

Article

A Novel Virtual Torque Sensor for Rescue Robots with Harmonic Drives

Kerui Xia ^{1,2}, Liang Ding ^{1,*}, Guangjun Liu ², Haibo Gao ¹ and Zongquan Deng ¹

¹ State Key Laboratory of Robotics and System, Harbin Institute of Technology, Harbin 150001, China; x_kerui@126.com (K.X.); keruixia@gmail.com (H.G.); denzq@hit.edu.cn (Z.D.)

² Department of Aerospace Engineering, Ryerson University, Toronto, ON M5B 2K3, Canada; kavinn@ryerson.ca

* Correspondence: 12B908067@hit.edu.cn

Abstract: In this paper, a method is developed for presenting a novel virtual torque sensor based on precise model and position measurements avoids the need of traditional strain gauges and amplifiers. More specifically, the harmonic drive compliance model and the Gaussian process regression (GPR) technique are used together to achieve virtual torque sensor measurement. While the harmonic drive compliance model provides the analytic part, the Gaussian process regression method is used to reconstruct the unmolded part based on motor-side and link-side joint angles as well as motor current. After an automatic offline calibration, the method allows for a lean online implementation. The virtual torque sensor measurement is compared with measurements of a commercial torque sensor, and the results have attested the effectiveness of the proposed method.

Keywords: robot joint; virtual torque sensor; Gaussian process regression; harmonic drive compliance model

I. INTRODUCTION

Joint torque feedback (JTF) is an effective technique for improving the performance of robot motion control [1]–[3], as well as force and impedance control [4]–[7]. To implement JTF control schemes, the measurement of robot joint torque is essential. Measurement of robot joint torque with a multi-axis wrist force/torque (F/T) sensor is often inaccurate due to computation delays and model errors [8]. While joint torque measurement can readily be obtained from dedicated joint torque sensors, it is often difficult to mount such sensors as additional space and mechanical modification are required at the joints [12]. Furthermore, joint torque sensors are usually strain gauge based and suffer temperature effect and amplifier drifts [13], [14], and [15]. To avoid these problems, joint torque sensing based on measuring the torsional deformation of the joint are proposed. This joint torque sensing that placed elastic-element in the output transmission line of joint was proposed in [9] by Tsetserukou et al. and its spin-off in [10]. In [11], a linear encoder is used to measure the torsional deformation of the additional elastic body. As measurement accuracy is in inverse proportion to sensor stiffness, low sensor stiffness is desirable in order to achieve high measurement resolution, which leads to complicated joint dynamics. As high-resolution absolute position encoders become commercially available. Mounting absolute position encoders at the link-side and motor-side of robotic joints makes it possible to measure the torsional deformation of the joint more accurate. A viable joint torque estimation method is developed by Zhang [16], [17]. The technique of virtual torque sensor uses measurements of both the link-side and motor shaft positions along with a proposed harmonic derive compliance model to realize stiff and sensitive torque estimation, obtaining joint torque with minimal mechanical modifications. Analytic models, such as harmonic derive compliance model, are often hard to model sufficiently due to transport error and coupling

characteristic which called uncertain part. So the ignoring of the uncertain part is a possible drawback of harmonic drive compliance model for joint torque estimation. For the estimation of joint torque more accurate, one can use a function approximate to the uncertain part in high-dimensional space using regression methods. A powerful regression method for accurate function approximation in high-dimensional space is Gaussian process regression [22]. As an alternative, learning the uncertain part from sampled data and achieving virtual torque sensor measurement using both harmonic drive compliance model and regression techniques poses an interesting problem.

This work builds up on the previous work reported in [17], aiming to provide improved virtual torque sensor accuracy. A method of virtual torque sensor is developed based on harmonic drive compliance model and extended Gaussian process regression. The proposed method consists of an analytical model based part and a mapping part. The analytical model based part adopts the algorithm that has been reported in [17]. The mapping part is developed by reconstructing the error model between harmonic drive compliance model and joint actual torque based on Gaussian process regression. The error model can be mapped by Gaussian process regression. Then using given samples measurements of both the link-side and motor shaft positions along with the motor current as input, and using the corresponding output harmonic drive compliance model based joint estimated torque and joint actual torque as targets, respectively. Using sampled data, after an automatic offline calibration, the method can then be used for real-time joint torque estimation. As demonstrated with experimental results, the proposed method provides accurate joint torque estimation in comparison with measurements of a commercial torque sensor.

The rest of this paper is organized as follows. The harmonic drive compliance model used to experimentally validate the theoretical developments of this paper is presented in Section II. After a short review of GPR, in Section III, we describe the proposed approach for mapping of the unmolded part. The mapping part is used to estimate the joint torque, then, the accuracy and computation speed for prediction are tested in Section IV. Experimental results using proposed joint torque estimation method are given and discussed in Section V. Concluding remarks are given in Section VI.

