque closely. On the other hand, the estimated external joint torque of the MOB significantly deviates from the true values in Figure 5 (b) due to the modeling errors and the joint frictions resulting in large errors (18.67 Nm). However, in Figure 5 (c), MOB-Net calibrates the erroneous signals of MOB by subtracting the uncertainty torque and results in a similar estimation performance to MOB with the sensor noise. The MOB-Net shows noisy signals similar to the MOB residual as marked with a dotted red circle in Figure 5 (c). FTS-e2e shows more smooth estimation signal than MOB-Net as shown in Figure 5 (d). However, when the target value changes fast, FTS-e2e can not estimate the sharp signal as shown in the dotted red circle in Figure 5 (d). This is because the estimated signals from the neural network are

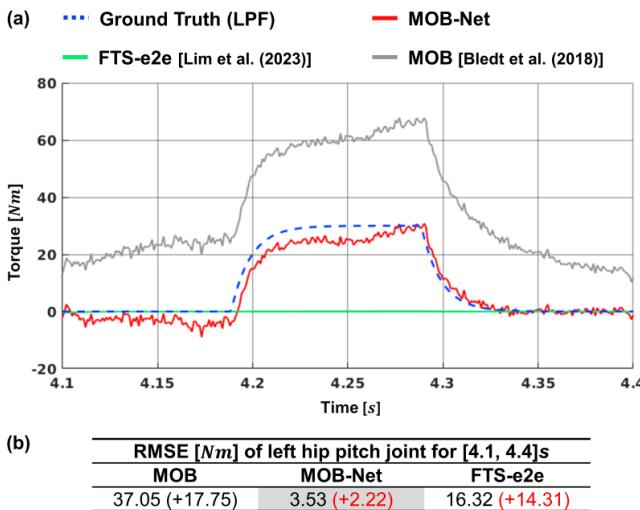


Figure 6. Simulation result of the step disturbance in the LL3 (left hip pitch) joint. (a) The estimated external joint torque of the LL3 joint is displayed when the external joint torque of 30 Nm is applied to the LL3 joint for 0.1 s during the swing phase. (b) The RMSE of the external torque estimation error in the LL3 joint for the plot range [4.1, 4.4] s.

smooth inherently but MOB-Net utilizes both MOB and the neural networks and the sharp estimation signals of the MOB appear in the estimated values of MOB-Net.

Walking data with a disturbance on the swing leg is tested to investigate the estimation performance for the unseen data. Unlike the external torque propagated from the ground reaction forces through the foot link, disturbance on the swing leg through the other link except the foot link generates unseen patterns from the training data. Therefore, a step external joint torque of 30 Nm is applied on the hip pitch (LL3) joint during the swing phase to mimic the collision on the knee. As shown in Figure 6 (a), MOB residual signal (gray line) resembles the changes of ground truth (dotted line) even if it shows large offset and errors of 37.05 Nm due to the uncertainty torque. MOB-Net estimates the external torque closely with relatively small errors (3.53 Nm) which is larger than the estimation errors of the straight walking test by 2.22 Nm. However, FTS-e2e can not estimate the external torque and infers almost zero external torques for the disturbance resulting in large errors (16.32 Nm). Even though MOB-Net and FTS-e2e use the same training data and the same network, the combination of the model-based method and the data-driven method (MOB-Net) shows robust performance for the unseen data compared with the sole data-driven method (FTS-e2e) or model based-method (MOB). The robustness of MOB-Net for the unseen data is possible due to the use of MOB as a base algorithm of MOB-Net. Even if the GRU network causes large estimation errors for the unseen data, MOB can capture the changes of the external torque based on the dynamics of the robot basically and the estimation error is not significantly increased in MOB-Net unlike the end-to-end learning method, FTS-e2e. Note that the ground truth in Figure 6 (a) is the low pass filtered signal of the step torque with the same cutoff frequency of MOB because MOB estimates the low pass filtered external torque theoretically and it is fair to compare with the low pass filtered true value.

The other limb groups (LA, W, V) are also validated in the simulation. Unlike the legs, MOB-Net for the upper body limbs (LA, W) does not include the desired torque in the input vector and the training data only consists of the collision free data. The networks of the upper body are trained using the random motion data explained in Section 4.3. The test data consists of the random upper body motion and the random walking commands with 0.7 s step duration. Three test data sets are tested with the same random motion: Collision free motion, Left hand load (10 N), and Left hand load (30 N). In *Collision free motion* data set, the robot did not receive any external forces while walking and moving the upper body randomly, and this data set is for the baseline of the learning performance as an in-distribution data. *Collision free motion* contains data from three different uncertainty levels. In two *Left hand load* data sets, the humanoid robot receives a step downward force on the left hand along the gravitational direction for 5 seconds, [10, 15] s while walking and moving the upper body randomly as similar to the collision free motion data. The two *Left hand load* data sets are tested to measure the estimation performance for the external force data, i.e., unseen data. Note that the RMSE of left hand load tests is calculated only for the disturbance duration ([10, 15] s) while the RMSE of collision free motion is calculated for the entire test data ([0, 60] s).

All the test results of the upper body joints and the virtual joints for the three data sets are summarized in Table 6. In the *Collision free motion*, the results of MOB show a similar tendency to the results of the left leg across the uncertainty levels. For the ideal environment, the estimation error is almost zero, but the error increases to 0.80 and 10.14 Nm for the upper body joint on average as the uncertainty components are added in the sensor noise and all uncertainty, respectively. Additionally, the estimation errors of MOB in virtual joints increase as the uncertainty components are added. For the virtual joints, joint friction torque is not applied and the modeling error is major in the estimation error. However, MOB-Net effectively reduces the estimation error to 2.07 Nm on average for upper body joints and to 4.73 N and 2.90 Nm on average for the virtual joints in the linear and angular joints, respectively.

In the *Left hand load (10 N)* results, all the methods have similar errors to the results of the collision free motion test, which means that MOB-Net for the upper body joints and the virtual joints can generalize to the small disturbance, 10 N force in the -z-axis. In the *Left hand load (30 N)* results, compared to the results of 10 N load, the estimation error of MOB-Net increases by 0.30 Nm on average for the upper body, but the errors of the virtual joint network increase by 2.42 N and 1.45 Nm for the linear and angular virtual joints, respectively. Similar or smaller amounts of increase in errors are also observed in the MOB results. These results demonstrate the robust performance of MOB-Net over the external forces on the hand although the training data contains only the collision-free motion.

Figure 7 shows the ground truth of external joint torque, the estimated value of MOB-Net, and the estimated value of MOB for the load of 30 N on the left hand with all uncertainty. The load is applied on the left hand of the robot in a step function for 5 seconds ([10, 15] s) and the ground truth of the external joint torque is calculated using

Table 6. Simulation result of the upper body joints and the virtual joints for the three test data sets: *Collision free motion*, *Left hand load (10 N)*, and *Left hand load (30 N)*. RMSE of the estimated external joint torque or force are summarized according to the uncertainty level, estimation method, and data set.

Uncertainty level	Method	Upper body joint										Virtual joint									
		W1	W2	W3	LA1	LA2	LA3	LA4	LA5	LA6	LA7	Avg (W+LA)	V1 [N]	V2 [N]	V3 [N]	Avg (V lin)	V4	V5	V6	Avg (V ang)	
Data Set 1: Collision free motion																					
Ideal	MOB	0.01	0.02	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.05	0.08	0.08	0.07	0.02	0.04	0.01	0.03	
Sensor noise	MOB	1.85	2.09	2.42	0.58	0.55	0.49	0.35	0.26	0.06	0.10	0.05	0.80	3.98	3.37	3.77	3.71	2.46	2.04	1.59	2.03
All uncertainty	MOB	9.43	11.77	7.91	7.52	9.95	10.21	10.31	10.97	12.70	10.51	10.28	10.14	5.45	7.24	109.21	40.63	4.75	18.72	2.32	8.60
	MOB-Net	2.84	2.75	3.64	2.72	1.45	1.63	1.91	1.63	0.84	1.31	2.06	2.07	4.87	4.77	4.56	4.73	3.41	3.34	1.95	2.90
Data Set 2: Left hand load (10 N)																					
All uncertainty	MOB	9.34	9.73	8.02	5.95	10.71	9.21	9.99	9.99	12.18	9.96	8.96	9.46	7.77	4.19	110.44	40.80	4.01	16.82	3.58	8.13
	MOB-Net	2.95	2.45	2.88	2.47	1.04	1.89	1.60	1.73	0.66	0.91	2.08	1.88	5.51	3.79	6.25	5.18	2.93	2.44	1.81	2.39
Data Set 3: Left hand load (30 N)																					
All uncertainty	MOB	11.44	10.38	11.88	6.51	8.33	9.70	9.74	10.66	10.17	8.53	6.97	9.48	5.99	12.77	109.66	42.81	7.81	16.75	3.43	9.33
	MOB-Net	2.50	3.01	3.79	2.37	1.73	2.15	1.37	1.96	1.48	1.18	2.42	2.18	5.84	9.71	7.25	7.60	5.21	3.64	2.68	3.84

