External Force Estimation During Compliant Robot Manipulation

Adrià Colomé, Diego Pardo, Guillem Alenyà and Carme Torras

***Abstract*— This paper presents a method to estimate external forces exerted on a manipulator during motion, avoiding the use of a sensor. The method is based on task-oriented dynamics model learning and a robust disturbance state observer. The combination of both leads to an efficient torque observer that can be incorporated to any control scheme. The use of a learning-based approach avoids the need of analytical models of joints' friction or Coriolis dynamics effects.**

I. Introduction

Nowadays robots adequately perform diverse manipulation tasks with high degree of autonomy and precision. Nevertheless, tasks requiring interaction with humans impose safety restrictions that still need to be addressed. The robotics research community is actively working on generating solutions to realize robotics abilities to support daily life domestic tasks [1] [2], such as manipulating cloth (see Fig. 1).

Robots able to safely interact with their surroundings should have structural features like lightweight links and coupled joints actuators mechanisms [3] enabling them to perform compliant motions. Besides these, their low-level control architecture should avoid excessive stiffness, usually imposed by accuracy demands.

Another major ingredient for the achievement of compliant robot behaviors is the need to supervise the external forces (and torques) generated along the robot motions. External forces may play diverse roles during the planning and execution of compliant robot motions. For instance, in force control schemes, the external manipulator wrench $\mathbf{f_e} \in \mathbb{R}^6$ is compared to a reference signal in order to have a desired end-effector interaction with the environment. Other schemes such as compliant control, impedance control or hybrid control also use the external wrench data to compute the corresponding system action [4].

For the purpose of making available the external wrench felt by a robot manipulator, expensive sensors are often used. In order to avoid the use of such devices, recent works [5] [6] present approaches for estimating the wrench or, at least, the joint torques due to an external action during manipulation.

However, most of the current approaches are based on the availability of an accurate analytical model of the robot dynamics, which may lead to inaccuracy due to modeling errors. This is specially true in modern robotics systems which are highly non-linear and can no longer be accurately modeled using the rigid body dynamics. In the specific case of the Barrett WAM [3], the analytical dynamic model becomes harder and much more complex to obtain for structural reasons, given that several spinning drives, some of them coupled, are in different references frames from the actuated joint while only one can be measured, resulting in effects such as reflective inertias.

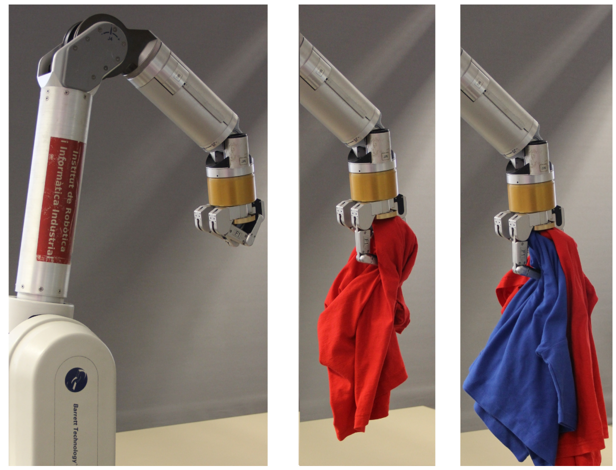

Fig. 1. 7-dof WAM robot holding none, one and two cloth garments. A proper external force estimation would help the robot to know how many garments have been picked after an action.

Moreover, in a lightweight robot, any small error in the dynamic parameters like the link masses represents a large percentage error for the model accuracy. Interestingly, those structural features that allow a robot to be compliant make it harder to model and, therefore, estimation of contact forces using state-of-the-art methods is more difficult, which conversely imposes restrictions to the exploitation of the physical compliance capability of the robot.

This document presents an approach based on machine learning techniques and disturbance state observers for the estimation of external forces/torques felt by the robot during common motion tasks. The presented method is based on the state-of-the-art on external forces estimation but extending it to those cases where an analytical model of the robot dynamics is not available/feasible. Moreover, we take into consideration the well-known issue that the use of accelerations is undesirable due to the error introduced by the numerical differentiation, to elaborate a better contact force

This work is partially supported by EU Projects IntellAct (FP7-269959) and μRALP (FP7-288233), by the Spanish Ministry of Science and Innovation under project PAU+ DPI2011-27510, and by the Catalan Research Commission through SGR-00155.

A. Colomé, G. Alenyà and C. Torras are with the Institut de Robòtica i Informàtica Industrial (CSIC-UPC), Llorens Artigas 4-6, 08028 Barcelona, Spain. E-mails: [acolome,torras,galenya]@iri.upc.edu. D. Pardo is with the Department of Advanced Robotics, Italian Institute of Technology (IIT), Via Morego 30, 16163 Genova, Italy. E-mail: diego.pardo@iit.it.

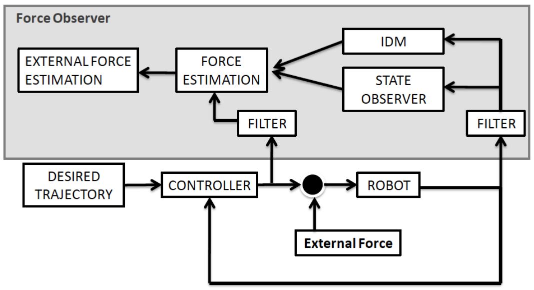

Fig. 2. The proposed scheme can be run parallel to any controller.

estimator, which will be built parallel to the controller as shown in Fig. 2.