II. HARMONIC DRIVE-BASED JOINT MODEL

The goal of the proposed harmonic drive compliance model here, was not to develop a precise model but to generate an appropriate model that adequately reflect the behavior of joint torque. A harmonic drive consists of three main components as shown in Fig. 1. Typically the wave generator (WG) is connected to the motor shaft, the circular spline (CS) is connected to the joint housing, and the flex spline (FS) is sandwiched in between (CS and WG) and connected to the joint output. The WG consists of an elliptical disk (rigid elliptical inner-race), called wave generator plug, and an outer ball bearing. The wave generator plug is inserted into the bearing, thereby giving the bearing an elliptical shape as well. The FS fits tightly over WG; when the WG plug is rotated, the FS deforms and molds into the shape of the rotating ellipse but does not rotate with WG. The wave generator plug is inserted into the bearing, thereby giving the bearing an elliptical shape as well. The FS fits tightly over WG; when the WG plug is rotated, the FS deforms and molds into the shape of the rotating ellipse but does not rotate with WG. The modeling of harmonic drive kinematic can be seen in [13]. $\Delta\theta$ is the total torsional angle of the harmonic drive, $\Delta\theta_f$ is the torsional angle of the flex spline, $\Delta\theta_w$ is wave generator torsional angle, θ_{err} is the kinematic error. From [13],

$$\Delta\theta = \Delta\theta_f + \frac{\Delta\theta_w}{N} + \theta_{err} \quad (1)$$

A model of the compliance of harmonic drive component was proposed by Zhang et al. (2015) [16]. One has:

$$\tau_{model} = \frac{\tan(\Delta\theta_f c_f K_f)}{c_f} \quad (2)$$

where c_f and K_f are known parameters.

From (1), $\Delta\theta_f$ can be modified as

$$\Delta\theta_f = \Delta\theta - \frac{\text{sgn}(\tau_w)}{c_w N K_w} (1 - e^{-c_w |\tau_w|}) - \theta_{err} \quad (3)$$

where c_w , N , and K_w are known parameters. τ_w can be calculated by the motor current, approximately.

The joint-deformation angle $\Delta\theta$ can be defined as

$$\Delta\theta = \theta_f - \theta_w \quad (4)$$

where θ_f , θ_w denote link-side and motor-side angles, respectively.

Kinematic error θ_{err} can be written as

$$\begin{aligned} \theta_{err} = & a_0 + a_{11} \cos(\theta_f \omega_f) + b_{11} \sin(\theta_f \omega_f) + a_{\omega 1} \cos(\theta_w \omega_w) + b_{\omega 1} \sin(\theta_w \omega_w) \\ & + a_{\omega 2} \cos(2\theta_w \omega_w) + b_{\omega 2} \sin(2\theta_w \omega_w) \end{aligned} \quad (5)$$

where $a_0, a_{11}, b_{11}, \omega_f, a_{\omega 1}, b_{\omega 1}, a_{\omega 2}, b_{\omega 2}, \omega_w$ are known parameters.

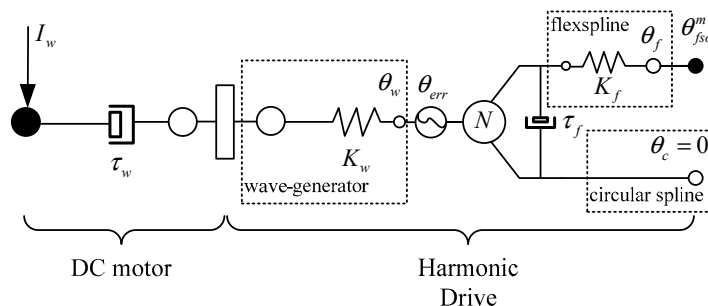


Fig. 1. Schematic illustration showing harmonic drive compliance components

One can use the compliance model (2) – (4) to estimate joint torque. As mentioned in the literature [15] - [16], the maximum difference between the estimated torque and the torque sensor measurement is 1.8245 N·m, about 36% of the total test torque. The RMS level of the difference is 0.445 N·m, about 10% of the whole test torque. The poor accuracy limit the method widely used. The main difference is generated by un-modeling part. Next, we explore a new method using Gaussian process regression based learning algorithm to map the un-modeling part. The overall structure of the torque estimation method proposed in this paper can be shown as in Fig. 2. The method consists of an analytical modeling part and a mapping part. The analytical modeling part is developed by a harmonic drive compliance model as (2) – (4). The mapping part is developed by reconstructing the un-modeling parting from motor current as well as motor-side and link-side joint angles to the torque estimation error.

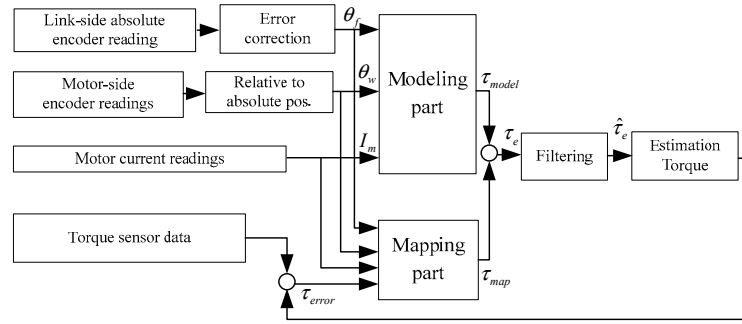


Fig. 2. Block diagram of the proposed torque estimation method.

The new method for virtual torque sensor measurement based on harmonic drives compliance model and Gaussian Process Regression, can be described as Eq. (6):

$$\tau_e = \frac{\tan(\Delta\theta_f c_f K_f)}{c_f} + f(\theta_f, \theta_w, I_m) \quad (6)$$

where $f(\theta_f, \theta_w, I_m)$ denote the un-modeling part, θ_f, θ_w, I_m denote the link-side joint angles, motor-side joint angles and motor current.