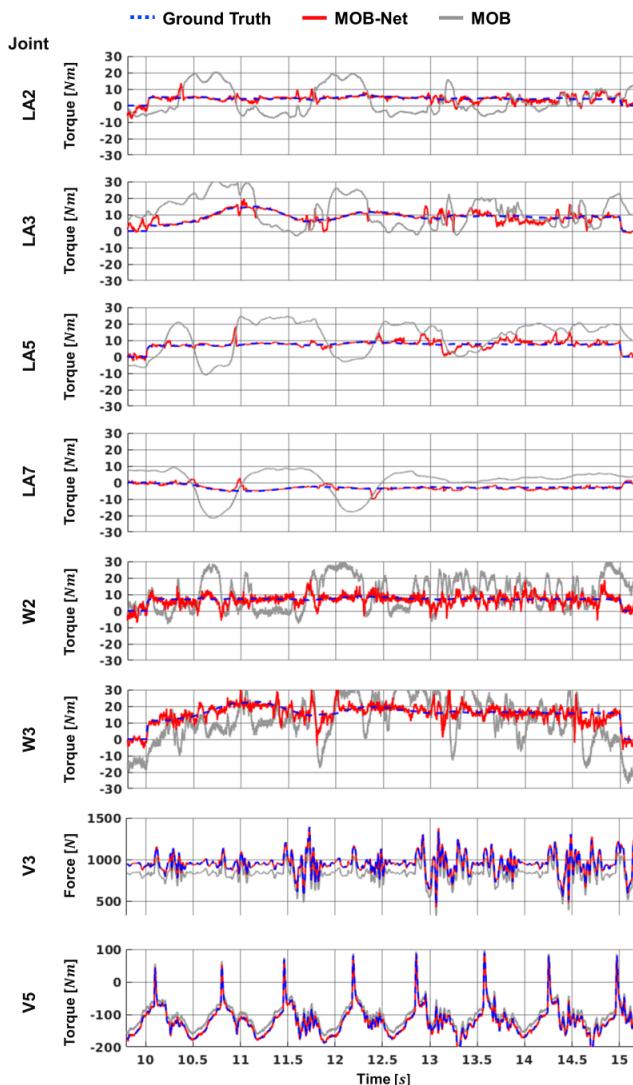


Figure 7. Simulation result of the partial upper body joints and the virtual joints for the *Left hand load (30N)* data set. A downward linear force of 30 N is applied on the left hand during [10, 15] s while the robot is walking and moving the upper body.

the Jacobian matrix of the hand link. In general, MOB-Net follows the ground truth but the estimated value of MOB-Net shows minor noisy signals due to the sensor noise while MOB has larger errors due to the uncertainty torque. In the last two plots of Figure 7, two virtual joints, V3 (linear

joint in the z-axis) and V5 (angular joint in the y-axis), are demonstrated. In the virtual joints, the estimated value of MOB-Net tracks the true value but MOB shows large bias signals due to the modeling error with a 10% lighter nominal model.

5.2 Comparison of External Joint Torque Estimation Performance in Real Humanoid Robot

In this experiment, the estimation performance of MOB-Net is validated using a real humanoid robot. Similar to the previous simulation section, the proposed method is compared with two methods (FTS-e2e, MOB), and an additional model-based method is also implemented for the comparison: *MOB-fric*. Proposed in Lee et al. (2015), *MOB-fric* utilizes a friction model to estimate more accurate external joint torques. The friction model requires joint velocity and joint torque for the estimation of the joint friction. The friction model consists of coulomb, static, and viscous friction, and the model parameters are regressed using the same training data of MOB-Net.

The estimation performance of the legs is tested first. The testing data contains the random walking data for one minute in the uneven terrain with a step duration of 0.7 s and a maximum foot height of 5.5 cm. The estimated uncertainty torque of the left leg is plotted in Figure 8 with the target value which is calculated by subtracting the external joint torque measured by FTS on the foot from the residual vector of MOB. The weight of the foot link is calibrated from the FTS measurements. The dotted blue line indicates the target value of the uncertainty torque learning, the red line indicates the estimated uncertainty torque from MOB-Net, and the red area around the estimated uncertainty torque shows the estimated standard deviation of the uncertainty torque from MOB-Net. This plot shows one cycle of the walking period consisting of the swing phase (green area) at first and the supporting phase following the swing phase. The estimated uncertainty torque follows the target value closely with the RMSE of [0.980 6.266 4.191 2.731 0.917 0.598] Nm. It is noted that in Figure 8 the estimation error of uncertainty torque increases when the robot's swing foot lands (right after the swing phase, the green area) and the estimated standard deviation of the uncertainty torque in the LL3 joint

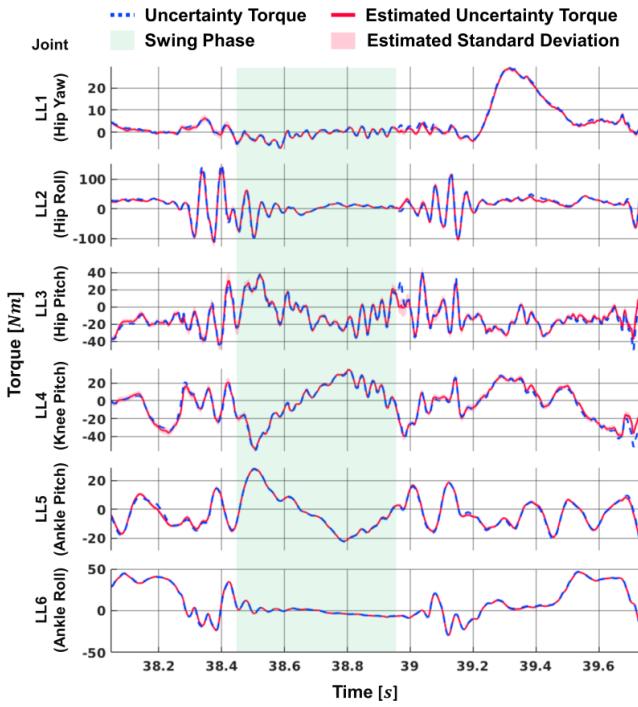


Figure 8. Uncertainty torque estimation of the joints in the left leg using MOB-Net and the corresponding target value.

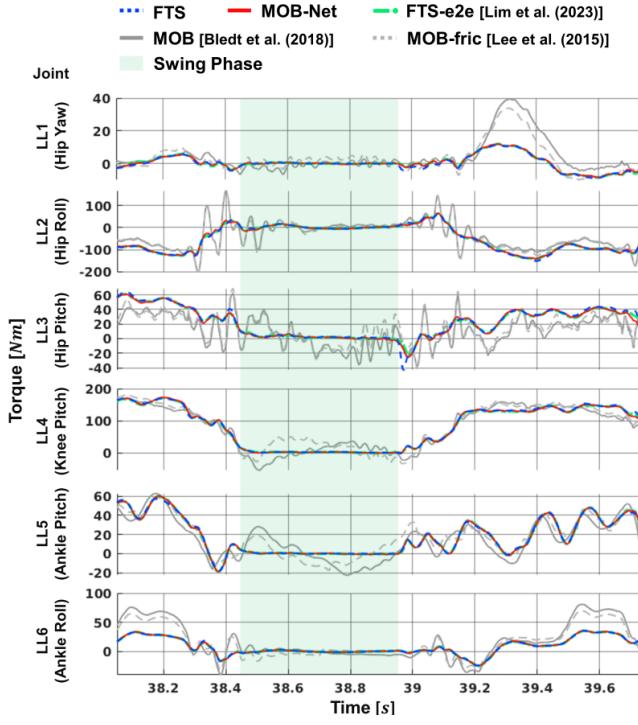


Figure 9. Comparison of external torque estimation according to the estimation methods in the joints of the left leg.

also increases accordingly due to the large sensor noise of the landing impact.