The paper is organized as follows: in Section II we define the inverse dynamic models and learning techniques used for our observers, to be used later in Section III, where we define our external force estimator, which is tested in Section IV.

II. Learning Robot Inverse Dynamics

Modern methods for wrench estimation are based on the use of state space observers [5][6]. Intuition behind this idea is that the robot is experimenting external forces that produce changes in its state, therefore, by estimating the internal state of the system and assuming that certain part of the total inputs is known, an estimation of perturbations (external inputs/forces) can be completed. As described in Section III, such observers are based on the availability of the analytical model of the manipulator dynamics. Here we assume that such model is not available and therefore we discuss this issue providing the required elements for the proposed wrench estimator.

The dynamics of a serial robot, as described in [4], is given by:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) + \mathbf{F_f}(\mathbf{q},\dot{\mathbf{q}}) = \mathbf{u_T}, \qquad (1)$$

where $\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}} \in \mathbb{R}^n$ denote joint angles, velocities and acceleration of the robot with n degrees of freedom (DoF), $\mathbf{M}(\mathbf{q})$ represents the inertia matrix, and $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}}), \mathbf{G}(\mathbf{q})$ and $\mathbf{F_f}(\mathbf{q})$ are the Coriolis and centripetal, gravity and friction forces acting on the robot. Finally, vector $\mathbf{u_T} \in \mathbb{R}^n$ is the vector of total input forces to the joints. We assume that such forces may proceed from applied torque commands $\mathbf{u_c}$ and from certain external torque $\mathbf{u_e}$. Thus,

$$\mathbf{u_T} = \mathbf{u_c} - \mathbf{u_e}.$$

At the same time, the inverse model of the robot dynamics is a function mapping the robot state to the actions that would generate it, which in the absence of external forces would be given by

$$\mathbf{u_c} = \mathbf{g}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}}). \qquad (2)$$

To obtain this function $\mathbf{g}$, model learning is a very active research field in robot control [7] where methods are developed allowing the approximation of (2) using input/output data. The-state-of-the-art in online model learning includes methods like Locally Weighted Projection Regression (LWPR) [8] and Local Gaussian Process (LGP) [7]. These approaches allow to improve the model even when the system is in operation. Here we used the LWPR open access library [9] in order to approximate the inverse dynamics of the robot.

Assuming that the function $\mathbf{g}$ has been learned, it can be stated that, given a dynamic state produced by both control and external torques, the inverse model would provide,

$$\mathbf{u_T} = \mathbf{u_c} - \mathbf{u_e} = \mathbf{g}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}}),$$

and as the vector $\mathbf{u_c}$ is assumed to be known, the estimation of the external torque would be straightforward.

However, there is a set of practical considerations that points towards the use of a state observer for the external wrench estimation.

A. Local learning vs. Global learning

In the case of a 7-dof robot such as the WAM robot, learning a function that maps a joint position, velocity and acceleration to a torque vector means a dimension 21 input and a dimension 7 output. This high dimensionality makes global learning difficult to achieve, as it would generate a large number of kernel functions to evaluate when using the model, and so a slow computation rate. In addition, various unmodelled friction factors such as static/dynamic friction, motor cogging and others cause different residual friction at a same position, depending on the trajectory followed. For this reason, global inverse dynamics learning is left for future research.

B. Learning with accelerations

Measurements obtained for the WAM arm are joint positions, velocities (obtained by differentiating positions), and accelerations (as a second derivative). These derivatives are very sensitive to noise, making the simple approximation unsuitable.

A very small noise on a joint position measurement results in a very large noise in acceleration measures. Even with the use of heavy filters, such as Parks-McClellan filters [10], which minimizes error in pass-and-stop bands and used here to damp frequencies repesenting high acceleration on joints, the noise could not be completely mitigated to have a good dataset for learning a task. In order to overcome this problem, we decided to use the desired trajectory $\mathbf{q}$, $\dot{\mathbf{q}}$, $\ddot{\mathbf{q}}$ of the robot when learning a task and, instead of learning the whole dynamic system, we propose to learn the function:

$$\mathbf{u_c} - \mathbf{M}(\mathbf{q_d})\ddot{\mathbf{q}}_\mathbf{d} = \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) + \mathbf{F_f}(\mathbf{q},\dot{\mathbf{q}}),$$

that is, assuming that the only parameter of the robot to be known is the inertia matrix $\mathbf{M}(\cdot)$,

$$\mathbf{u_c} - \mathbf{M}(\mathbf{q_d})\ddot{\mathbf{q}}_\mathbf{d} = \mathbf{n}(\mathbf{q},\dot{\mathbf{q}}).$$

This function, $\mathbf{n}(\mathbf{q},\dot{\mathbf{q}})$, only depends on the joint positions and velocities which allows an accurate learning. Figure 3 presents an example of the data used to learn this relation.