III. UN-MODELING PART LEARNING USING GAUSSIAN PROCESS REGRESSION

For the estimation of joint torque more accurate, $f(\theta_f, \theta_w, I_m)$ in Eq. (6) can be learned by the Gaussian process regression (GPR) techniques. Given samples $\mathbf{x}=[\theta_f, \theta_w, I_m]$ as input, and using the corresponding output $\mathbf{y}=[\tau_{sensor} - \tau_{model}]$ as targets, respectively. Using sampled data $\{\mathbf{x}, \mathbf{y}\}^m$, we would like to reconstruct a function $f(\mathbf{x})$ transforming the input vector \mathbf{x} into the target value \mathbf{y} given by Eq. (7):

$$\mathbf{y} = f(\mathbf{x}) + \varepsilon \quad (7)$$

where ε is Gaussian noise with zero mean and variance σ_n^2 .

As a result, the observed targets can also be described as Eq. (8):

$$\mathbf{y} \sim N(\boldsymbol{\theta}, \mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I}) \quad (8)$$

where $\mathbf{K}(\mathbf{X}, \mathbf{X})$ denotes the covariance matrix.

As covariance function, a Gaussian kernel is frequently used as Eq. (9):

$$\mathbf{k}(\mathbf{x}_p, \mathbf{x}_q) = \sigma_s^2 \exp\left(-\frac{1}{2}(\mathbf{x}_p - \mathbf{x}_q)^T \mathbf{W}(\mathbf{x}_p - \mathbf{x}_q)\right) \quad (9)$$

where σ_s^2 denotes the signal variance and \mathbf{W} are the widths of the Gaussian kernel.

The joint distribution of the observed target values and predicted value for a query point \mathbf{x}_* is given by Eq. (10):

$$\begin{bmatrix} \mathbf{y} \\ f(\mathbf{x}_*) \end{bmatrix} \sim N\left(\mathbf{0}, \begin{bmatrix} \mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I} & \mathbf{k}(\mathbf{X}, \mathbf{x}_*) \\ \mathbf{k}(\mathbf{x}_*, \mathbf{X}) & \mathbf{k}(\mathbf{x}_*, \mathbf{x}_*) \end{bmatrix}\right) \quad (10)$$

The conditional distribution yields the predicted mean value $f(\mathbf{x}_*)$ with the corresponding variance $\mathbf{V}(\mathbf{x}_*)$:

$$f(\mathbf{x}_*) = \mathbf{k}_*^T (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{y} = \mathbf{k}_*^T \boldsymbol{\alpha} \quad (11)$$

$$V(\mathbf{x}_*) = \mathbf{k}(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^T (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{k}_* \quad (12)$$

with $\mathbf{k}_* = \mathbf{k}(\mathbf{X}, \mathbf{x}_*)$, $\mathbf{K} = \mathbf{K}(\mathbf{X}, \mathbf{X})$ and $\boldsymbol{\alpha}$ denotes the so-called prediction vector. The hyper parameters of a Gaussian process with Gaussian kernel are $\boldsymbol{\theta} = [\sigma_n^2, \sigma_f^2, \mathbf{W}]$ and their optimal value for a particular data set can be derived by maximizing the log marginal likelihood using common optimization procedures, e.g., Quasi-Newton methods [22].

The major limitation of Gaussian process is the expensive computation of the inverse matrix $(\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1}$ which yields a cost of $O(n^3)$. Thus, our algorithm consists out of two stages: (i) clustering of data, i.e., insertion of new data points into the local models, (ii) learning of corresponding local models. Clustering input data can be performed efficiently using a similarity measure between the input point \mathbf{x} and the centers of the respective local models. From a machine learning point of view, the similarity or proximity of data points can be defined in terms of a kernel. Kernel functions represent the dot product between two vectors in the feature space and, hence, naturally incorporate the similarity measure between data points. The clustering step described in this section results from the basic assumption that nearby input points are likely to have similar target values. Thus, training points that belong to the same local region (represented by a center) are informative about the prediction for query points next to this local region.

So, to reducing this computational cost, we cluster the training data in local regions and, subsequently, train the corresponding Gaussian process models on these local clusters. Clustering input data is efficiently performed by considering a distance measure of the input point \mathbf{x} to the centers of all local models. The distance measure w_k is given by the kernel used to learn the local Gaussian process models, e.g., Gaussian kernel:

$$w_k = \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{c}_k)^T \mathbf{W}(\mathbf{x} - \mathbf{c}_k)\right) \quad (13)$$

where \mathbf{c}_k denotes the center of the k -th local model and \mathbf{W} a diagonal matrix represented the kernel width.