The estimation result of the external joint torque according to each method is depicted in Figure 9. The blue dotted line is the ground truth of the external joint torque measured by the FTS. The red line indicates the estimated external torque of the proposed method, MOB-Net. The green line indicates the estimated external torque of our previous

Table 7. RMSE and r-RMSE of External Torque Estimation of Various Methods for Left Leg.

Method	Statistics	Left leg joint						avg
		LL1	LL2	LL3	LL4	LL5	LL6	
MOB	RMSE [Nm]	8.589	37.380	25.673	26.921	15.238	18.121	21.987
MOB	r-RMSE [%]	33.76	32.65	18.94	13.18	15.24	35.09	-
MOB-fric	RMSE [Nm]	6.439	36.907	23.960	23.398	11.929	14.556	19.531
MOB-fric	r-RMSE [%]	25.31	32.24	17.68	11.45	11.93	28.18	-
MOB-Net	RMSE [Nm]	0.908	6.266	4.191	2.731	0.917	0.598	2.602
MOB-Net	r-RMSE [%]	3.57	5.47	3.09	1.34	0.92	1.16	-
FTS-e2e	RMSE [Nm]	0.727	4.054	4.372	2.719	0.987	0.531	2.232
FTS-e2e	r-RMSE [%]	2.86	3.54	3.23	1.33	0.99	1.03	-

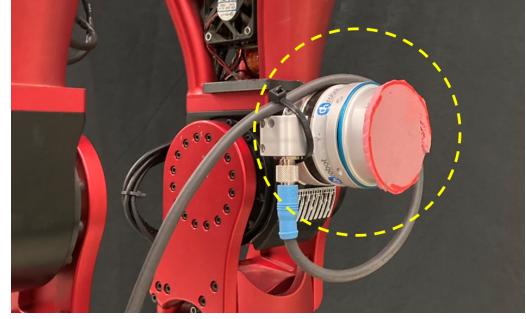


Figure 10. Experimental setup for the knee collision. An FTS attached on the left knee link to measure the external torque due to knee collisions.

method, FTS-e2e. Lastly, the gray line and the dotted gray line show the estimated external torques from MOB and MOB-fric, respectively. As shown in Figure 9, the two data-driven methods, MOB-Net and FT-e2e, show more correct estimation results following the target value closely than the two model-based methods, MOB and MOB-fric. During the swing phase (green area), MOB-Net and FTS-e2e estimate near zero torques while MOB and MOB-fric estimate non-zero torques due to the uncertainty torques.

The RMSE and relative RMSE (r-RMSE) of the external torque estimation for each method are summarized in Table 7. r-RMSE is the percentage of the RMSE relative to the maximum external joint torque measured by FTS for each joint. The result shows that MOB-Net has an error of 2.602 Nm on average. Among the leg joints, the hip roll joint (LL2) has the largest estimation error while ankle joints (LL5 and LL6) have less than 1 Nm errors. In this random walking test data, FTS-e2e shows a similar estimation performance to MOB-Net resulting in 2.232 Nm error on average. In FTS-e2e, LL3 joint shows the largest error among the leg joints. The two model-based methods, MOB and MOB-fric, show much higher estimation error than the data-driven methods due to the model uncertainty while the friction model reduces the estimation error by 2.456 Nm on average.

To test the external joint torque estimation performance for the unseen data in the real humanoid robot, an external force is applied to the left knee link when the left leg is in the swing phase while the robot is walking in place with a step duration of 0.7 s. The training data does not include such collision data on the swing leg. To measure the external joint torque that occurred from the knee collision correctly, an FTS is attached to the left knee link of the robot using 3D-printed parts as shown in Figure 10, and the external joint torque from the knee collision is calculated using the contact Jacobian of the FTS on the knee. The experimental results of

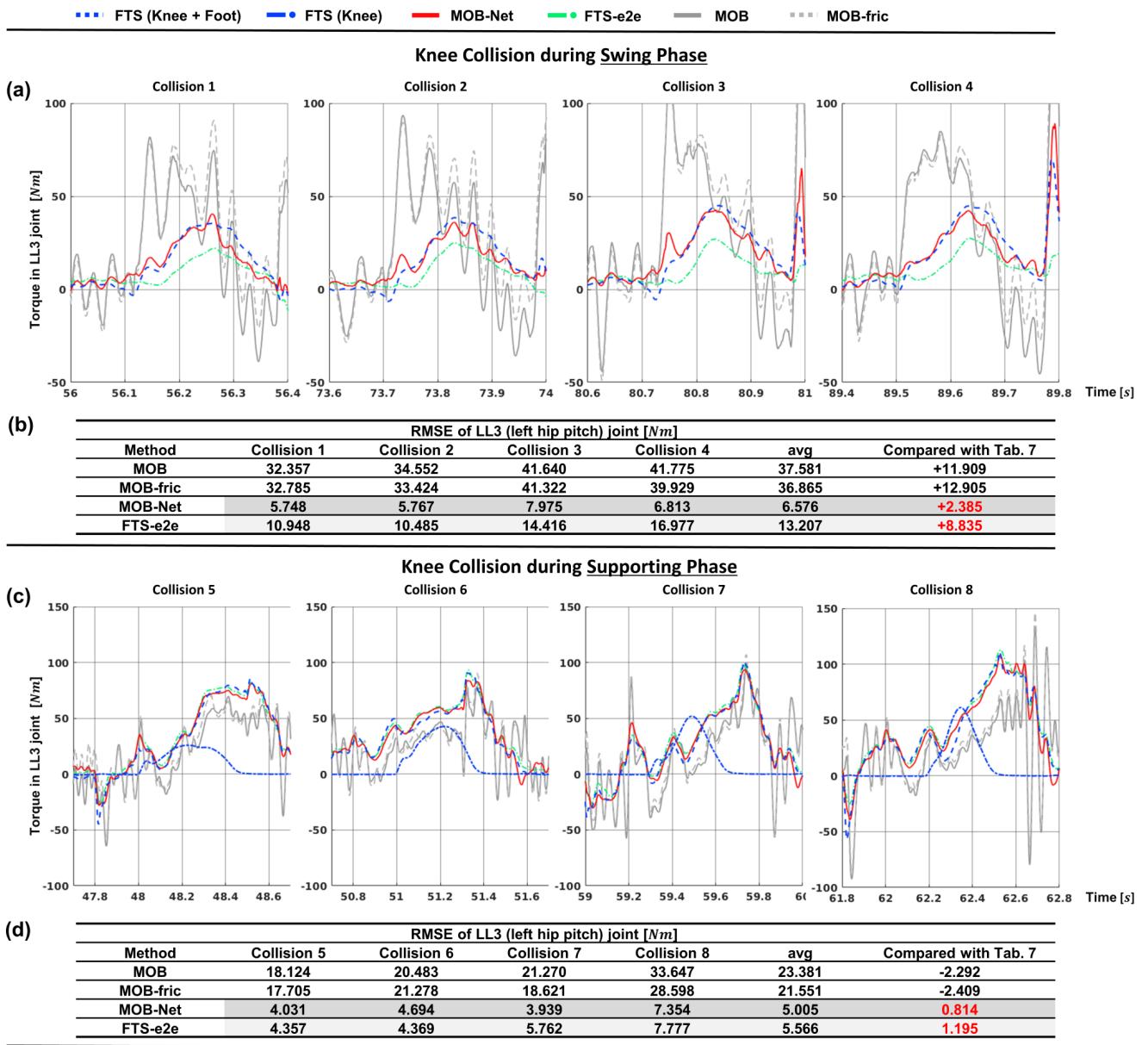


Figure 11. Comparison of external joint torque estimation performance for the knee collisions. Collisions are made during both the swing phase and the supporting phase. (a), (c) The measured torque and the estimated torque of each method in the LL3 joint for 4 collision cases during the swing and supporting phase, respectively. (b), (d) RMSE of the estimation for each method and collision case. The changes in the average RMSE compared to the error of the random walking test are displayed in the last column.

the four collision cases during the swing phase are plotted in Figure 11 (a). These plots show the measured and estimated external joint torque in the LL3 (left hip pitch) joint which is mainly affected by the front knee collision. The dotted blue line indicates the measured external joint torque using the FTS on the knee and the foot link to measure the collision force on the knee and the ground reaction force on the foot; Although the knee collision starts during the swing phase, the knee collision can last after the foot landing.