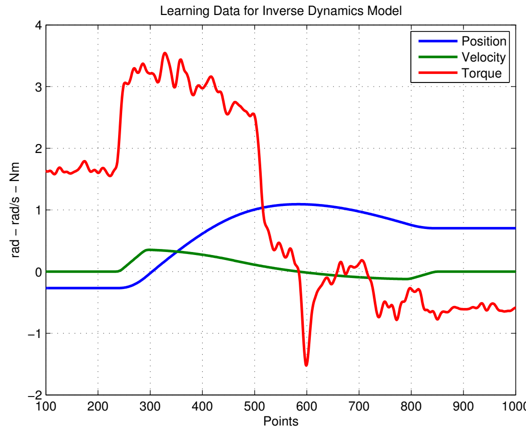

Fig. 3. Data used for learning a trajectory, using position and velocity as inputs and torque as output.

III. External Wrench Estimation

In this section we describe how this proposed function approximation can be used to estimate the external wrench when present. Equation (1) can be rewritten as

$$\begin{array}{rcl} \dot{\mathbf{x}}_1 & = & \mathbf{x}_2 \\ \dot{\mathbf{x}}_2 & = & \mathbf{\Gamma}(\mathbf{u}_c, \mathbf{x}) - \mathbf{M}^{-1}(\mathbf{x_1})\mathbf{u}_e \end{array}, \quad (3)$$

where $\mathbf{x} = [\mathbf{x}_1 \quad \mathbf{x}_2]^T$, with $\mathbf{x}_1 = \mathbf{q}$ and $\mathbf{x}_2 = \dot{\mathbf{q}}$. Here, accelerations due to external forces have been separated from those produced by gravity, Coriolis, internal friction and torque commands, which have been gathered in the term,

$$\begin{array}{rl} \mathbf{\Gamma}(\mathbf{u}_c, \mathbf{x}) = & \mathbf{M}^{-1}(\mathbf{x}_1)[\mathbf{u}_c - \mathbf{C}(\mathbf{x}_1, \mathbf{x}_2)\mathbf{x}_2 - \mathbf{F_f}(\mathbf{x}_1, \mathbf{x}_2) \\ & -\mathbf{G}(\mathbf{x}_1)] = \mathbf{M}^{-1}(\mathbf{x}_1)[\mathbf{u}_c - \mathbf{n}(\mathbf{x_1}, \mathbf{x_2})], \end{array} \quad (4)$$

where $\mathbf{\Gamma}$ can be evaluated with the measurements of $\mathbf{u_c}$ and the learned function $\mathbf{n}$.

In [6] a force estimator is presented, which basically computes a state observer with gain $\mathbf{K}$ and deduces that the position error from the observer is due to an external force that can be computed as the *missing* force to make the observer perfectly track the state:

$$\mathbf{\Delta}_2\ddot{\mathbf{e}}_1 + \mathbf{\Delta}_1\dot{\mathbf{e}}_1 + \mathbf{\Delta}_0\mathbf{e}_1 = \mathbf{u}_e, \quad (5)$$

where $\mathbf{e}_1$ is the position estimation error, $\mathbf{\Delta}_2$ the inertia matrix, $\mathbf{\Delta}_1$ various coriolis and friction terms, and $\mathbf{\Delta}_0 = -\mathbf{K}$, thus at stationary response,

$$\mathbf{f}_e = -\mathbf{J}^{T^\dagger}(\mathbf{x}_1)\mathbf{K}\mathbf{e}_1.$$

This observer does not assume a measurement of the joint velocities. However, it has the following drawbacks:

- It needs to compute the Coriolis matrix for different input values, and also the friction effects separately.
- It assumes a perfect model of the manipulator.
- No hints on the observer gains, $\mathbf{K}$, are given.
- Eq. (5) is not necessarily stable as defined. This equation is analysed in [11], where the restrictions for its convergence are given. Nevertheless, its convergence radius depends directly on the eigenvalues of the unknown matrix $\mathbf{F_f}$.
- A good value of the external force is only guaranteed at steady state, when $\dot{\mathbf{e}}_1 \simeq 0$ and $\ddot{\mathbf{e}}_1 \simeq 0$. Otherwise, as the term $\mathbf{\Delta}_1$ in equation (5) is very hard to learn or measure for learning purposes, the force estimation may have a large error. This results in a very slow response to external force steps, which can be seen in [6].

A. Proposed Observer

Keeping in mind the issues in [6], we thought to treat the external force as a disturbance of the dynamic system, and use a disturbance observer. To this purpose, in [12] a state observer for dynamic systems is proposed that also estimates external unmodelled disturbances. To follow this we can rewrite equation (3) as:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}(\mathbf{x})\mathbf{d} + \mathbf{\Gamma}^*(\mathbf{u}_c, \mathbf{x}), \quad (6)$$

with $\mathbf{d} = -\mathbf{u}_e$, $\mathbf{A} = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix}$, $\mathbf{B} = \begin{bmatrix} 0 \\ \mathbf{M}^{-1}(\mathbf{x}_1) \end{bmatrix}$, and

$$\mathbf{\Gamma}^*(\mathbf{u}_c, \mathbf{x}) = \begin{bmatrix} 0 \\ \mathbf{\Gamma}(\mathbf{u}_c, \mathbf{x}) \end{bmatrix}.$$

