Momentum Observer-Based Collision Detection Using LSTM for Model Uncertainty Learning

Daegyu Lim*,1, Donghyeon Kim*,1 and Jaeheung Park1,2

***Abstract*— As robots begin to collaborate with people in real life, their applicability and practicality are continuously increasing. To reliably employ robots nearby, safety needs to be rigorously ensured. In addition to collision prevention algorithms, studies are being actively conducted on collision handling methods. Momentum Observer (MOB) was developed to estimate disturbance torque without using joint acceleration. However, the estimated disturbance from MOB contains not only the applied external torque but also model uncertainty such as friction and modeling error due to imprecise system identification. Our proposed method handles this problem by learning the model uncertainty with Long Short-Term Memory (LSTM) and thereby estimates the purely applied external torque with only proprioceptive sensors. The proposed method can be applied even when the information on the robot model is not available. The experiments using a real robot show that the external torque can be estimated and collisions can be detected accordingly even in a limited situation where a precise dynamics model and friction model are not available.**

I. Introduction

Over the past few decades, studies have been actively conducted on physical human-robot interaction (pHRI) in line with efforts to expand the use of robots in real life. For robots to be able to physically interact with humans, one of the most important issues is safety. Unlike industrial robots in a factory, humans will share the social robot's workspace and there could be the unexpected collisions between robots and people. This makes it important to find safety measures to prevent or cope with unexpected collision.

One way to ensure the safety of humans in the robot workspace is to do so in advance during the pre-collision phase. In this phase, the robot needs information about the nearby environment to plan a collision-free trajectory. However, this procedure requires additional sensors such as vision sensors and real-time computation for recognition. Moreover, any abrupt changes in the environment, such as human motion, make such prevention unreliable. Therefore, to ensure safety, in addition to a collision prevention algorithm, collision handling procedures should be prepared. In [1], Haddadin et al. defined the collision event pipeline to ensure safety during the collision phase; detection, isolation, identification, classification, reaction. Among these steps, this paper will mainly deal with the collision detection step, which determines whether the collision occurred, and the collision identification step, which estimates the applied external force/torque.

Other than using direct skin sensors[2], [3], an indirect way to detect unexpected robot collisions is to estimate the disturbance using a dynamics model of the robot. If the joint torque is available from a joint torque sensor (JTS), external torque can be obtained by subtracting the dynamics torque occurred by its motion from the measured joint torque. However, joint angular acceleration must be measured or estimated to calculate the dynamics torque. Since joint angular acceleration attained from the numerical differentiation of the joint angle is too noisy to use, in [4], the IMU signal is fused for each joint to estimate joint velocity and acceleration, and then it is used to estimate external torque.

To avoid using the joint acceleration signal, a velocity observer [5], energy observer [6], and MOB [7], [8], [9] using joint velocity and the identified system model were introduced. MOB-based methods which do not use JTS [8], [9] have advantages, in that the low-pass-filtered disturbance can be estimated using only proprioceptive sensors, such as an encoder and motor current sensor. However, the estimated disturbance includes various model uncertainty factors, such as modeling error due to imprecise system identification, friction, the elasticity of robot joints[10], and control delay [11] as well as the applied external torque. Among these uncertainty factors, [8] and [9] used a friction model to offset the effect of friction torque, but modeling the friction is difficult and the other uncertainty effects remain unstructured. As a result, it is difficult to differentiate between the pure externally applied torque and unstructured model uncertainties.

To take this model uncertainty into account, a threshold which is higher than the maximum model uncertainty can be used. However, the high threshold decreases the sensitivity of collision detection. Thus, time-varying dynamic threshold methods were suggested which can estimate threshold parameters online recursively by the least square method or gradient correction and determine the threshold accordingly [12], [13]. However, both methods require a joint acceleration signal.

To differentiate modeling error torque and external torque, [14], [15] introduced a high pass filter and bandwidth filter to filter out modeling error torque, assuming that the modeling error has slow dynamics, thus the modeling error torque is limited to low frequency. However, setting a cut-off frequency is heuristic, and collisions with slowly

*Both authors contributed equally.

[1]Daegyu Lim, Donghyeon Kim and Jaeheung Park are with the Department of Transdisciplinary Studies, Seoul National University, Seoul, KS013, Republic of Korea kdh0429, dgyo3784, park73@snu.ac.kr

[2]Jaeheung Park is also with the Advanced Institutes of Convergence Technology, Suwon, KS002 Republic of Korea.

increasing external forces would not be detected. A novel model-adaptive algorithm is also suggested in [16] using a regressor-based model adaptive observer, which compares the estimated joint torques with measured torques. This method shows high sensitivity but needs IMU to estimate joint acceleration and excludes the effect of joint friction by using JTS.