As displayed in Figure 11 (a), MOB-Net shows the best estimation performance among the four methods. On the other hand, FTS-e2e estimates the external joint torque with delayed and lower values resulting in large errors. The average errors of the estimation for the four cases are summarized in Figure 11 (b). The average error of MOB-Net for the knee collisions is 6.576 Nm in the LL3 joint which is 2.385 Nm larger than the errors of the LL3 joint in

the random walking test set (in-distribution data). However, the average error of FTS-e2e significantly increases to 13.207 Nm which is 8.835 Nm larger than the RMSE of the LL3 joint in the random walking test set. This result validates the robust performance of MOB-Net for the knee collision that is unseen data while the estimation of FTS-e2e deteriorates for the unseen data, which is similar to the result of hip pitch disturbance in the simulation. This is the well-known limitation of the deep learning method, but MOB-Net reduces the negative effect of the learning method for the unseen data by combining both the model-based observer and the deep learning together.

The knee collision experiment during the supporting phase is also performed to investigate the external torque estimation performance when the support leg is disturbed. The left knee is pushed when it is in the supporting phase and the external torques of the left leg are measured using

Table 8. Real robot experiment result of the upper body joints and the virtual joints for the two test data sets: *Collision free motion* and *External wrench on the hand*. RMSE of the estimated external joint torque or force are summarized according to the estimation method, and the data set.

Method	Upper body joint												Virtual joint							
	W1	W2	W3	LA1	LA2	LA3	LA4	LA5	LA6	LA7	LA8	Avg (W+LA)	V1 [N]	V2 [N]	V3 [N]	Avg (V lin)	V4	V5	V6	Avg (V ang)
Data set1: Collision free motion																				
MOB	14.44	14.35	10.55	3.14	3.73	4.31	3.45	3.27	1.94	1.31	1.18	5.61	36.69	55.31	41.41	44.47	44.32	30.04	10.64	28.34
MOB-fric	12.86	13.38	9.18	2.68	3.72	3.12	1.94	4.03	1.93	1.40	1.24	5.04	-	-	-	-	-	-	-	-
MOB-Net	5.40	2.75	4.46	1.06	1.07	1.09	0.80	0.65	0.37	0.19	0.27	1.65	14.18	17.12	22.58	17.96	11.62	11.91	4.73	9.42
Data set2: External wrench on the hand																				
MOB	16.02	12.93	10.06	4.58	3.51	4.82	3.93	3.94	1.65	1.10	1.38	5.81	14.24	20.11	12.32	15.56	12.42	18.45	10.44	13.77
MOB-fric	14.24	11.08	8.46	3.96	2.42	3.17	2.35	3.04	2.11	0.96	1.24	4.82	-	-	-	-	-	-	-	-
MOB-Net	4.26	3.77	3.18	2.39	1.71	1.47	1.36	1.03	0.31	0.49	0.37	1.85	5.94	7.79	7.62	7.12	4.45	3.54	3.67	3.89

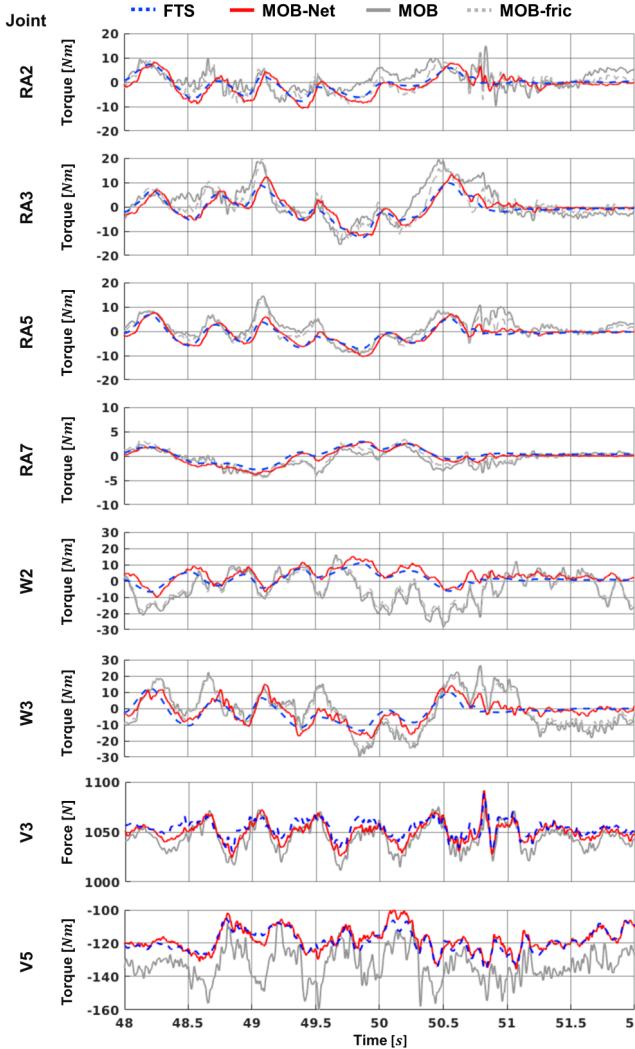


Figure 12. Real robot experiment result of the partial upper body joints and the virtual joints for the *External wrench on the hand* data set. A random external wrench is applied on the left hand.

both FTS on the knee and the foot. Figure 11 (c) displays the measured external torque and the estimation results of the LL3 joint for the four collision cases. The estimated torque of MOB-Net and FTS-e2e follows the measured external torque tightly even if the support leg is pushed. However, the two model-based methods still show much higher errors. Figure 11 (d) summarizes the estimation errors of each

method. The estimation errors of MOB-Net and FTS-e2e are roughly 1 Nm larger than the random walking test results of the LL3 joint in Table 7. The difference in errors between MOB-Net and FTS-e2 for the supporting knee collisions is smaller than the difference for the swing knee collisions. This is because the training data includes various disturbance patterns of the supporting leg from the random walking data on uneven terrain and the supporting knee collision can be regarded as in-distribution data, but the swing knee collision is not covered by the training data (out-of-distribution data).

The upper body and virtual joints are tested for two test data sets: *Collision free motion* and *External wrench on the hand*. *Collision free motion* contains random upper body motion while the robot is walking randomly (in-distribution data). In *External wrench on the hand*, the experimenter applies a random external wrench to the right hand of the robot while the upper body is moving randomly. All the test results are summarized in Table 8. MOB-Net results in much smaller errors than the model-based methods across all the joints and the test sets. Although MOB-Net is trained only using collision free motion data, MOB-Net can estimate external torque from the external wrench on the hand as validated in the second data set (External wrench on the hand). Figure 12 shows the measured and estimated external torques in the upper body and the virtual joints. The estimated torque from MOB-Net tracks the measured external torque by FTS while MOB residual shows large errors.

In summary, MOB-Net has superior estimation performance than two model-based methods for both in-distribution and out-of-distribution data and shows similar performance to FTS-e2e for the in-distribution data. However, for the out-of-distribution data (swing knee collisions), MOB-Net shows more robust estimation results than FTS-e2e.