And then, define a state observer (using a hat to denote estimated or learned values):

$$\dot{\hat{\mathbf{x}}} = \mathbf{A}\hat{\mathbf{x}} + \mathbf{B}\hat{\mathbf{d}} + \mathbf{K}(\mathbf{x} - \hat{\mathbf{x}}) + \hat{\mathbf{\Gamma}}^*(\mathbf{u}_c, \hat{\mathbf{x}}), \quad (7)$$

which can be written as:

$$\begin{bmatrix} \dot{\hat{\mathbf{x}}}_1 \\ \dot{\hat{\mathbf{x}}}_2 \end{bmatrix} = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{x}}_1 \\ \hat{\mathbf{x}}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \mathbf{M}^{-1}(\mathbf{x}_1) \end{bmatrix} \hat{\mathbf{d}} + \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 - \hat{\mathbf{x}}_1 \\ \mathbf{x}_2 - \hat{\mathbf{x}}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \hat{\mathbf{\Gamma}}(\mathbf{u}_c, \mathbf{x}) \end{bmatrix},$$

Or, separating the two equations:

$$\begin{array}{rcl} \dot{\hat{\mathbf{x}}}_1 & = & \hat{\mathbf{x}}_2 + K_{11}(\mathbf{x_1} - \hat{\mathbf{x_1}}) + K_{12}(\mathbf{x_2} - \hat{\mathbf{x_2}}) \\ \dot{\hat{\mathbf{x}}}_2 & = & \mathbf{M}^{-1}(\mathbf{x_1})\hat{\mathbf{d}} + K_{21}(\mathbf{x_1} - \hat{\mathbf{x_1}}) + K_{22}(\mathbf{x_2} - \hat{\mathbf{x_2}}) + \\ & & +\hat{\mathbf{\Gamma}}(\mathbf{u}_c, \mathbf{x}), \end{array}$$

where $\hat{\mathbf{\Gamma}}(\mathbf{u_c}, \hat{\mathbf{x}})$ is the estimation of $\mathbf{\Gamma}(\mathbf{u_c}, \mathbf{x})$, computed as defined in Eq. (4), with the observed value of $\mathbf{x}$:

$$\hat{\mathbf{\Gamma}}(\mathbf{x}_1, \hat{\mathbf{x}}_2) = \mathbf{M}^{-1}(\mathbf{x}_1)[\mathbf{u}_c - \mathbf{n}(\mathbf{x_1}, \hat{\mathbf{x}}_2)]. \quad (8)$$

From now on, we will use $\hat{\mathbf{\Gamma}} = \hat{\mathbf{\Gamma}}(\mathbf{u_c}, \hat{\mathbf{x}})$ and $\mathbf{\Gamma} = \mathbf{\Gamma}(\mathbf{u}_c, \mathbf{x})$.

We must remark that in [12], this last term in (8) is assumed to be known. However, using its learned value will not affect the error dynamics. In fact, if the state estimation error is $\mathbf{e} = \hat{\mathbf{x}} - \mathbf{x}$ and the disturbance estimation error $\mathbf{e}_d = \hat{\mathbf{d}} - \mathbf{d}$, the error dynamics, subtracting (6) from (7), is:

$$\dot{\mathbf{e}} = (\mathbf{A} - \mathbf{K})\mathbf{e} + \mathbf{B}\mathbf{e}_d + \hat{\mathbf{\Gamma}}^* - \mathbf{\Gamma}^* \quad (9)$$

Where, if we define (following the steps in [12]):

$$\hat{\mathbf{d}} = \mathbf{F}_1\mathbf{x} + \mathbf{F}_2\dot{\mathbf{x}} + \mathbf{G}_1\hat{\mathbf{x}} + \mathbf{G}_2\dot{\hat{\mathbf{x}}} + \mathbf{G}_3\mathbf{\Gamma}^* \quad (10)$$

then, as $\mathbf{F}_2\mathbf{B} - I \neq 0 \ \forall \mathbf{F}_2$, [12] proposes to take:

$$\mathbf{G}_1 = -(\mathbf{F}_1 + \mathbf{B}^\dagger \mathbf{A})$$
$$\mathbf{G}_2 = -(\mathbf{F}_2 - \mathbf{B}^\dagger)$$
$$\mathbf{G}_3 = -\mathbf{B}^\dagger,$$

thus

$$\hat{\mathbf{d}} = \mathbf{F}_1\mathbf{x} + \mathbf{F}_2\dot{\mathbf{x}} - (\mathbf{F}_1 + \mathbf{B}^\dagger\mathbf{A})\hat{\mathbf{x}} - (\mathbf{F}_2 - \mathbf{B}^\dagger)\dot{\hat{\mathbf{x}}} - \mathbf{B}^\dagger\hat{\mathbf{\Gamma}}^*. \quad (11)$$