On the other side, recent works that use a deep learning approach for collision detection are superior when handling the above-mentioned model uncertainty. Data-driven deep learning approaches can detect collisions either by learning the end-to-end relationship of the collision [17] or by estimating residual [18], [19], and both approaches consider model uncertainty by itself. However, implementing an end-to-end collision detection algorithm requires collecting collision data, which is dangerous and expensive, and once trained, the sensitivity of the trained network cannot be adjusted. In contrast, the residual-based algorithms only require free-motion data which is relatively easy to collect.

In this paper, a novel MOB-based collision detection algorithm using LSTM is introduced. The LSTM is adopted to learn model uncertainty during free-motion and the external torque is estimated from the residual between the MOB and LSTM, as shown in Fig 1. The contribution of the proposed algorithm is as follows. First, it can detect collisions by estimating external torques when the modeling error and friction exist. The LSTM can handle various types of modeling error, even when there is no model on the dynamics of the robot at all. Second, the proposed method does not require any additional expensive sensors, such as JTS, 6-axis force/torque sensor, and IMU to estimate the external torque. Lastly, the proposed method is validated with a real robot showing that the method can detect collisions and estimate external torque under various types of modeling error and friction. Since the proposed method neither requires precise system identification of dynamics parameters and friction model nor any additional exteroceptive sensors, it can be generally applied to any robots with proprioceptive sensors.

The remainder of this paper is organized as follows. In Section II, the mathematical derivation to estimate disturbance torque is formulated and the momentum observer is explained. In Section III, the proposing algorithm where LSTM is used to learn model uncertainty is presented. In Section IV, it is validated that our proposed method can estimate external torque and detect collisions properly in the presence of modeling error and friction, using a 2 DOF test platform robot. Lastly, this paper summaries the proposed method and results, and suggests future work in Section V.

II. Disturbance Torque Estimation

A. Problem Formulation

The fundamental rigid body dynamics of a n degrees of freedom (DOF) robot can be formulated as

$$M(q)\ddot{q}+C(q,\dot{q})\dot{q}+g(q)=\tau_m-\tau_f+\tau_{ext}, \tag{1}$$

where $M(q)\in\mathbb{R}^{n\times n}$, $C(q,\dot{q})\in\mathbb{R}^n$, $g(q)\in\mathbb{R}^n$ are the inertia matrix, the centripetal and Coriolis matrix, the gravitational

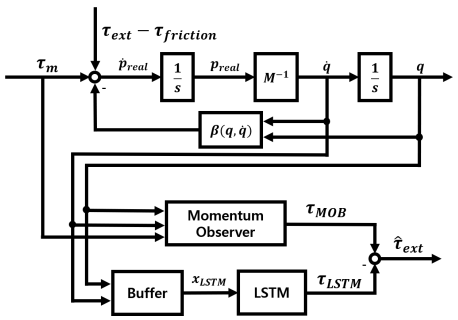


Fig. 1. Block diagram of the proposed framework. LSTM learns model uncertainty and the external torque is estimated from the residual between MOB and LSTM.

torque, $q,\dot{q},\ddot{q}\in\mathbb{R}^n$ are the joint position, the angular velocity, the angular acceleration, $\tau_m\in\mathbb{R}^n$ is the motor torque, $\tau_f\in\mathbb{R}^n$ is the friction torque and $\tau_{ext}\in\mathbb{R}^n$ is the external torque, respectively. The dynamics parameters such as $M(q)$, $C(q,\dot{q})$, $g(q)$ should be identified, but modeling error is inevitable. Therefore, the estimated rigid body dynamics of the robot can be represented as

$$\hat{M}(q)\ddot{q}+\hat{C}(q,\dot{q})\dot{q}+\hat{g}(q)=\tau_m-\tau_f-\tau_e+\tau_{ext}, \tag{2}$$

where hat ($\hat{}$) indicates estimated values and $\tau_e=M(q)\ddot{q}+C(q,\dot{q})\dot{q}+g(q)-(\hat{M}(q)\ddot{q}+\hat{C}(q,\dot{q})\dot{q}+\hat{g}(q))$ is the modeling error torque.