5.3 Comparison of Collision Detection Performance

In this experiment, the collision detection performance is compared between various methods to validate the superiority of the proposed method. Similar to the previous section, FTS-e2e, MOB, and MOB-fric are selected as comparison methods. Collision is detected when the collision signal (normally estimated external joint torque) is over the threshold value in each joint, and collision detection is only performed for unexpected collision during the swing phase

Table 9. The number of collision detection successes and collision delay of the left leg for detection Method, collision link, and impact direction

Method	Collision Detection Success							Collision Detection Delay [ms]						
	Foot		Ankle		Knee		Total	Foot		Ankle		Knee		Avg
	Front	Side	Front	Side	Front	Side		Front	Side	Front	Side	Front	Side	
MOB-Net-OR	10	10	10	10	10	10	60/60	9.60	6.30	17.20	14.60	15.50	8.20	11.90
MOB-Net-mean	10	10	10	5	8	10	53/60	10.20	6.30	17.20	14.00	19.25	8.20	12.53
MOB-Net-sigma	10	10	9	10	9	10	58/60	10.10	6.60	20.00	15.20	19.67	10.50	13.68
FTS-e2e-mean	10	10	10	4	9	10	53/60	9.50	7.70	34.00	11.50	95.67	11.60	28.33
FTS-e2e-sigma	1	0	0	0	0	0	1/60	261.00	-	-	-	-	-	261.00
MOB	4	10	5	10	1	9	39/60	21.50	18.50	19.00	17.10	31.00	14.11	20.20
MOB-fric	8	10	10	10	2	10	50/60	17.63	10.50	17.70	19.90	29.50	10.20	17.57
MOB-fric-BPF	3	1	7	1	0	1	13/60	11.67	20.00	13.57	17.00	X	19.00	16.25

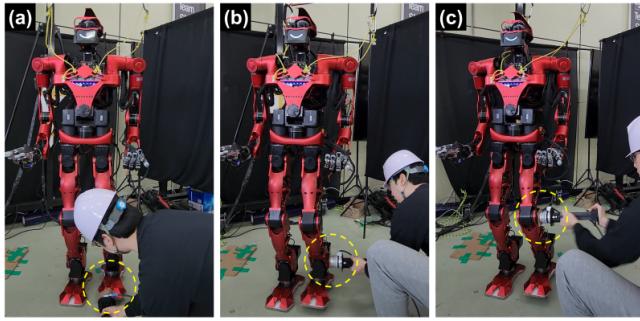


Figure 13. Example snapshots of collision detection experiment. (a) A foot-front collision. (b) An ankle-side collision. (c) A knee-side collision.

except for repetitive foot contacts of walking. The collision threshold of each method is determined not to have any false alarms during 10 minutes of random walking data for a fair comparison, and the unexpected collision among repetitive foot contacts is classified based on several heuristic criteria. Continuous filtering of 5 ms time horizon and low pass filtering of 15 Hz cut-off frequency are also utilized for more robust collision detection performance. The collision detection of the data-driven methods is performed using two collision signals, $\hat{\tau}$ and $\hat{\sigma}$ that are the mean and standard deviation in the network outputs. Thus, three variants of MOB-Net are tested; *MOB-Net-OR* uses both $\hat{\tau}_e$ and $\hat{\sigma}_u$ for collision detection using OR logic operation, *MOB-Net-mean* uses only $\hat{\tau}_e$, and *MOB-Net-sigma* uses only $\hat{\sigma}_u$ for collision detection. In the same way, two variants of FTS-e2e are tested: *FTS-e2e-mean* and *FTS-e2e-sigma*. *FTS-e2e-OR* is not tested because *FTS-e2e-sigma* shows poor collision detection performance. A model-based method, *MOB-fric-BPF*, is implemented additionally using band-pass-filter to suppress the error caused by modeling error as in Cho and Song (2013); Van Dam et al. (2022) in addition to the MOB and MOB-fric. The cutoff frequency of the band-pass filter is chosen as [2, 15] Hz.

The collisions were made by an experimenter using a collision tool equipped with FTS and a rudder pad on the collision side while the robot was walking in place. 10 collisions occurred at each collision point of the left leg with two different directions (front and side), and three different links (foot, ankle, and knee) resulting in a total of 60 collisions. Figure 13 shows three examples of the collision detection experiment. The maximum collision

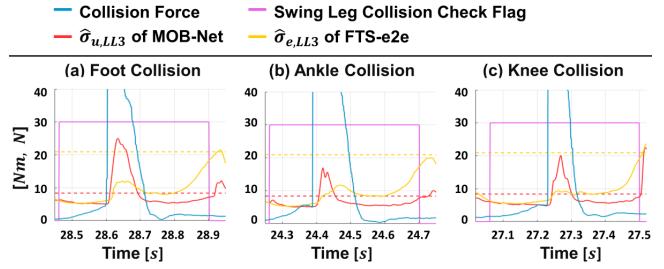


Figure 14. Examples of the estimated standard deviation of MOB-Net and FTS-e2e in LL3 joint for three different front collisions. The dotted line shows the collision detection threshold of each method. (a) A foot collision. (b) An ankle collision. (c) A knee collision.

force ranges in [100, 200] N approximately and the average contact duration is 78 ms. The average and maximum impact momentum of all collisions is 7 Nm and 10 Nm, respectively.

Table 9 summarizes all the collision detection results. MOB-Net-OR shows the best performance without detection failure and with the fastest detection time. Although both MOB-Net-mean and MOB-Net-sigma failed to detect collisions several times, each method complements the other's failure by combining both signals together in MOB-Net-OR. FTS-e2e-mean resulted in similar detection accuracy to MOB-Net-mean for all collisions and similar detection delay for the foot collisions. However, the detection delay of FTS-e2e-mean increased for ankle and knee collisions which is the unseen data resulting in a more than 6 times longer detection delay for the Knee-Front compared to the delay of MOB-Net-OR. This is because the external torque estimation of FTS-e2e shows delayed signals and its error increases for the knee impact (unseen data) as validated in Section 5.2. FTS-e2e-sigma shows poor performance only detecting a single foot-collision which is even too slow to be used for a safe collision reaction. MOB fails to detect 20 front collisions among 30 collisions while successful for the side collisions. The friction model improves the accuracy of the collision detection resulting in 50 successes over 60 collisions and decreases the collision detection delay from MOB. The BPF, however, was not helpful for collision detection unlike the result in Van Dam et al. (2022) because the modeling error is not in the low-frequency domain during dynamic motions such as locomotion which is different from the stationary task performed in the other studies.

Figure 14 displays the behavior of the estimated standard deviation of two data-driven methods for the swing leg collision. The estimated standard deviations of MOB-Net (red line) and FTS-e2e (yellow line) in the LL3 joint are plotted for the three front collisions in Figure 14. The collision force (light blue line) is displayed to show when the collision occurs and the purple line indicates the swing leg collision check flag which is determined from the walking planner and heuristic conditions to only detect unexpected collisions on the swing leg and to ignore repetitive ground contacts of the foot. The dotted line means the collision threshold for each method. As shown in the figure, $\hat{\sigma}_{u,LL3}$ of MOB-Net surges according to the collision, which results in fast collision detection. On the other hand, $\hat{\sigma}_{e,LL3}$ of FTS-e2e fluctuates slightly but under the threshold value. Thus, FTS-e2e-sigma fails to detect collisions because of the high threshold value and low collision signals.

6 Ablation Study

In this section, several ablation studies are provided to validate the network architecture of MOB-Net and the random torque exploration method.

6.1 Network Structure Comparison

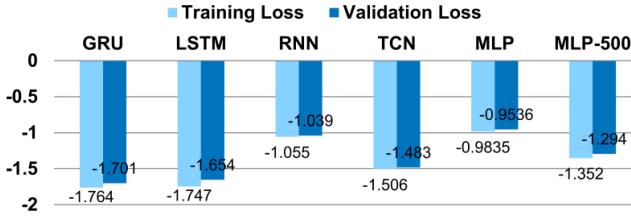


Figure 15. The training loss and the validation loss according to the kind of neural network. GRU shows the lowest losses.

Five different kinds of neural networks are compared to validate the use of the GRU module including two other RNN-series networks. For this experiment, only MOB-Net for the left leg (f_{θ}^{LL}) is compared using the random walking data set, but other limbs show similar results. The five comparison networks are summarized below.

- LSTM (Long Short-Term Memory), [Hochreiter and Schmidhuber \(1997\)](#): The GRU module in MOB-Net is replaced by an LSTM module. The network structure is LSTM(30, 128)-Linear(128, 12) making the learning parameter size (83468) similar to the parameter size of MOB-Net (83712).
- RNN: The GRU module in MOB-Net is replaced by a vanilla RNN. The network structure is RNN(30, 268)-Linear(268, 12). The network size is 83628.
- TCN (Temporal Convolutional Network), [Bai et al. \(2018\)](#): TCN is a variant of 1D-CNN using causal and dilated convolutions, and it is specialized in sequence modeling. TCN with dilation factors $d = 2, 4, 8$ and filter size $k = 3$ is used following the linear output layer. The hidden size is 32 resulting in a model size of 83956 and a dropout of 0.2 is used.