From (6), we can isolate $\mathbf{d}$ as $\mathbf{B}$ is full column rank, using its pseudoinverse $\mathbf{B}^\dagger$:

$$\mathbf{d} = \mathbf{B}^\dagger\dot{\mathbf{x}} - \mathbf{B}^\dagger\mathbf{A}\mathbf{x} - \mathbf{B}^\dagger\mathbf{\Gamma}^*, \quad (12)$$

and, with (11) and (12), knowing that, in the case of study, $\mathbf{B}^\dagger\mathbf{A} = \mathbf{0}$:

$$\mathbf{e}_d = \hat{\mathbf{d}} - \mathbf{d} = (\mathbf{B}^\dagger - \mathbf{F}_2)\dot{\mathbf{e}} - \mathbf{F}_1\mathbf{e} + \mathbf{B}^\dagger(\hat{\mathbf{\Gamma}}^* - \mathbf{\Gamma}^*), \quad (13)$$

where $\mathbf{B}^\dagger(\hat{\mathbf{\Gamma}}^* - \mathbf{\Gamma}^*) = \mathbf{n}(\mathbf{q}, \hat{\mathbf{x}}_2) - \hat{\mathbf{n}}(\mathbf{q}, \hat{\mathbf{x}}_2)$ is the error with the learned model of the function $\mathbf{n}$ defined before. This means that, as one could expect, the force estimation error will depend on:

- The model estimation error.
- The estimated joint acceleration error.
- The estimated joint velocity error.

Thus, our objective will be to have a small-as-possible estimation error for the state space, to reduce the force estimation error $\mathbf{e}_d$.

Substituting (13) into (9), the $\mathbf{\Gamma}$'s cancell out and we obtain the position estimation error dynamics equation:

$$(I + \mathbf{B}\mathbf{F}_2 - \mathbf{B}\mathbf{B}^\dagger)\dot{\mathbf{e}} = (\mathbf{A} - \mathbf{K} - \mathbf{B}\mathbf{F}_1)\mathbf{e}.$$

Now, as we intend not to use acceleration measures in Eq. (12), we need $\mathbf{F}_2 = \begin{bmatrix} 0 & 0 \end{bmatrix}$, and with $\mathbf{F}_1 = \begin{bmatrix} 0 & \mathbf{M}(\mathbf{x}_1)\mathbf{\Sigma} \end{bmatrix}$, $\mathbf{\Sigma}$ being another gain, we obtain:

$$\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \dot{\mathbf{e}} = \begin{bmatrix} -\mathbf{K}_{11} & I - \mathbf{K}_{12} \\ -\mathbf{K}_{21} & -\mathbf{\Sigma} - \mathbf{K}_{22} \end{bmatrix} \mathbf{e}, \quad (14)$$

which can be operated to obtain the system:

$$\begin{aligned} \dot{\mathbf{e}}_1 &= -K_{11}\mathbf{e}_1 + (I - K_{12})\mathbf{e}_2 \\ \mathbf{e}_2 &= -(\mathbf{\Sigma} + K_{22})^{-1}K_{21}\mathbf{e}_1 \end{aligned},$$

thus $\mathbf{e}_2$ can be substituted in the first equation to obtain $\mathbf{e}_1$'s dynamics, the dynamics of the position estimation error:

$$\dot{\mathbf{e}}_1 = -\left[K_{11} + (I - K_{12})(\mathbf{\Sigma} + K_{22})^{-1}K_{21}\right]\mathbf{e}_1, \quad (15)$$

which converges for any values of K_{ij} and $\mathbf{\Sigma}$ for which (15)'s matrix in brackets is positive definite.

In addition, we have a dependency between $\mathbf{e}_2$'s dynamics and $\mathbf{e}_1$'s, meaning that if the position estimation converges, so does the velocity estimation. Also, if (14) has an asymptotically stable equilibrium point at $\mathbf{e} = 0$, from (9) we have (at steady state)

$$\mathbf{e}_d = \mathbf{B}^\dagger\left(\hat{\mathbf{\Gamma}}^* - \mathbf{\Gamma}^*\right) = \hat{\mathbf{n}} - \mathbf{n},$$

which is the error of modelling the dynamics.

Moreover, from (14) we have:

$$\begin{aligned} \dot{\hat{\mathbf{x}}}_1 &= \hat{\mathbf{x}}_2 + K_{11}(\mathbf{x}_1 - \hat{\mathbf{x}}_1) + K_{12}(\mathbf{x}_2 - \hat{\mathbf{x}}_2) \\ 0 &= K_{21}(\mathbf{x}_1 - \hat{\mathbf{x}}_1) + (\mathbf{\Sigma} + K_{22})(\mathbf{x}_2 - \hat{\mathbf{x}}_2) \end{aligned},$$

which can be operated to get a linear dynamic equation for $\mathbf{x}_1$, $\mathbf{x}_2$:

$$\begin{aligned} \dot{\hat{\mathbf{x}}}_1 &= \left[K_{11} + (I - K_{12})(\mathbf{\Sigma} + K_{22})^{-1}K_{21}\right](\mathbf{x}_1 - \hat{\mathbf{x}}_1) + \mathbf{x}_2 \\ \hat{\mathbf{x}}_2 &= (\mathbf{\Sigma} + K_{22})^{-1}K_{21}(\mathbf{x}_1 - \hat{\mathbf{x}}_1) + \mathbf{x}_2 \end{aligned},$$

seeing $\mathbf{x}_1$, $\mathbf{x}_2$ as inputs of the observer, this results in a dynamic system on $\hat{\mathbf{x}}_1$, being $\hat{\mathbf{x}}_2$ an output.