The model uncertainty torque τ_{uncrt} and disturbance torque τ_{dist} are defined as follows.

$$\tau_u := -\tau_f-\tau_e\in\mathbb{R}^n, \tag{3}$$

$$\tau_{dist} := -\tau_f-\tau_e+\tau_{ext} \tag{4}$$

$$= \tau_{uncrt}+\tau_{ext}\in\mathbb{R}^n. \tag{5}$$

Then, the disturbance torque can be calculated directly from Eq.(2) and the definition (4) as below.

$$\tau_{dist}=\hat{M}(q)\ddot{q}+\hat{C}(q,\dot{q})\dot{q}+\hat{g}(q)-\tau_m. \tag{6}$$

B. Momentum Observer

To avoid using noisy $\ddot{q}$ when calculating τ_{dist}, a classical disturbance observer using the generalized momentum p of the robot was introduced in [7].

$$p=\hat{M}(q)\dot{q}. \tag{7}$$

Its derivative combined with Eq.(2) is

$$\dot{p}=\tau_m+\tau_{dist}+\hat{C}^T(q,\dot{q})\dot{q}-\hat{g}(q) \tag{8}$$

$$=\tau_m+\tau_{dist}-\hat{\beta}(q,\dot{q}), \tag{9}$$

where $\hat{\beta}(q,\dot{q})=-\hat{C}^T(q,\dot{q})\dot{q}+\hat{g}(q)$ is defined for the simplification.

To estimate disturbance torque, a residual vector $r\in\mathbb{R}^n$ and its dynamics are defined as below.

$$\dot{r}=K_0(\dot{p}-\dot{\hat{p}}) \tag{10}$$

$$\dot{\hat{p}}=\tau_m-\hat{\beta}(q,\dot{q})+r, \tag{11}$$

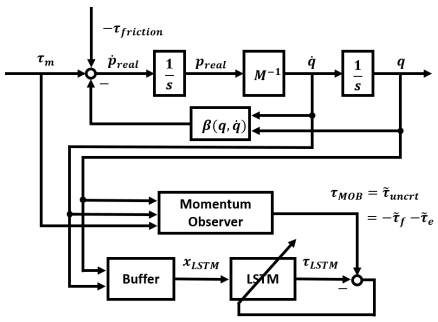


Fig. 2. Block diagram of the proposed framework during training under free-motion. LSTM is trained to learn the low-pass filtered model uncertainty $\tilde{\tau}_{uncrt}$ with time-series input x_{LSTM}.

where $K_0 = diag\{k_{0,i}\} > 0$ is the positive diagonal gain matrix and $\dot{\hat{p}}$ is the derivative of the estimated momentum.

Integrating (10) results in

$$r = K_0 \left\{ p(t) - p(0) - \int_0^t (\tau_m - \hat{\beta}(q,\dot{q}) + r)dt \right\}. \quad (12)$$

Substituting (9) and (11) to (10) derives the relation between r and τ_{dist} as

$$\dot{r} = K_0(\tau_{dist} - r). \quad (13)$$

Laplace transforming of the Eq.(13) in each joint leads to

$$\frac{r_i(s)}{\tau_{dist,i}(s)} = \frac{k_{0,i}}{s + k_{0,i}} \quad (i = 1,2,...,n), \quad (14)$$

which means that the MOB residual vector r is the first order low-pass filtered disturbance torque τ_{dist}.

Consequently, the output of the MOB ($\tau_{MOB} = r$) contains not only external torque τ_{ext} but also torques derived from model uncertainty τ_{uncrt}.

III. Model Uncertainty Learning LSTM

A. Uncertainty Learning Using LSTM Under Free-Motion

When executing free-motion, there is no external torque τ_{ext}. Therefore, the output value of the MOB is low-pass filtered model uncertainty torque $\tilde{\tau}_{uncrt}$ which contains low-pass filtered friction $\tilde{\tau}_f$ and modeling error $\tilde{\tau}_e$, as in Fig. 2 and Eq.(3). Here, tilde ($\tilde{\cdot}$) denotes low-pass filtered value. LSTM learns these model uncertainties with the time-series data input. Although some may argue that using a simple fully connected network that only uses current time data $x(k)$ as an input to infer current model uncertainty is enough, time-series data is used with LSTM since it is well-known that friction shows hysteresis behavior, such that the friction is dependent on the history of the states. Additionally, because the output of the MOB is the low-pass-filtered value of τ_{dist} as shown in Eq.(14), there is a time delay between current disturbance torque and MOB output. Thus, using time-series data as an input is appropriate for inferring MOB output τ_{MOB}.

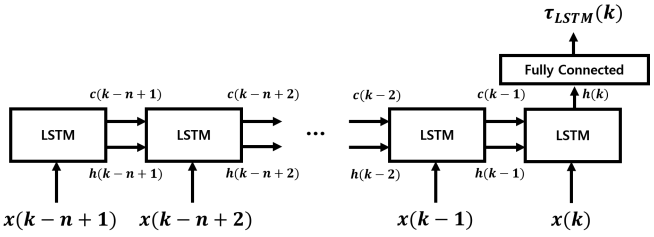


Fig. 3. Structure of Many-to-one LSTM. Time-series proprioceptive data are fed into LSTM and the LSTM estimates a torque due to model uncertainty.