Since the hyper parameters of a Gaussian process model can be achieved by likelihood optimization, it is straightforward to adjust the open parameters for the similarity measure. For example, we can subsample the available training data and, subsequently, perform the standard optimization procedure. It should be noted that during online learning, we have to deal with an endless stream of data (e.g., at a 500-Hz sampling rate we get a new data point every 2 ms and have to treat 30 000 data points/min). If one want cope with the real-time requirements, the maximum number of training examples needs to be limited so that the local models do not end up with the same complexity as a standard GPR regression. The localization procedure is summarized in Fig. 3.

```

Input: new data point  $\{x, y\}$ .
For  $k=1$  to number of local models do
  Compute distance to the  $k$ -th local model:
     $w_k = \exp(-\frac{1}{2}(x - c_k)^T W (x - c_k))$ 
End
For
  Take the nearest local model:
     $v = \max(w_k)$ 
  If  $v > w_{gen}$  then
    Insert  $\{x, y\}$  to nearest local model:
       $X_{new} = [X, x]$ 
       $y_{new} = [Y, y]$ 
    Update corresponding center:
       $c_{new} = \text{mean}(X_{new})$ 
    Compute inverse covariance matrix and
    prediction vector of local model:
       $K_{new} = K(X_{new}, X_{new})$ 
       $a_{new} = (K_{new} + \sigma^2 I)^{-1} y_{new}$ 
  else
     $c_{k+1} = x$ 
     $K_{k+1} = [x], y_{k+1} = [y]$ 
End

```

Fig. 3. Partitioning of training data and model learning.

We train the un-modeling part using sampled data from real joint, e.g., test-joint presented in [16] (335720 training points, 3000 test points), as well as a physically realistic simulation (340940 training points, 3000 test points) [24]. Fig. 14 show the errors on learning the un-modeling part using sampled data, where the error is computed as root mean squared error (RMSE). Here, partitioning of the training examples for LGP can be performed either in the same input space (where the model is learned) or in another space which has to be physically consistent with the approximated function. In the following, we localize the data depending on the position and partition of training data is performed in a 6-dim space. On our current hardware, a Core Duo at 2GHz, the training time can be controlled by the number of local models M . A large number of local models may provide a smooth approximation but on the other hand increases the prediction time complexity, which will be discussed in the following Section IV.

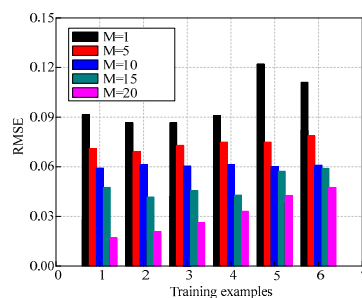


Fig. 4. The errors on learning the un-modeling part with different local models

IV. UN-MODELING PART PREDICTION USING GAUSSIAN PROCESS REGRESSION

A. Prediction for a Query Point

The prediction for a mean value \hat{y} is performed using weighted averaging over M local predictions \bar{y}_k for a query point x . The weighted prediction \hat{y} is then given by $\hat{y} = E\{\bar{y}_k | x\} =: \sum_{k=1}^M \bar{y}_k p(k|x)$. According to the Bayesian

theorem, the probability of the model k given \mathbf{x} can be expressed as $p(k|\mathbf{x}) = p(k|\mathbf{x}) / \sum_{k=1}^M p(k|\mathbf{x}) = \mathbf{w}_k / \sum_{k=1}^M \mathbf{w}_k$. Hence, we have

$$\hat{\mathbf{y}} = \frac{\sum_{k=1}^M \mathbf{w}_k \bar{\mathbf{y}}_k}{\sum_{k=1}^M \mathbf{w}_k} \quad (14)$$

The probability $p(k|\mathbf{x})$ can be interpreted as a normalized distance of the query point \mathbf{x} to the local model k where the measure metric \mathbf{w}_k is used as given in Eq. (13). Thus, each local prediction $\bar{\mathbf{y}}_k$, determined using Equation (11), is additionally weighted by the distance \mathbf{w}_k between the corresponding center \mathbf{c}_k and the query point \mathbf{x} . The prediction procedure is summarized in Fig. 5.

```

Input: query data point  $\mathbf{x}, M$ .
Determine  $M$  local models next to  $\mathbf{x}$ .
for  $k=1$  to  $M$  do
    Compute distance to the  $k$ -th local model:
         $\mathbf{w}_k = \exp(-\frac{1}{2}(\mathbf{x} - \mathbf{c}_k)^T \mathbf{W}(\mathbf{x} - \mathbf{c}_k))$ 
    Compute local mean using the  $k$ -th local
    model:
         $\bar{\mathbf{y}}_k = \mathbf{k}_k^T \mathbf{a}_k$ 
End for
    Compute local mean using the  $k$ -th local
    model:
         $\hat{\mathbf{y}}_k = \mathbf{k}_k^T \mathbf{a}_k$ 
End

```

Fig. 5. Prediction for a query point.

B. Accuracy for Prediction

To test the prediction accuracy of the learning model, we use two data sets, (i) The gradually changing torque, getting the data from the torque sensor and torque estimation using harmonic drive compliance model (ii) The frequent changing torque, getting the data from the torque sensor and torque estimation using harmonic drive compliance model. The number of local models was defined as $M=20$. The data comparison of torque sensor and the joint torque evaluation using harmonic drive compliance model show as in Fig. 6 and Fig. 8. Fig. 7 and Fig. 9 denote the prediction accuracy for the error. From Fig. 7 and Fig. 9, we can see that the accuracy of the prediction for the un-modeling part is about 0.1Nm, approximate 1.25% of the maximum joint torque.