Finally, the external torque estimation (using equation (10) and our proposed values) is:

$$\hat{\mathbf{d}} = \mathbf{M}(\hat{\mathbf{x}}_1)\left(\dot{\hat{\mathbf{x}}}_2 + \mathbf{\Sigma}(\mathbf{x}_2 - \hat{\mathbf{x}}_2)\right) + \hat{\mathbf{n}}(\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2) - \mathbf{u}_\mathbf{c}. \quad (16)$$

This latter method is the first approach at applying [12] to a robotic manipulator, but it should be noted that:

- The approximate value of $\mathbf{\Gamma}$ in Eq. (11) would turn into noise on the observer, but as it is cancelled out in (16), it is not supposed to affect the convergence of the state observer, although it indeed includes error in the contact force estimation.
- Criteria on the tuning of $\mathbf{K}$ are given.
- We require the use of the true joint velocities, $\mathbf{x}_2$. This is not a problem, as these can be measured by differentiating joint positions.
- No assumptions on the disturbance behaviour or model are taken, except that $\mathbf{n}$ depends only on position and acceleration variables. Here it must be pointed that most disturbance observers in literature assume the disturbances have a *Lipschitz* behaviour [13], or a known model [14]. Also, no steady-state requirements are needed, although the model may have more error.
- This estimation is independent of the control scheme used. It can be run online and parallel to any controller, even at a different frequency. However, as we will see later, it does become control dependent in certain situations due to unmodelled static friction and other unlearnable effects.

As a conclusion, the estimation of the disturbance, which is the external force, can be done with guarantees of convergence.

IV. Experimentation

To test the observer proposed in Sec. III, we implemented the previous equations on a 7-dof WAM robot. As a control law, we used a computed torque control scheme [7] [15], with $\mathbf{u_c} = \mathbf{M}(\mathbf{x}_1)\dot{\mathbf{x}}_2^\mathbf{d} + \hat{\mathbf{n}}(\mathbf{x}_1^\mathbf{d}, \mathbf{x}_2^\mathbf{d}) + \mathbf{u_{PD}}$, $\hat{\mathbf{n}}(\mathbf{x}_1^\mathbf{d}, \mathbf{x}_2^\mathbf{d})$ being the learned model using the desired trajectory, instead of real measurement, and $\mathbf{u_{PD}}$ a PD control action on the joint state that compensates modelling error and any external force. Using it in Eq. (1) we obtain:

$$\mathbf{M}(\mathbf{x}_1)\dot{\mathbf{x}}_2 + \mathbf{n}(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{M}(\mathbf{x}_1)\dot{\mathbf{x}}_2^\mathbf{d} + \hat{\mathbf{n}}(\mathbf{x}_1^\mathbf{d}, \mathbf{x}_2^\mathbf{d}) + \mathbf{u_{PD}} - \mathbf{u_e},$$

and, isolating $\mathbf{u_e}$, the real disturbance value is:

$$\mathbf{u_e} = \mathbf{u_{PD}} + \mathbf{M}(\mathbf{x}_1)\left(\dot{\mathbf{x}}_2^\mathbf{d} - \dot{\mathbf{x}}_2\right) + \hat{\mathbf{n}}(\mathbf{x}_1^\mathbf{d}, \mathbf{x}_2^\mathbf{d}) - \mathbf{n}(\mathbf{x}_1, \mathbf{x}_2).$$

However, with Eq. (16), we can substitute the control action $\mathbf{u_c}$ to obtain the estimated external perturbation:

$$\begin{aligned}\hat{\mathbf{u}}_\mathbf{e} = -\hat{\mathbf{d}} = \quad & \mathbf{u_{PD}} + \mathbf{M}(\mathbf{x_1})\left(\boldsymbol{\Sigma}(\hat{\mathbf{x}}_\mathbf{2} - \mathbf{x_2}) + \dot{\mathbf{x}}_\mathbf{2}^\mathbf{d} - \dot{\hat{\mathbf{x}}}_\mathbf{2}\right) + \\ & + \hat{\mathbf{n}}(\mathbf{x_1^d}, \mathbf{x_2^d}) - \hat{\mathbf{n}}(\mathbf{x_1}, \hat{\mathbf{x}}_\mathbf{2}),\end{aligned}$$

where the real acceleration $\dot{\mathbf{x}}_\mathbf{2}$ is not used, but the estimated $\dot{\hat{\mathbf{x}}}_\mathbf{2}$ and desired $\dot{\mathbf{x}}_\mathbf{2}^\mathbf{d}$ ones are instead.

This system was discretised, and then the state observer system was run at 500Hz, while the LWPR system run at 50Hz with a zero order hold to attach it to the other 500Hz system. However, although simulation showed excellent results, after implementing the algorithms in a real robot, we found 3 factors that had to be mitigated in order to have better results. They are described in the following three subsections.

A. Noise

The joints position signal presented little noise, but differentiating in order to get the velocity increased noise. Moreover, when derivating $\hat{\mathbf{x}}_\mathbf{2}$ in order to get the estimated acceleration, a very noisy signal was obtained. In order to reduce this noise, which would directly affect the external torque result, we added a Parks-McClellan Filter [10] at the readings of the joint state and velocity. With this filter, the estimated acceleration was less noisy than the obtained by directly differentiating the position readings twice.