B. Input Selection and LSTM Structure

The utilized many-to-one LSTM has 1 hidden layer and 20 hidden units for each hidden state h and cell state c, as in Fig.3. The LSTM takes time-series data as input and learns the output value of the MOB under free-motion, which is low-pass filtered model uncertainty $\tilde{\tau}_{uncrt}$. The low-pass filtered model uncertainty torque consists of low-pass filtered friction $\tilde{\tau}_{friction}$ and modeling error $\tilde{\tau}_e$. When the torque from friction τ_f is assumed to be a friction under the sliding region, the low-pass filtered friction torque $\tilde{\tau}_{friction}$ can be expressed with a function of $\dot{q}$ and its histories. Likewise, since the torque due to modeling error $\tau_e = M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) - (\hat{M}(q)\ddot{q} + \hat{C}(q,\dot{q})\dot{q} + \hat{g}(q))$ is a function of $q, \dot{q}, \ddot{q}$, the low-pass filtered modeling error torque $\tilde{\tau}_e$ can be expressed with time-series data of q, $\dot{q}$, $\ddot{q}$. However, as mentioned in Section II-B, since $\ddot{q}$ may not be measured, the joint velocity of the previous time step $\dot{q}(k-1)$ along with current joint velocity $\dot{q}(k)$ can be used where k is a discretized time variable. Thus, $\tilde{\tau}_{uncrt}$ can be expressed using time-series input variable x_{uncrt} as follows.

$$\tilde{\tau}_{uncrt} = -\tilde{\tau}_e - \tilde{\tau}_f \quad (15)$$

$$= f(x_{uncrt}) \quad (16)$$

where f is a true mapping from input to low-pass filtered model uncertainty and x_{uncrt} is a time-series input as follows.

$$x_{uncrt} = [x(k-n+1),\ \ldots\ ,\ x(k-1),\ x(k)], \quad (17)$$

where one time step data $x(t)$ is organized as

$$x(k) = [q(k),\ \dot{q}(k),\ \dot{q}(k-1)]. \quad (18)$$

Thus, the LSTM network f_θ is trained to learn the true mapping function f and the input of the LSTM x_{LSTM} is chosen to be same as x_{uncrt} where a time interval between the data is $dt = 0.01s$ and the size of data buffer sequence is $n = 100$.

C. Extreme Modeling Error Case

The modeling error torque τ_e can be extreme when there is not model information at all, thereby assuming that $\hat{M}(q) = 0_{n\times n}$, $\hat{C}(q,\dot{q}) = 0_{n\times n}$, $\hat{g}(q) = 0_{n\times 1}$. In this case, Eqs.(10) and (11) can be modified as

$$\dot{r} = -K_0\dot{\hat{p}} \quad (19)$$

$$\dot{\hat{p}} = \tau_m + r, \quad (20)$$

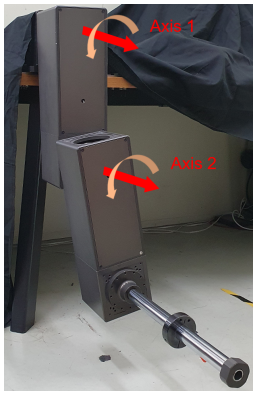


Fig. 4. Torque-controlled multi-functional test flatform (MOCCA). MOCCA can be assembled with various combinations of kinematic structures, the number of joints, joint stiffness, and so on. In the presented experiments, MOCCA is set vertically with two joints.

which further modifies Eq.(14) as

$$\frac{r_i(s)}{\tau_{m,i}(s)} = \frac{k_{0,i}}{s+k_{0,i}} \quad (i = 1,2,...,n). \tag{21}$$

Therefore, when there is no model at all, the output of MOB is equivalent to the low-pass-filtered value of motor torque. Thus, in this extreme case, the LSTM learns the low-pass-filtered value of motor torque.

D. External Torque Estimation and Collision Detection

After the LSTM network f_θ is trained updating the parameters θ under free-motion data, LSTM predicts the low-pass-filtered model uncertainty torque under the current state x_{LSTM} as below.

$$\tau_{LSTM} = f_\theta(x_{LSTM}). \tag{22}$$

Thus, as shown in Fig. 1, the external torque can be estimated by subtracting LSTM prediction τ_{LSTM} from the MOB output τ_{MOB}. When the estimated external torque $\hat{\tau}_{ext}$ exceeds the threshold which is determined by the maximum regression error in the validation set, the algorithm infers that a collision occurred.