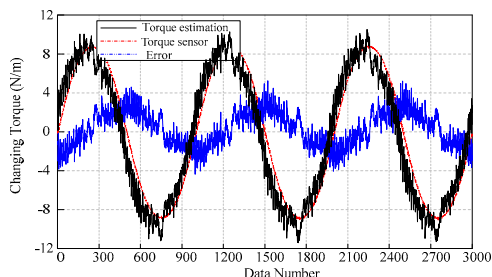


Fig. 6. The test of gradually changing torque

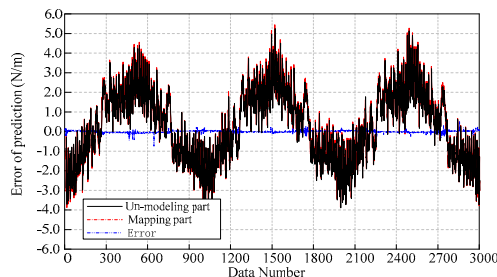


Fig. 7. Error of prediction when torque gradually changing

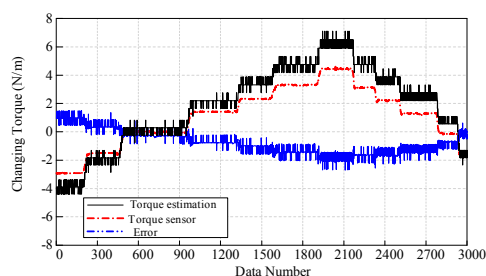


Fig. 8. The test of frequent changing torque

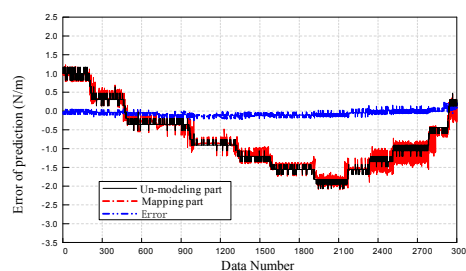


Fig. 9. Error of prediction when torque frequent changing

C. Computation Speed for Prediction

Beside the accuracy for prediction, reduction of prediction time (i.e., matrix inversion), is also very important due to the need during control cycle. On our current level, the prediction time can be controlled by the number of local models. The comparison of computation speed for prediction is shown in Fig. 10. Here, we use data sets with increasing training examples (5000, 20000, 80000, 320000, and 640000 data points, respectively). Subsequently, using different number of local models, we compute the average time needed to make a prediction for a point.

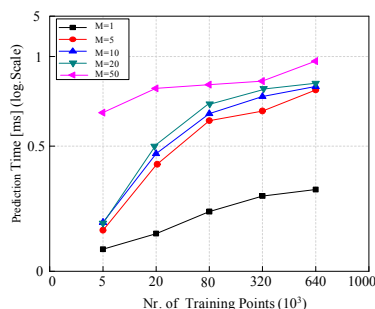


Fig. 10. Average time in millisecond needed for prediction of one query point.

V. EXPERIMENTAL RESULTS

A. Joint Torque Estimation

This section provides the experimental results of testing the effectiveness of the proposed joint torque estimation method. Experiments were performed, in which the estimated joint torque was compared with the torque measurement from the commercial F/T sensor. The experimental setup used to conduct the experimental verification is shown in Fig. 11.

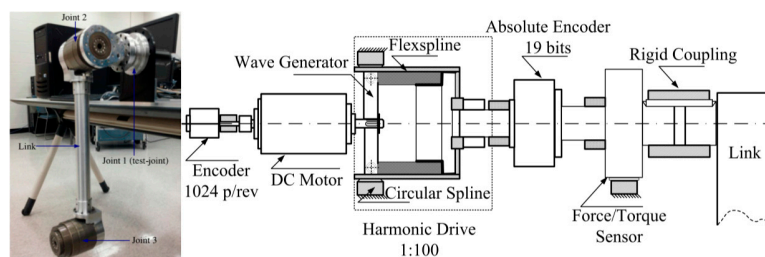


Fig. 11. Experimental setup and schematic diagram of virtual torque sensor.

Four different experiments were performed. In the first experiment, the load torque was changed slowly by controlling the test-joint motion with a sinusoidal reference trajectory. The gradually changing load torque was introduced through moving links and joints driven by the test-joint. The torque estimate and the torque sensor output

were recorded as a function of time, as depicted in Fig. 12. The maximum difference between the estimated torque and the torque sensor measurement is $0.28 \text{ N}\cdot\text{m}$, approximate 2.75% of the maximum joint torque and the RMS level of the difference is $0.08 \text{ N}\cdot\text{m}$, approximate 1.62% of the RMS joint torque.

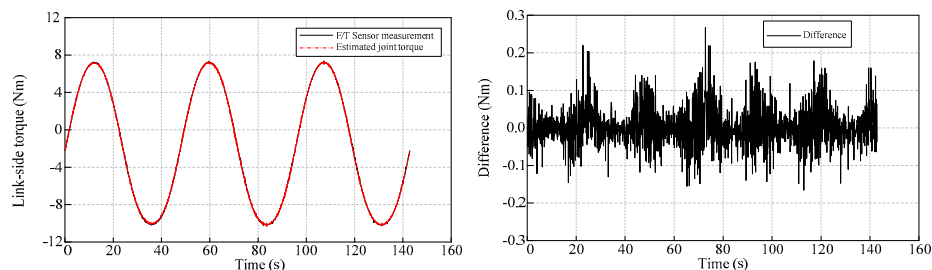


Fig. 12. Estimated torque versus torque measured by F/T sensor (response to slow changes).