B. Friction

The WAM robot is driven by cables in an architecture designed to reduce friction. However, when working with small controller gains and small velocities, the motor cogging and static friction become very evident. Static friction causes unpredicted stationary errors when the robot stops, and motor cogging adds a variable hysthesis on friction that makes model estimation difficult and causes a discontinuous tracking with the lowest inertia joints. As these friction cannot be learned, our work has focused on compensating the error they cause.

C. Error

The residual error, higher than expected, caused by unlearned static friction, results in a large PD action, where it should be small. The PD action, multiplied by the error, may give fake external torques in Eq. (16) if its constants are not small, i.e., large controller gains give larger force estimation errors for the same position error.

To minimize these false torques, we propose to modify the PD controller, to make its gain linearly depend on the error (with upper and lower saturation limits to ensure trajectory tracking and avoid too stiff controllers). This ensures that, when the error is large, a strong action is applied to reduce it, while for a small error, the low controller gain results in a negligible residual torque.

D. Experimental Setup and Results

To evaluate the behaviour of the estimator, we trained a 10 seconds trajectory to a WAM robot, hanging different loads at its end-effector. The loads were of 0, 0.5, 1, 1.5 and 2 kg. To analyse the results, we compared the outputs of the vertical force estimated (F_z) at each trajectory, and the results are shown in Fig. 4, where we see that the estimation has low error, but accumulates slight error for large weights. This is because, as commented, a heavier load results in more error at stationary state, which implies a larger gain of the controller, thus more uncertainty at the estimation.

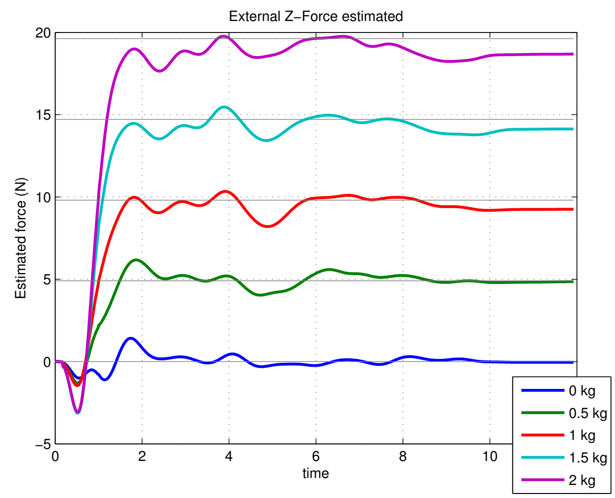

Fig. 4. Experimental results when hanging weights from the robot's end-effector. Horizontal lines represent the real weight. At time=$10s$, the robot ended its trajectory. At time=$5s$, joint 1 changes its direction with a step on the desired acceleration, thus creating a transient in force estimation.

In Fig. 5 we can see the estimated torques along the trajectory, while in Fig. 6 we plot the resulting wrench estimations. There we can observe unexpected peaks due to joint 1 (with the most inertia) changing its direction. Static friction appears in that moment, causing the observed behaviour. This results in an unexpected transient estimated force, that rapidly decreases to zero afterwards. In addition, unmodelled frictions compensate part of the weight, thus the force estimation is slightly lower than the true weights on the end-effector.

The results show that unmodelled friction reduces the precision of our external force estimator. However, despite the uncertainties around unmodelled forces, the results in Fig. 4 and Fig. 6 are accurate, showing the potential use for any robot.

V. Conclusions

Estimating the external force applied on a robot without having an expensive force sensor at its end-effector is a step forward for control and manipulation purposes. For this reason, this is a state-of-the-art topic with a high potential. Some works have good results at simulation, but rely on the availability of analytical models of the robot, the possibility of having the true values of friction or Coriolis forces or they assume almost-stationary situations, meaning the estimations are not available while the robot moves. In addition, their algorithms are tested with simple robots such a 2R manipulator.

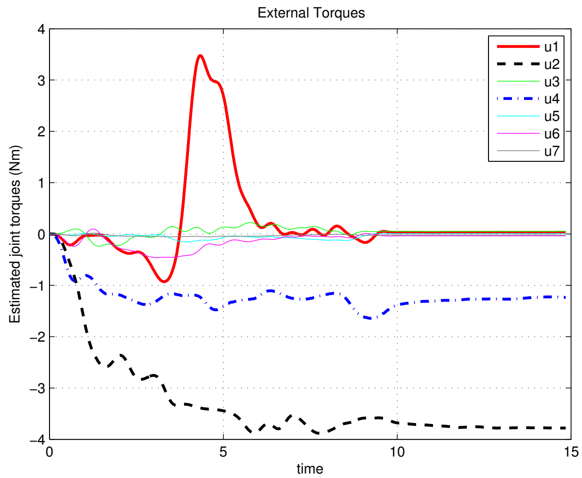

Fig. 5. Estimated external torques for a load of $1kg$. Along the executed trajectory, joints 2 and 4 hold the vertical force. The transient peak on joint at time=5s is due to a step on the desired acceleration along trajectory.