IV. Experimental Validation

A. Experiment Setup

The proposed method was validated using a torque-controlled multi-functional test platform MOCCA (MOdular Configuration Changeable Actuator test platform) [20]. The MOCCA was designed to change various physical variables such as kinematic structure, motor drive shaft inertia, joint stiffness, and so on. Among the various kinematic structures which MOCCA can be assembled to, a 2-DOF vertical structure was adopted as shown in Fig. 4 to clearly show the effect of modeling error, which is evident in gravitational torque $g(q)$ even in a static pose. The length of each link is $0.3m$ and the total mass is $18.111kg$. A joint torque sensor is embedded but only used to compare the applied external torque with the proposed method. The robot was controlled using joint Proportional–Derivative (PD) control in 1000 hz.

TABLE I

LSTM regression error according to the type of model error in the free-motion test set.

Type of model error	joint #	L1 norm (Nm)	L2 norm (Nm)	Variance (N^2m^2)
Overall Region				
W/O modeling error	1	0.402	0.661	0.435
	2	0.262	0.369	0.136
Mass error	1	0.401	0.687	0.468
	2	0.256	0.365	0.132
COM error	1	0.392	0.680	0.458
	2	0.253	0.355	0.126
Mass + COM error	1	0.415	0.681	0.464
	2	0.258	0.368	0.134
W/O model	1	0.409	0.683	0.466
	2	0.259	0.376	0.141

The experiment was designed to show that the proposed method can be applied to various types of modeling error, even the case when there is no model. Among the dynamics parameters (mass, inertia, Center of Mass (COM)), inertia showed a minor effect on modeling error torque τ_e. Thus, mass (m_1, m_2) and the position of COM (p_{c1}, p_{c2}) are perturbed from our best-estimated parameters ($m_1 = 8.191kg$, $m_2 = 9.920kg$, $p_{c1} = [0.12079, 0.0, 0.0]m$, $p_{c2} = [0.16078, 0.0, 0.0]m$) by design as follows.

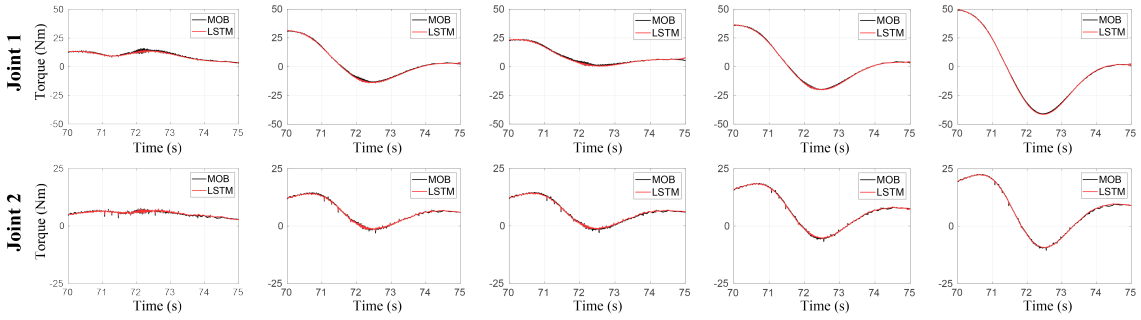
- Without modeling error: Our best-estimated parameters
- Mass error: Mass of each link is reduced to 50%
- COM error: Position of COM is set 50% closer to the axis in each direction
- Mass+COM error: Mass and position are perturbed simultaneously
- Without model: Mass of the robot is set to $0kg$.

B. Data Collection and Training

To collect the data to train LSTM, each MOB using the perturbed dynamic parameters described in the previous section was activated with $K_0 = diag\{20, 20\}$ simultaneously to estimate corresponding modeling uncertainty during free-motion. Free-motion consists of quintic-splined trajectories from the initial joint position to the target joint position which is randomly chosen within a joint position limit of [-180, 180]° and maximum average joint speed of [20, 50]°/s. LSTM input $q(k)$, $\dot{q}(k)$, $\dot{q}(k-1)$ and the MOB values τ_{MOB} for each perturbed model are collected with 100hz for 90 minutes of free-motion. 70% of the data are used for training, 15% for validation, and 15% for testing. A batch size of 1000, the number of epochs 500, Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and, learning rate $lr = 0.001$ are used for optimization. Here, note that one LSTM is trained for each perturbed model.