Table 10. Random walking test results of left leg according to the use of random torque exploration.

Method	Statistics	Left leg joint					
		LL1	LL2	LL3	LL4	LL5	avg
MOB-Net W/ RTE	RMSE [Nm]	0.908	6.266	4.191	2.731	0.917	0.598
	rRMSE [%]	3.57	5.47	3.09	1.34	0.92	1.16
MOB-Net WO/ RTE	RMSE [Nm]	1.150	9.134	7.265	4.704	1.418	1.311
	rRMSE [%]	4.52	7.98	5.36	2.30	1.42	2.54

Table 11. Swing knee collision test results of the LL3 joint of the left leg according to the use of random torque exploration.

RMSE of LL3 joint [Nm]					
Method	Collision 1	Collision 2	Collision 3	Collision 4	avg
MOB-Net W/ RTE	5.748	5.767	7.975	6.813	6.576
MOB-Net WO/ RTE	13.293	12.039	15.173	13.218	13.431

- MLP: A simple 2-layer MLP network is used with a hidden size of 55, i.e., MLP(150, 55, 12). The model size is 83227.
- MLP-500: A larger 2-layer MLP network with a hidden size of 500 is also tested resulting in the model size of 756512 which is about 9 times larger than the other networks.

The model size of each network is constrained to be similar to the one of MOB-Net in order to give a fair comparison of the architecture of each network except MLP-500. The average training loss and the validation loss of five runs for each network are compared in Figure 15. The results show that GRU has the lowest loss compared to other networks. LSTM shows a lower loss than the other comparison networks but higher than GRU. The vanilla RNN network shows poor performance compared to the other RNN-series networks (GRU and LSTM) and similar performance with the simple 2-layer MLP network which shows the worst performance. TCN ranked third showing better performance than the other simple architectures. When the model size increases, the performance gain is observed not only for MLP-500 but also for other networks. However, the gain in the estimation accuracy was minor compared to the increased calculation time, and the network size can not be increased further from the current size due to the computation time limit of real-time calculation in 1000hz. Therefore, the model size is compromised considering the learning result and the limited calculation time on the embedded computer of the robot. We conjecture that the reason for the better performance of GRU than LSTM is that the uncertainty torque learning only requires a short history of the joint state while LSTM has an advantage in the task requiring long-term memory such as the natural language process. Therefore, the GRU is efficient and suitable for uncertainty torque learning with constraints of network size.

6.2 Effectiveness of Random Torque Exploration

The random torque exploration (RTE) method is validated in this study. During the data collection process, additional random torques are added to the robot to obtain widely distributed training data. To validate the effect of RTE on the training results, we prepared two training data sets with and

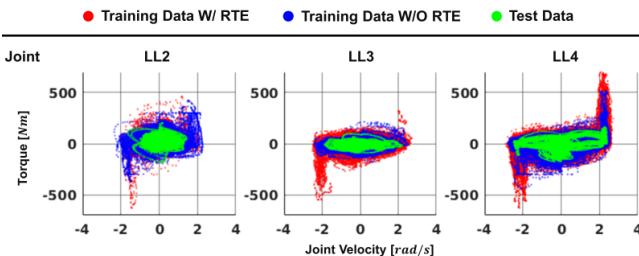


Figure 16. Joint velocity-torque plots of training data and test data. Only three joints of the left leg are displayed.

without RTE. Table 10 represents the test results according to the use of RTE. The estimation errors increase when RTE is not used for the data collection in all joints. On average, the estimation error with RTE is less than the result without RTE by 1.562 Nm. We also tested the trained network without RTE for the unseen data. Table 11 represents the RMSE of the LL3 joint for the four knee collisions during the swing phase that is the same experiment in 5.2. The results show that the RMSE increases to 13.431 Nm when the RTE is not used during data collection. The error without RTE is 6.855 Nm larger than the RMSE with RTE, which means that RTE is effective for both in-distribution and out-of-distribution tasks.

To visualize the data distribution according to the use of RTE, the joint velocity-torque graphs for three joints of the left leg are displayed in Figure 16. Because it is difficult to visualize the entire multi-dimensional input vectors, only a subset of input features (joint velocity and torque) are displayed. As shown in Figure 16, the use of RTE expands the distribution of training data, and the more distributed training data results in better estimation performance.

7 Application: Sensorless Locomotion, Collision Detection, and Collision Reaction Scenarios

We present two realistic collision handling scenarios using MOB-Net, the proposed collision detection, and the heuristic collision reaction method as shown in Figure 17 (Extension 2). Three additional scenarios can be found in Extension 2. Although implementation details of the collision reaction strategy are not addressed in this paper, these scenarios present the practical application and highlight the robust estimation performance of the proposed method for various collision cases. Note that only proprioceptive sensors of the robot (encoders and IMU) are used and FTS is not used in these scenarios for walking control and collision handling. The walking control method using the estimated external joint torque is the same as in Lim et al. (2023).

These scenarios exemplify typical and representative collision situations that are likely to happen in human-centered industrial sites. In Figure 17 (a), the humanoid robot walks forward and its right knee collides with a fixed tool cart. Then, the collision is detected in the hip-pitch joint of the right leg and the robot reacts to the collision by walking three steps along the external force direction (backward) to avoid additional collisions with an obstacle. After three steps backward, the robot entered the stop mode safely. Figure 17 (b) demonstrates a door collision scenario.

In this scenario, the robot moved forward to get through the door, but the right shoulder of the robot collided with the door frame. The proposed method can also detect such collision on the upper body and enable the robot to react to the collision by returning to the way it passed. This collision reaction prevents the robot from making further collisions with the door and falling. Note that these collisions can not be detected using FTS on the end-effectors of the humanoid.

8 Conclusion

In this paper, the limb-modularized uncertainty torque learning method, MOB-Net, is presented. By utilizing the sparsity of the dynamics of the humanoid robot as an inductive bias, an efficient and effective network architecture is designed. The combination of a model-based method (MOB) and the deep learning technique results in not only accurate estimation performance for the normal walking data (in-distribution data) but also robust estimation performance for the collision data (out-of-distribution data). Furthermore, the estimated standard deviation of the uncertainty torque from the Gaussian output feature contributes to the fast and robust collision detection performance. Extensive simulations and experiments are conducted to validate the superior estimation performance of the proposed method to the end-to-end learning method and other model-based observers. Finally, the sensorless collision handling method is implemented in the real humanoid robot. This is made possible due to MOB-Net, the core technology: accurate external joint torque estimation from MOB-Net in all joints including virtual joints, and sensitive collision detection using MOB-Net's output. We conclude that MOB-Net can enhance the safety of humanoids by implementing the whole-body collision handling method along with the balancing controller. It can also reduce the manufacturing cost of humanoids by replacing FTS with MOB-Net in mass production although FTS on the foot is required for data collection.

8.1 Limitations and Future Works

Although MOB-Net only requires encoders and IMU measurements for inference of the network, FTS on the foot is necessary to calculate the external joint torque and, finally, uncertainty torque on the legs and virtual joints in the walking data. For research groups with a few humanoid robots, it would be difficult to benefit from the effect of cost reduction due to the requirement of FTS in the data collection. Another limitation of the proposed method is that the robust estimation performance for the unseen data is not guaranteed theoretically and the estimation performance is closely related to the quality of the training data. However, this limitation is still an open problem in the field of deep learning research.