In the second experiment, the results of which are depicted in Fig. 13, the load torque was changed rapidly by manually applying external torque in both directions. The maximum difference between the estimated torque and the torque sensor measurement for the second experiment is $0.35 \text{ N}\cdot\text{m}$, approximate 3.18% of the maximum joint torque and the RMS level of the difference is $0.125 \text{ N}\cdot\text{m}$, approximate 2.53% of the RMS joint torque.

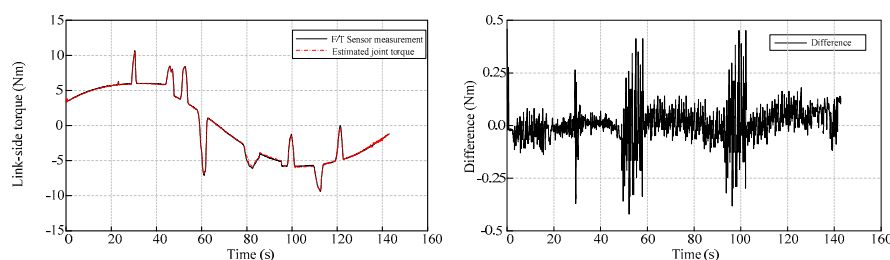


Fig. 13. Estimated torque versus torque measured by F/T sensor (response to fast changes).

In the third experiment, a 10-N weight is attached to the link by a wire at a distance of 400 mm from the first joint axis. The test-joint is controlled to keep its position constant and the load was released suddenly by cutting the wire. The results of which are shown in Fig. 14. This strategy was driven by the intention of verifying the response of the proposed torque estimation method to sudden payload changes. The maximum difference between the estimated torque and the torque sensor measurement for the second experiment is $0.40 \text{ N}\cdot\text{m}$, approximate 3.33% of the maximum joint torque and the RMS level of the difference is $0.055 \text{ N}\cdot\text{m}$, approximate 0.63% of the RMS joint torque.

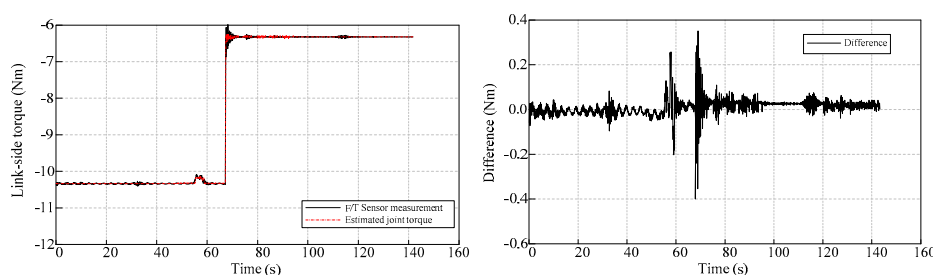


Fig. 14. Estimated torque versus torque measured by F/T sensor (response to sudden payload changes).

In the fourth experiment, frequent payload changes were applied at the link-side of the test-joint. The test-joint was loaded to the rated torque limit of the harmonic drive ($16 \text{ N}\cdot\text{m}$). The result of this experiment is shown in Fig.

15, where the maximum difference between the estimated torque and the torque sensor measurement for the experiment is 0.58 N·m, approximate 3.87% of the maximum joint torque and the RMS level of the difference is 0.125 N·m, approximate 2.5% of the RMS joint torque.

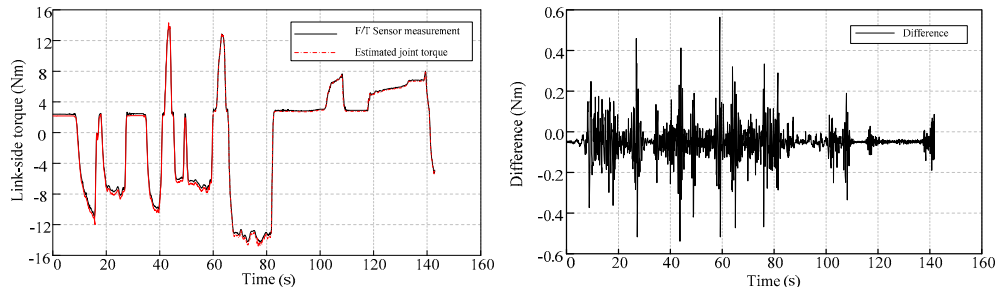


Fig. 15. Estimated torque versus torque measured by F/T sensor (response to frequent payload changes).

B. Motion Control With Joint Torque Feedback

In this section, first, we use proposed joint torque estimation model in Section IV for a JTF -based tracking control task [10]. Here, the joint torque estimation model is applied for an online prediction of the feedforward torques \mathbf{u}_{FF} given the desired trajectory $[\mathbf{q}_d \quad \dot{\mathbf{q}}_d \quad \ddot{\mathbf{q}}_d]$. Subsequently, the model approximated by LGP is used for an online prediction performance. Demonstrating the online prediction, the GP models are adapted in real-time using rank-one update. As shown in Fig. 16, the controller command \mathbf{u} consists of the feedforward part \mathbf{u}_{FF} and the feedback part $\mathbf{u}_{FB} = \mathbf{K}_p \mathbf{e} + \mathbf{K}_v \dot{\mathbf{e}}$, where $\mathbf{e} = \mathbf{q}_d - \mathbf{q}$ denotes the tracking error and \mathbf{K}_p , \mathbf{K}_v denote position-gain and velocity-gain, respectively. During the control experiment we set the gains to very low values taking the aim of compliant control into account. As a result, the proposed joint torque model has a stronger effect on computing the predicted torque \mathbf{u}_{FF} and, hence, a better performance of method results in a lower tracking error.