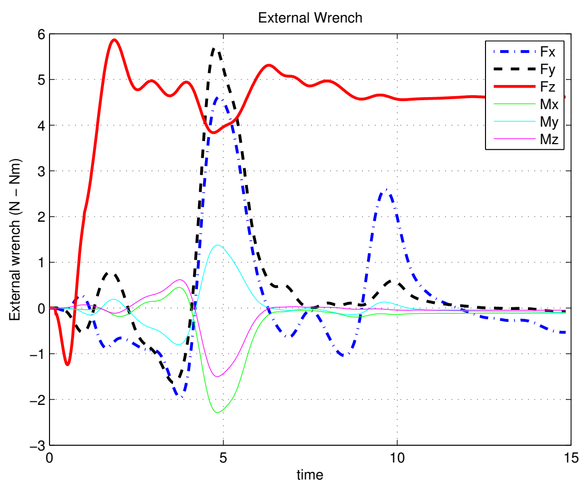

Fig. 6. Estimated external wrenches for a load of $0.5kg$. Acceleration discontinuities when changing direction or stopping cause transient behaviour at time=$5s$ and $10s$.

In this work, we propose an algorithm that, despite its estimation may have a small delay caused by the filters applied and can carry errors due to unknown friction, outperforms the previous works based on state observers, with a rigorous deduction of equations and proof of its convergence, making no assumptions on the external torque applied, nor requiring stationary situations.

The results, in Section IV, with a 7-dof robot capable of performing various manipulation tasks, show that good force estimations can be obtained while the robot is still in motion. These vertical force results are being used to know how many cloth garments have been picked by a robot. Another advantage of this proposal is that it can be implemented on any control scheme (see Fig. 2).

As future work, some research to reduce the effects of uncertainties on the dynamic behaviour of the robot, which generate error, needs to be carried out. Other possible improvements are to globalize the inverse model to the whole workspace, instead of tailoring it to trajectories, and to optimize the robot controller in order to reduce the residual torques observed, or model friction apart so as to gain precision.

REFERENCES

[1] S. Haddadin, A. Albu-Schaffer, G. Hirzinger, "Safety evaluation of physical human-robot interaction via crash-testing". *Robotics: Science and Systems Conference*, pp. 217-224, 2007.

[2] T. Tamei, T. Shibatab, "Fast Reinforcement Learning for Three-Dimensional Kinetic Human-Robot interaction via crash-testing". *Advanced Robotics*. vol 25, issue 25, 2011.

[3] W.T. Townsend, J.K. Salisbury, P. Dario, G. Sandini, P. Aebischer, "Mechanical Design for Whole-Arm Manipulation",*Robots and Biological Systems: Towards a New Bionics?*. NATO ASI Series, Springer Berlin Heidelberg, pp. 153-164.

[4] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo. "Modelling, Planning and Control." *Advanced Textbooks in Control and Signal Processing*. Springer, 1st edition, 2009.

[5] M. Van Damme, P. Beyl, B. Vanderborght, V. Grosu, R. Van Ham, I. Vanderniepen, A. Matthys, D. Lefeber, "Estimating Robot End-Effector Force from Noisy Actuator Torque Measurements". *IEEE International Conference on Robotics and Automation*, pp. 1108-1113, 2011.

[6] A. Alcocer, A. Robertsson, A. Valera, R. Johansson, "Force Estimation and Control in Robot Manipulators". *Robot control 2003*. pp. 31-36 2004.

[7] D. Nguyen-Tuong, J. Peters, "Learning Robot Dynamics for Computed Torque Control Using Local Gaussian Processes Regression". *ECSIS Symposium on Learning and Adaptive Behaviors for Robotic Systems*, pp 59-64, 2008.

[8] S. Klanke, S. Vijayakumar and S. Schaal, "A library for Locally Weighted Projection Regression".*Journal of Machine Learning Research*, vol. 9, pp. 623-626, 2008.

[9] S. Klanke, S. Viyajakumar, S. Schaal, "A Library for Locally Weighted Projection Regression" url:http://jmlr.csail.mit.edu/papers/v9/klanke08a.html.

[10] L.R. Rabiner, J.H. McClellan, T.W. Parks, "FIR digital filter design techniques using weighted Chebyshev approximation" *IEEE Proceedings*. vol 63, issue 4, pp.595-610, 1975.

[11] P.J. Hacksel, S.E. Salcudean, "Estimation of Environment Forces and Rigid-Body Velocities using Observers".*IEEE International Conference on Robotics and Automation*, vol. 2, pp. 931-936, 1994.

[12] Chia-Shan Liu, Huei Peng, "Inverse-Dynamics Based State and Disturbance Observers for Linear Time-Invariant Systems".*Journal of Dynamic Systems, Measurement, and Control*. vol. 124 pp. 375-381, 2002.

[13] W.H. Chen, L. Guo, "Analysis of Disturbance Observer Based Control for Nonlinear Systems under Disturbances with Bounded Variation".*International Conference on Control*, pp.1-5, 2004.

[14] W.H. Chen: "Disturbance Observer Based Control for Nonlinear Systems." *IEEE/ASME Transactions on Mechatronics*, vol. 9, issue 4, pp. 706-710, 2004.

[15] Duy Nguyen-Tuong, M. Seeger, J. Peters, "Computed torque control with nonparametric regression models". *American Control Conference*, pp. 212-217, 2008.