As one direction of future works, the estimated uncertainty torque and the external joint torque can be used to improve the control performance or for the compliant force control of humanoids. Similar to the work for aerial vehicles in O'Connell et al. (2022), the learned disturbances can be incorporated into the controllers and improve the control performance. As another possible direction of future work, the estimated uncertainty torque from MOB-Net can be used

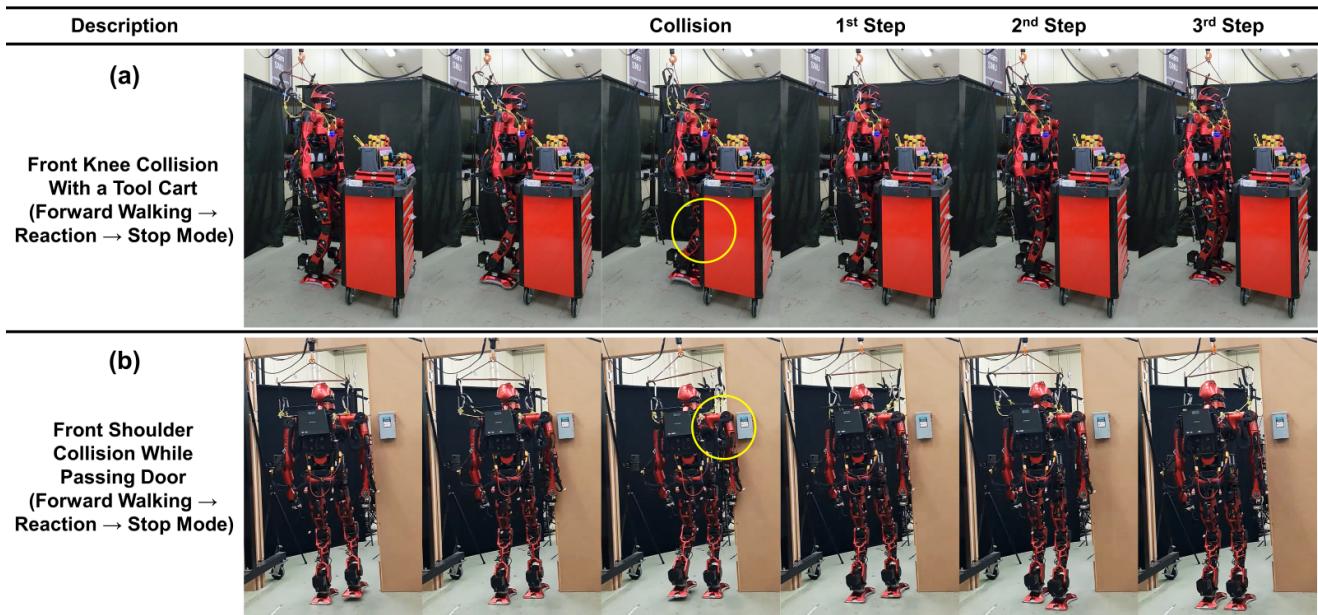


Figure 17. Collision detection and reaction scenarios of humanoids using the estimated external torque of MOB-Net (see also Extension 2). (a) The front knee of the humanoid collides with a tool cart fixed on the ground while walking forward. (b) The right shoulder of the humanoid robot collides with a door while passing the door.

to narrow the sim-to-real gap of deep reinforcement learning. As in the study of Hwangbo et al. (2019), the learned uncertainty torque from the real robot could be utilized to simulate the virtual robot similar to the real robot.

Acknowledgements

The authors would like to thank JuneWhee Ahn and Kwanwoo Lee for their gracious help in the real robot experiments.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2021R1A2C3005914).

References

- Abeykoon AHS and Chinthaka MD (2014) Position based static friction estimation for dc motors using disturbance observer. In: *7th International Conference on Information and Automation for Sustainability*. IEEE, pp. 1–6.
- Ackerman E (2023) Human in the loop: What the avatar xprize revealed about the future of telepresence robots. *IEEE Spectrum* 60(5): 38–44.
- Bai S, Kolter JZ and Koltun V (2018) An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*.
- Barasuol V, Fink G, Focchi M, Caldwell D and Semini C (2019) On the detection and localization of shin collisions and reactive actions in quadruped robots. In: *International Conference on Climbing and Walking Robots*, volume 49. p. 51.
- Benallegue M, Gergondet P, Audrerr H, Mifsud A, Morisawa M, Lamiraux F, Kheddar A and Kanehiro F (2018) Model-based external force/moment estimation for humanoid robots with no torque measurement. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 3122–3129.
- Birjandi SAB and Haddadin S (2020) Model-adaptive high-speed collision detection for serial-chain robot manipulators. *IEEE Robotics and Automation Letters* 5(4): 6544–6551.
- Bledt G, Wensing PM, Ingersoll S and Kim S (2018) Contact model fusion for event-based locomotion in unstructured terrains. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 4399–4406.
- Caldas A, Makarov M, Grossard M, Rodriguez-Ayerbe P and Dumur D (2013) Adaptive residual filtering for safe human-robot collision detection under modeling uncertainties. In: *2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. IEEE, pp. 722–727.
- Chiu JR, Sleiman JP, Mittal M, Farshidian F and Hutter M (2022) A collision-free mpc for whole-body dynamic locomotion and manipulation. In: *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 4686–4693.
- Cho C and Song J (2013) Collision detection algorithm robust to model uncertainty. *International Journal of Control, Automation and Systems* 11(4): 776–781.
- Cirillo A, Ficuciello F, Natale C, Pirozzi S and Villani L (2015) A conformable force/tactile skin for physical human–robot interaction. *IEEE Robotics and Automation Letters* 1(1): 41–48.
- Daneshmand E, Khadiv M, Grimminger F and Righetti L (2021) Variable horizon mpc with swing foot dynamics for bipedal walking control. *IEEE Robotics and Automation Letters* 6(2): 2349–2356.
- De Luca A, Albu-Schaffer A, Haddadin S and Hirzinger G (2006) Collision detection and safe reaction with the dlr-iii lightweight manipulator arm. In: *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 1623–1630.