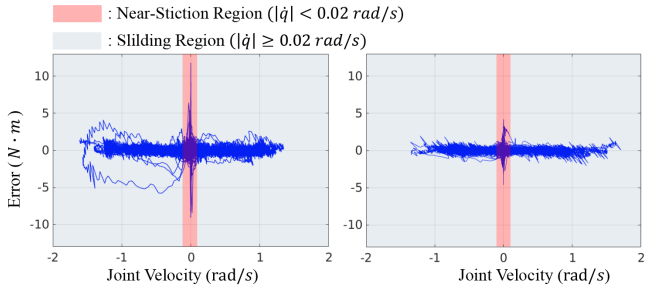
C. Validation of Model Uncertainty Learning

The trained LSTM network was compared with the MOB under the free-motion test data set. Since the LSTM is trained to learn the model uncertainty and the output of the MOB is also model uncertainty under free-motion as mentioned in Section III-A, the output of the LSTM should be similar to the value of MOB when properly trained.



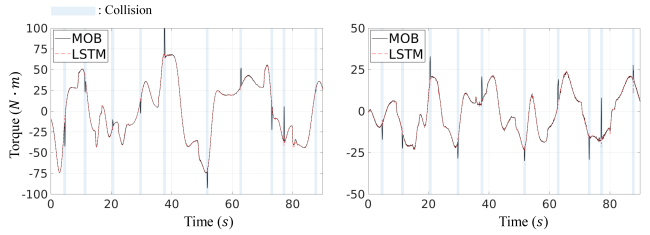
(a) Without Modeling Error (b) Mass Error (C) COM Error (d) Mass + COM Error (e) Without Model

Fig. 5. Estimated disturbance torque by MOB(blue) and LSTM(red) in free-motion test set with various model uncertainty. The upper figures are the results of the first joint and the lower figures are the results of the second joint. (a): Without modeling error, the model uncertainty is a friction torque. (b) (e): With error on mass or COM position, the model uncertainty contains friction torque and modeling error due to inaccurate model information. Note that as the modeling error increases from (a) to (e), the scale of the model uncertainty also increases.



(a) Joint 1 (b) Joint 2

Fig. 6. Error of the estimated external torque using the proposed method over joint speed in free-motion test data set. In the near-stiction region (pink area), the robot is under the static fiction and LSTM could not predict the static fiction resulting in large error. In the sliding region (light blue area), the robot is under the friction in sliding mode and the errors in this region was smaller than the errors in the near-stiction region.



(a) Joint 1 (b) Joint 2

Fig. 7. Outputs of the MOB and LSTM in case of when there is no model at all. 10 collisions occurred in light blue areas where the disturbance torque estimated from MOB (black line) peaked, while the model uncertainty torque estimated from the LSTM (dotted red line) did not change rapidly.

The results according to the type of perturbed model are shown in Fig. 5. First, in Fig. 5 (a), the results of the LSTM and MOB with no modeling error are represented. In this case, it is assumed that the modeling error is minor and the output of the MOB is mainly friction torque. From Fig. 5 (a) and TABLE I, it is observed that the LSTM output is similar to the MOB output in the free-motion with an L1 norm error of 0.402 Nm and 0.262 Nm for each joint in the test set. This demonstrates that the LSTM properly learns the friction torque in free-motion.

Then, the LSTM was trained with perturbed versions of MOB (Mass Error, COM Error, Mass+COM Error, Without Model). Fig. 5 (b)-(e) shows that the LSTM can also estimate these uncertainty-containing MOB outputs successfully. Additionally, as shown in Table I, test results across the perturbed models show small regression errors in the test set.

However, when the error between the MOB and the LSTM is plotted over joint velocity, the error is significantly high in the near-stiction region ($|\dot{q}|$ <0.02 rad/s) as shown in Fig. 6. In all cases, the magnitude and variance of the error are higher in the near-stiction region than in the sliding region, which implies that the LSTM can not properly find the pattern of the uncertainty from the input when the robot rarely moves. Representatively, when there is no modeling error, root-mean-squared error (RMSE) of the first joint in the sliding region is 0.418Nm, whereas RMSE in the near-stiction region is 1.344Nm. The deterioration of the performance in the near-stiction region is presumed to be the effect of the static friction. It was assumed that the estimated friction torque is the friction in the sliding region in Section III-B, however, static friction cannot be estimated from the selected input, time series of q and $\dot{q}$. More specifically, as the motor torque gradually increases, the static friction increases until the motor torque exceeds the maximum stiction force without changes in states q and $\dot{q}$. Therefore, with the input of q and $\dot{q}$, the static friction could not be determined uniquely. Consequently, the data under static friction deteriorate training in the near-stiction region. For practical and efficient collision detection, the threshold can be set differently in the near-stiction region and the sliding region.

D. External Torque Estimation and Collision Detection

In this section, it is demonstrated that the pure external torques can be estimated and collisions can be successfully detected using the proposed method.