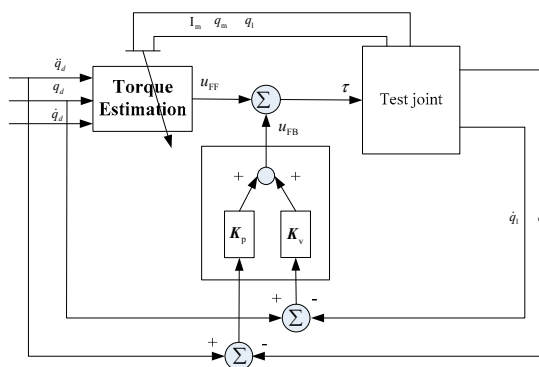


Fig. 16. Schematic showing model-based robot control.

The external torque applied to the test-joint, which can be measured by the F/T sensor or estimated based on the proposed torque estimation method. As desired trajectory, we generate a test trajectory which is similar to the one used for learning the joint torque estimation in Fig. 12 shows the desired trajectory and real trajectory for the test-joint, while the position tracking error of the motion control experiment is shown in Fig. 18. During the first 40 s of this experiment, no JTF was provided, while JTF using estimated torque of the proposed method during the subsequent 40 s. The position error when using JTF from F/T sensor is shown in the last 40 s of the experiment time.

The RMS values of the tracking error are 0.15, 0.055, and 0.035 for the cases when no JTF is used, JTF with the torque estimate, and JTF with the torque sensor, respectively. This results show 63% and 77% improvements when using JTF with the torque estimate and the torque sensor readings, respectively.

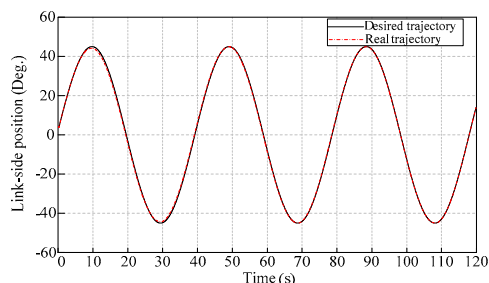


Fig. 17. Trajectory comparison for motion control.

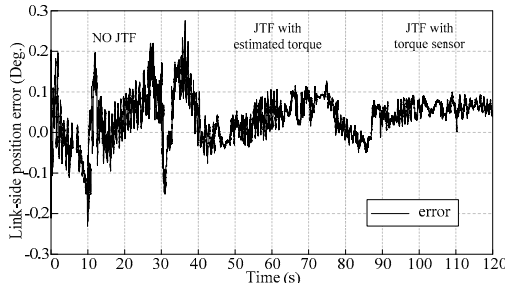


Fig. 18. Motion control experimental results.

VI. CONCLUSION

In this paper, we have proposed a new method for the virtual torque sensor based on harmonic drive compliance model and local Gaussian process regression. An advantage associated with the method using other joint torque sensing is its accuracy. Using the proposed method for the joint torque estimation instead of an additional elastic component does not change the joint dynamics and simplify the joint mechanism design and build. Moreover, adding a link-side position sensor leads to the potential to improve joint control accuracy and reduce the cost of joint torque sensing.

We have successfully demonstrated our approach on the robotic joint with motor current reader, motor-side encoders and link-side encoders. We showed that our method can be used to dramatically improve the efficiency and accuracy of estimating. We believe that the use will allow this approach to be applicable to other robotic systems as well. Furthermore, the estimated torque can find its applications in robotic control algorithms with JTF, as well as in collision detection/avoidance schemes. In the future, we aim to extend our approach to operating more complex manipulator.

REFERENCES

- [1] F. Aghili, M. Buehler, and J. M. Holler Bach, "Motion control systems with H_∞ positive joint torque feedback," *IEEE Transactions on Control System Technology*, vol. 9, no. 5, pp. 685–695, Sept. 2001.
- [2] F. Aghili and M. Namvar, "Adaptive control of manipulators using uncalibrated joint-torque sensing," *IEEE Trans. Robot.*, vol. 22, no. 4, pp. 854–860, Aug. 2006.
- [3] G. Liu, S. Abdul, and A.A. Goldenberg, "Distributed modular and reconfigurable robot control with torque sensing," in *Proc. IEEE Int. Conf. Mechatron. Autom.*, Jun. 2006, pp. 384–389.
- [4] W.-H. Zhu, E. Dupuis, and M. Doyon, "Adaptive control of harmonic drives," *ASME Trans. Dyn. Syst., Meas., Contr.*, vol. 129, no. 2, pp. 182–193, Aug. 2007.
- [5] L.L. Tien, A. Albu-Schaffer, A. D. Luca, and G. Hirzinger, "Friction observer and compensation for control of robots with joint torque measurement," in *Proc. IEEE/RSJ Int. Conf. Intel. Robot. Syst.*, Sep. 2008, pp. 3789–3795.