- De Luca A and Flacco F (2012) Integrated control for phri: Collision avoidance, detection, reaction and collaboration. In: *2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*. IEEE, pp. 288–295.
- De Luca A and Mattone R (2003) Actuator failure detection and isolation using generalized momenta. In: *2003 IEEE international conference on robotics and automation (Cat. No. 03CH37422)*, volume 1. IEEE, pp. 634–639.
- De Luca A and Mattone R (2005) Sensorless robot collision detection and hybrid force/motion control. In: *Proceedings of the 2005 IEEE international conference on robotics and automation*. IEEE, pp. 999–1004.
- De Luca A, Schroder D and Thummel M (2007) An acceleration-based state observer for robot manipulators with elastic joints. In: *Proceedings 2007 IEEE international conference on robotics and automation*. IEEE, pp. 3817–3823.
- El Dine KM, Sanchez J, Corrales JA, Mezouar Y and Fauroux JC (2018) Force-torque sensor disturbance observer using deep learning. In: *International Symposium on Experimental Robotics*. Springer, pp. 364–374.
- Featherstone R (2014) Rigid body dynamics algorithms : 110–111.
- Flacco F, Paolillo A and Kheddar A (2016) Residual-based contacts estimation for humanoid robots. In: *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*. IEEE, pp. 409–415.
- Gaertner M, Bjelonic M, Farshidian F and Hutter M (2021) Collision-free mpc for legged robots in static and dynamic scenes. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 8266–8272.
- Griffin RJ, Wiedebach G, Bertrand S, Leonessa A and Pratt J (2017) Walking stabilization using step timing and location adjustment on the humanoid robot, atlas. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 667–673.
- Guo M, Zhang H, Feng C, Liu M and Huo J (2018) Manipulator residual estimation and its application in collision detection. *Industrial Robot: An International Journal*.
- Haddadin S, Albu-Schaffer A, De Luca A and Hirzinger G (2008) Collision detection and reaction: A contribution to safe physical human-robot interaction. In: *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 3356–3363.
- Haddadin S, Albu-Schäffer A, Eiberger O and Hirzinger G (2010) New insights concerning intrinsic joint elasticity for safety. In: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 2181–2187.
- Haddadin S, De Luca A and Albu-Schäffer A (2017) Robot collisions: A survey on detection, isolation, and identification. *IEEE Transactions on Robotics* 33(6): 1292–1312.
- Heo Y, Kim D, Lee W, Kim H, Park J and Chung W (2019) Collision detection for industrial collaborative robots: A deep learning approach. *IEEE Robotics and Automation Letters* 4(2): 740–746.
- Hochreiter S and Schmidhuber J (1997) Long short-term memory. *Neural computation* 9(8): 1735–1780.
- Hwangbo J, Bellicoso CD, Fankhauser P and Hutter M (2016) Probabilistic foot contact estimation by fusing information from dynamics and differential/forward kinematics. In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 3872–3878.
- Hwangbo J, Lee J, Dosovitskiy A, Bellicoso D, Tsounis V, Koltun V and Hutter M (2019) Learning agile and dynamic motor skills for legged robots. *Science Robotics* 4(26): eaau5872.
- Ito T, Ayusawa K, Yoshida E and Kheddar A (2019) Experimental study for controller-friendly contact estimation for humanoid robot. In: *2019 IEEE International Conference on Advanced Robotics and its Social Impacts (ARSO)*. IEEE, pp. 28–33.
- Jeong H, Lee I, Oh J, Lee KK and Oh JH (2019) A robust walking controller based on online optimization of ankle, hip, and stepping strategies. *IEEE Transactions on Robotics* 35(6): 1367–1386.
- Joe HM and Oh JH (2018) Balance recovery through model predictive control based on capture point dynamics for biped walking robot. *Robotics and Autonomous Systems* 105: 1–10.
- Kaneko K, Kanehiro F, Morisawa M, Yoshida E and Laumond JP (2012) Disturbance observer that estimates external force acting on humanoid robots. In: *2012 12th IEEE International Workshop on Advanced Motion Control (AMC)*. IEEE, pp. 1–6.
- Khadiv M, Herzog A, Moosavian SAA and Righetti L (2020) Walking control based on step timing adaptation. *IEEE Transactions on Robotics* 36(3): 629–643. DOI:10.1109/TRO.2020.2982584.
- Kim D, Lim D and Park J (2021) Transferable collision detection learning for collaborative manipulator using versatile modularized neural network. *IEEE Transactions on Robotics* 38(4): 2426–2445.
- Kim MJ, Lim D, Park G and Park J (2022) Humanoid balance control using centroidal angular momentum based on hierarchical quadratic programming. In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 6753–6760.
- Kim MJ, Lim D, Park G and Park J (2023) Foot stepping algorithm of humanoids with double support time adjustment based on capture point control. In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 12198–12204.
- Kircanski NM and Goldenberg AA (1997) An experimental study of nonlinear stiffness, hysteresis, and friction effects in robot joints with harmonic drives and torque sensors. *The International Journal of Robotics Research* 16(2): 214–239.
- Kobayashi T, Dean-Leon E, Guadarrama-Olvera JR, Bergner F and Cheng G (2022) Whole-body multicontact haptic human–humanoid interaction based on leader–follower switching: A robot dance of the “box step”. *Advanced Intelligent Systems* 4(2): 2100038.
- Koptev M, Figueroa N and Billard A (2021) Real-time self-collision avoidance in joint space for humanoid robots. *IEEE Robotics and Automation Letters* 6(2): 1240–1247.
- Krotkov E, Hackett D, Jackel L, Perschbacher M, Pippine J, Strauss J, Pratt G and Orlowski C (2018) The darpa robotics challenge finals: Results and perspectives. *The DARPA Robotics Challenge Finals: Humanoid Robots To The Rescue* : 1–26.
- Lee H, Lee Y and Park J (2018) Contact-consistent disturbance observer for floating-base robots. In: *International Symposium on Experimental Robotics*. Springer, pp. 475–484.
- Lee S, Kim M and Song J (2015) Sensorless collision detection for safe human-robot collaboration. In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 2392–2397.

- Lee T and Park FC (2018) A geometric algorithm for robust multibody inertial parameter identification. *IEEE Robotics and Automation Letters* 3(3): 2455–2462.
- Lim D, Kim D and Park J (2021) Momentum observer-based collision detection using lstm for model uncertainty learning. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 4516–4522.
- Lim D, Kim D and Park J (2022) Online telemanipulation framework on humanoid for both manipulation and imitation. In: *2022 19th International Conference on Ubiquitous Robots (UR)*. IEEE, pp. 8–15.
- Lim D, Kim MJ, Cha J, Kim D and Park J (2023) Proprioceptive external torque learning for floating base robot and its applications to humanoid locomotion. *arXiv preprint arXiv:2309.04138 (Accepted by 2023 IROS)*.
- Lin TY, Zhang R, Yu J and Ghaffari M (2021) Legged robot state estimation using invariant kalman filtering and learned contact events. *arXiv preprint arXiv:2106.15713*.
- Ma D, Yan S, Yin Z and Yang Y (2018) Investigation of the friction behavior of harmonic drive gears at low speed operation. In: *2018 IEEE International Conference on Mechatronics and Automation (ICMA)*. IEEE, pp. 1382–1388.
- Mesesan G, Englsberger J and Ott C (2021) Online dcm trajectory adaptation for push and stumble recovery during humanoid locomotion. In: *2021 IEEE international conference on robotics and automation (ICRA)*. IEEE, pp. 12780–12786.
- Narukawa K, Yoshiike T, Tanaka K and Kuroda M (2017) Real-time collision detection based on one class svm for safe movement of humanoid robot. In: *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*. IEEE, pp. 791–796.
- Nix DA and Weigend AS (1994) Estimating the mean and variance of the target probability distribution. In: *Proceedings of 1994 ieee international conference on neural networks (ICNN'94)*, volume 1. IEEE, pp. 55–60.
- O’Connell M, Shi G, Shi X, Azizzadenesheli K, Anandkumar A, Yue Y and Chung SJ (2022) Neural-fly enables rapid learning for agile flight in strong winds. *Science Robotics* 7(66): eabm6597.
- Park KM, Kim J, Park J and Park FC (2020) Learning-based real-time detection of robot collisions without joint torque sensors. *IEEE Robotics and Automation letters* 6(1): 103–110.
- Park KM, Park Y, Yoon S and Park FC (2021) Collision detection for robot manipulators using unsupervised anomaly detection algorithms. *IEEE/ASME Transactions on Mechatronics* 27(5): 2841–2851.
- Park Y, Chau K, Black RJ and Cutkosky MR (2007) Force sensing robot fingers using embedded fiber bragg grating sensors and shape deposition manufacturing. In: *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE, pp. 1510–1516.
- Piperakis S, Maravgakis M, Kanoulas D and Trahanias P (2022) Robust contact state estimation in humanoid walking gaits. In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 6732–6738.
- Quiroga CWP, Abba G, Antoine JF, Raharjaona T and Garrec P (2021) Load-dependent friction laws of three models of harmonic drive gearboxes identified by using a force transfer diagram. In: *12th International Conference on Mechanical and Aerospace Engineering*.
- Rotella N, Bloesch M, Righetti L and Schaal S (2014) State estimation for a humanoid robot. In: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 952–958.
- Ruderman M and Iwasaki M (2015) Sensorless torsion control of elastic-joint robots with hysteresis and friction. *IEEE Transactions on Industrial Electronics* 63(3): 1889–1899.
- Shan S and Pham QC (2023) Fine robotic manipulation without force/torque sensor. *arXiv preprint arXiv:2301.13413*.
- Todorov E, Erez T and Tassa Y (2012) Mujoco: A physics engine for model-based control. In: *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, pp. 5026–5033.
- Tran N, Wu JY, Deguet A and Kazanzides P (2020) A deep learning approach to intrinsic force sensing on the da vinci surgical robot. In: *2020 Fourth IEEE International Conference on Robotic Computing (IRC)*. IEEE, pp. 25–32.
- Van Dam J, Tulbure A, Minniti MV, Abi-Farraj F and Hutter M (2022) Collision detection and identification for a legged manipulator. In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 13602–13609.
- Vorndamme J, Schappler M and Haddadin S (2017) Collision detection, isolation and identification for humanoids. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 4754–4761.
- Xiao J, Zhang Q, Hong Y, Wang G and Zeng F (2018) Collision detection algorithm for collaborative robots considering joint friction. *International Journal of Advanced Robotic Systems* 15(4): 1729881418788992.
- Yilmaz N, Wu JY, Kazanzides P and Tumerdem U (2020) Neural network based inverse dynamics identification and external force estimation on the da vinci research kit. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 1387–1393.

Appendix

A. Index to multimedia extensions

Table 12

Extension	Media Type	Description
1	Video	Demonstrations of training data procedure including random walking data, random motion data, and random torque exploration.
2	Video	Demonstrations of five different collision detection and reaction scenarios of a humanoid robot, TOCABI.