Ten collisions occurred while the robot was randomly moving, and the outputs of the MOB and LSTM, when there is no model at all, are observed. As shown in Fig. 7, the MOB output rapidly increased when the human pushed the robot, although there was no significant change in LSTM because

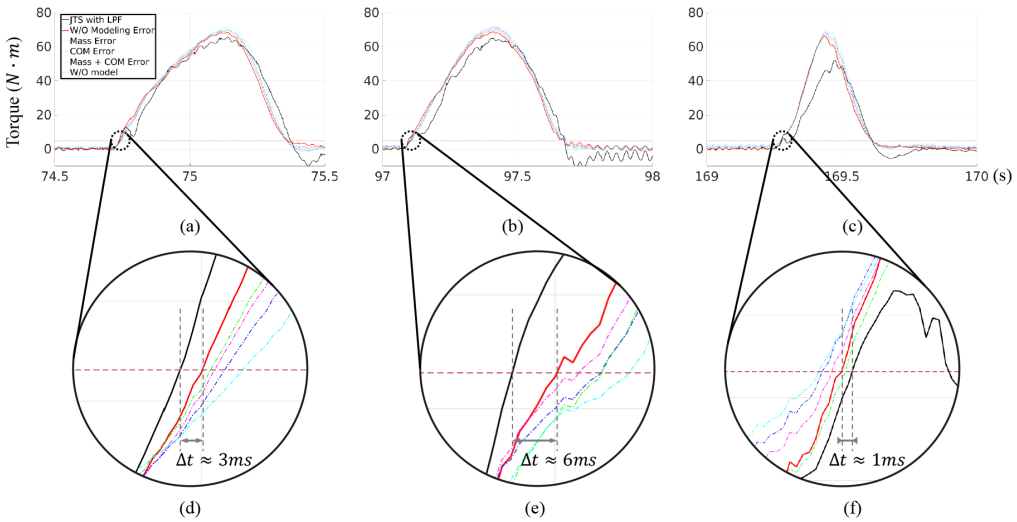


Fig. 8. External torque measured with JTS (black) and estimated with the proposed method (red, pink, green, blue and light blue) and collision detection delay at three collision cases. The proposed method is applied to the cases when there is no modeling error (red), mass error (pink), COM error (green), mass+COM error (blue), and no information on the model (light blue) respectively. (a), (b), (c): Estimated external torque using the proposed method according to the perturbed model (colored line) and the measured external torque using JTS (black). (d), (e), (f): Enlarged figures at the moment of the collision.

it only learns model uncertainty torque. This difference between MOB and LSTM is the estimated external torque occurred from the collision, and all the estimated external torque in 10 collisions exceeded the threshold obtained from the validation data set. Although the results of other types of modeling errors are not shown, all 10 collisions were also successfully detected in all cases.

Then, in the first joint, the estimated external torque ($\hat{\tau}_{ext} = \tau_{MOB} - \tau_{LSTM}$) obtained from the proposed method was compared with the measurement from JTS in Fig. 8. The joint torque produced by the motion $\tau_{dyn} = M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q)$ is directly subtracted from the JTS measurement to calculate the reference external torque and low-pass-filter with $\omega_c = 20 rad/s$ is applied. In Fig. 8 (a), (b), and (c), the result verifies that the external torques estimated by the proposed algorithm across various types of modeling error are similar to the external torque measured from the torque sensor with respect to magnitude and responsiveness. However, occasionally there is a gap between the estimated external torque and the measured external torque, as shown in Fig. 8 (c). This difference could have occurred due to the static friction, measurement error, or a training error.

The collision detection delay in the first joint was calculated in the case without modeling error using the proposed method. Here, the threshold is set to the maximum regression error of the LSTM in the sliding region ($\tau_{thr} = 4.62Nm$), and the baseline of the detection time is calculated from the JTS with the same threshold. Three cases of collision are represented in Fig. 8. In Fig. 8 (d), (e), and (f), the relative detection delays of the LSTM from the baseline were $3ms$, $6ms$, and $-1ms$, respectively. It can be seen that the JTS detects the collision faster than the proposed method in the usual cases, as shown in Fig. 8 (d), (e), but the sensitivity is reversed in some cases, as shown in Fig. 8 (f). The results of other types of modeling error shows a similar sensitivity and tendency to the estimated external torque, as shown in Fig. 8 (d), (e), and (f).

V. Conclusions

In this paper, a MOB-based collision detection method considering model uncertainty with LSTM is proposed. In the proposed method, the model uncertainty, which contains modeling error and friction, is learned with the LSTM under free-motion. When properly trained, the applied external torque can be identified by subtracting the output of the LSTM from the disturbance torque estimated from MOB. Using the proposed method, the external torque can be estimated and collisions can be detected accordingly using only proprioceptive sensors even in a limited situation where a precise dynamics model and friction model are not available.