- [6] F. Aghili, J. Hollerbach, and M. Buehler, "A modular and high-precision motion control system with an integrated motor," *IEEE/ASME Trans. Mechatronics*, vol. 12, no. 3, pp. 317–329, Jun. 2007.
- [7] G. Liu, S. Abdul, and A.A. Goldenberg, "Distributed control of modular and reconfigurable robot with torque sensing," *Robotica*, vol. 26, pp. 75–84, Jan. 2008.
- [8] M. Randazzo, M. Fumagalli, F. Nori, L. Natale, G. Metta, and G. Sandini, "A comparison between joint level torque sensing and proximal f/t sensor torque estimation: Implementation on the iCub," in *Proc. IEEE/RSJ Int. Conf. Intel. Robot. Syst.*, Sep. 2011, pp. 4161–4167.
- [9] D. Tsetserukou, R. Tadakuma, H. Kajimoto, and S. Tachi, "Optical torque sensors for implementation of local impedance control of the arm of humanoid robot," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2006, pp. 1674–1679.
- [10] S. Shams, D. Shin, J. Han, J.-Y. Lee, K. Shin, and C.-S. Han, "Compact design of a torque sensor using optical technique and its fabrication for wearable and quadruped robots," in *Proc. IEEE/RSJ Int. Conf. Intel. Robot. Syst.*, Sep. 2011, pp. 5127–5132.
- [11] T. Kawakami, K. Ayusawa, H. Kaminaga, and Y. Nakamura, "High fidelity joint drive system by torque feedback control using high precision linear encoder," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 3904–3909.
- [12] M. Hashimoto, Y. Kiyosawa, and R. Paul, "A torque sensing technique for robots with harmonic drives," *IEEE Trans. Robot. Autom.*, vol. 9, no. 1, pp. 108–116, Feb. 1993.
- [13] H. Taghirad and P. Belanger, "Torque ripple and misalignment torque compensation for the built-in torque sensor of harmonic drive systems," *IEEE Trans. Instrum. Meas.*, vol. 47, no. 1, pp. 309–315, Feb. 1998.
- [14] I. Godler, M. Horiuchi, M. Hashimoto, and T. Ninomiya, "Accuracy improvement of built-in torque sensing for harmonic drives," *IEEE/ASME Trans. Mechatron.*, vol. 5, no. 4, pp. 360–366, Dec. 2000.
- [15] J. Sensinger and R. Weir, "Improved torque fidelity in harmonic drive sensors through the union of two existing strategies," *IEEE/ASME Trans. Mechatron.*, vol. 11, no. 4, pp. 457–461, Aug. 2006.
- [16] H. Zhang, S. Ahmad, and G. Liu, "Torque estimation technique of robotic joint with harmonic drive transmission," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2013, pp. 3034–3039.
- [17] H. Zhang, S. Ahmad, and G. Liu, "Torque Estimation for Robotic Joint with Harmonic Drive Transmission Based on Position Measurements," *IEEE TRANSACTIONS ON ROBOTICS*. APRIL 2015, pp. 3034–3039.
- [18] J. Seo, M.J. Yu, C.G. Park, and J.G. Lee, "An extended robust H_∞ filter for nonlinear constrained uncertain systems," *IEEE Transactions on Signal Processing*. vol. 54, no. 11. pp. 4471-4475, 2006.
- [19] K. Xiong, H. Zhang, and L. Liu, "Adaptive robust extended Kalman filter for nonlinear stochastic systems," *IET Control Theory and Applications*, vol. 2, no. 3, pp. 248-250, 2008.
- [20] Y.K. Li, Z.L. Jing, and S.Q. Hu, "Redundant adaptive robust tracking of active satellite and error evaluation," *IET Control Theory and Application*, vol. 4, no. 11, 2010. pp. 2539-2553.
- [21] S. Schaal, C. G. Atkeson, and S. Vijayakumar, "Scalable techniques from non-parametric statistics for real-time robot learning," *Applied Intelligence*, pp. 49–60, 2002.

- [22] J. Q. Candela and C. E. Rasmussen, "A unifying view of sparse approximate Gaussian process regression," *Journal of Machine Learning Research*, 2005.
- [23] C. E. Rasmussen and Z. Ghahramani, "Infinite mixtures of gaussian process experts," *Advances in Neural Information Processing Systems*, 2002.
- [24] Gao, H.; Xia, K.; Ding, L.; Liu, G.; Deng, Z.; Liu, Z. Mobile Manipulator Simulation Based on Contact Mechanics and Dynamics with Motion Control [J]. *International Journal of Nonlinear Sciences and Numerical Simulation*, IJNSNS.2015.0042 (under review), 2015.
- [25] Ding, L.; Xia, K.; Gao, H.; Liu, G.; Deng, Z.; Liu, Z. Robust Adaptive Control of Door Opening by a Mobile Rescue Manipulator Based on Unknown Force-related Constraints Estimation Using Position Measurements [J]. *International Journal of Control*, TCON-2015-0656 (under review), 2015.
- [26] H. Zhang, S. Ahmad, and G. Liu, "Modeling of torsional compliance and hysteresis behaviors in harmonic drives," *IEEE/ASME Trans. Mechatron.*, vol. 20, no. 1, pp. 178–185, Feb. 2015.
- [27] S. Ahmad, H. Zhang, and G. Liu, "Multiple working mode control of door opening with a mobile modular and reconfigurable robot," *IEEE/ASME Trans. Mechatron.*, vol. 18, no. 3, pp. 833–844, Jun. 2013.



© 2016 by the authors; licensee *Preprints*, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons by Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).