The method's ability to detect a collision and to estimate external torque were shown by experiments using a 2-DOF torque-controlled test platform. The experiments demonstrated that the proposed algorithm can handle friction and various types of modeling error which ranges from an error of mass and COM position to the extreme case where no information on the model is available.

However, since the friction was assumed to be in the sliding region, the effect of static friction could not be reflected and consequently deteriorated the performance. An additional input variable or network structure which can model the static friction can be considered to improve performance in future work. Also, human-robot interaction using the estimated external torque can be explored to extend the applicability of the method.

VI. Acknowledgement

We would like to express our special thanks to Jaehoon Sim for his assistance with experiment setup and maintenance.

REFERENCES

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

[2] A. Cirillo, F. Ficuciello, C. Natale, S. Pirozzi, and L. Villani, "A conformable force/tactile skin for physical human–robot interaction," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 41–48, 2015.

[3] Y. Park, K. Chau, R. J. Black, and M. R. Cutkosky, "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, 2007, pp. 1510–1516.

[4] S. A. B. Birjandi, J. Kühn, and S. Haddadin, "Observer-extended direct method for collision monitoring in robot manipulators using proprioception and imu sensing," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 954–961, 2020.

[5] S. Haddadin, *Towards safe robots: approaching Asimov's 1st law*. Springer, 2013, vol. 90.

[6] A. De Luca, A. Albu-Schaffer, S. Haddadin, and G. Hirzinger, "Collision detection and safe reaction with the dlr-iii lightweight manipulator arm," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2006, pp. 1623–1630.

[7] A. De Luca and R. Mattone, "Actuator failure detection and isolation using generalized momenta," in *2003 IEEE international conference on robotics and automation (Cat. No. 03CH37422)*, vol. 1. IEEE, 2003, pp. 634–639.

[8] S. Lee, M. Kim, and J. Song, "Sensorless collision detection for safe human-robot collaboration," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 2392–2397.

[9] J. Xiao, Q. Zhang, Y. Hong, G. Wang, and F. Zeng, "Collision detection algorithm for collaborative robots considering joint friction," *International Journal of Advanced Robotic Systems*, vol. 15, no. 4, p. 1729881418788992, 2018.

[10] A. De Luca, "Feedforward/feedback laws for the control of flexible robots," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 1. IEEE, 2000, pp. 233–240.

[11] J. Jung, S. Hwang, Y. Lee, J. Sim, and J. Park, "Analysis of position tracking in torque control of humanoid robots considering joint elasticity and time delay," in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*. IEEE, 2017, pp. 515–521.

[12] A. Caldas, M. Makarov, M. Grossard, P. Rodriguez-Ayerbe, and D. Dumur, "Adaptive residual filtering for safe human-robot collision detection under modeling uncertainties," in *2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. IEEE, 2013, pp. 722–727.

[13] M. Guo, H. Zhang, C. Feng, M. Liu, and J. Huo, "Manipulator residual estimation and its application in collision detection," *Industrial Robot: An International Journal*, 2018.

[14] S. Haddadin, A. Albu-Schaffer, A. De Luca, and G. Hirzinger, "Collision detection and reaction: A contribution to safe physical human-robot interaction," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2008, pp. 3356–3363.

[15] C. Cho and J. Song, "Collision detection algorithm robust to model uncertainty," *International Journal of Control, Automation and Systems*, vol. 11, no. 4, pp. 776–781, 2013.

[16] S. A. B. Birjandi and S. Haddadin, "Model-adaptive high-speed collision detection for serial-chain robot manipulators," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6544–6551, 2020.

[17] Y. Heo, D. Kim, W. Lee, H. Kim, J. Park, and W. Chung, "Collision detection for industrial collaborative robots: A deep learning approach," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 740–746, 2019.

[18] M. H. Terra and R. Tinós, "Fault detection and isolation in robotic manipulators via neural networks: A comparison among three architectures for residual analysis," *Journal of Robotic Systems*, vol. 18, no. 7, pp. 357–374, 2001.

[19] C. Cho, J. Hong, and H. Kim, "Neural network based adaptive actuator fault detection algorithm for robot manipulators," *Journal of Intelligent & Robotic Systems*, vol. 95, no. 1, pp. 137–147, 2019.

[20] J. Ko, J. Sim, J. Jung, S. Hwang, and J. Park, "Multi-functional test platform for torque-controlled robot: Design and performance verification," *Journal of Institute of Control, Robotics and Systems*, vol. 25, no. 12, pp. 1093–1100, 2019.