



# LUND UNIVERSITY

## Sensorless Friction-Compensated Passive Lead-Through Programming for Industrial Robots

Stolt, Andreas; Bagge Carlson, Fredrik; Ghazaei, Mahdi; Lundberg, Ivan; Robertsson, Anders; Johansson, Rolf

*Published in:*  
2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)

*DOI:*  
[10.1109/IROS.2015.7353870](https://doi.org/10.1109/IROS.2015.7353870)

2015

*Document Version:*  
Peer reviewed version (aka post-print)

[Link to publication](#)

*Citation for published version (APA):*  
Stolt, A., Bagge Carlson, F., Ghazaei, M., Lundberg, I., Robertsson, A., & Johansson, R. (2015). Sensorless Friction-Compensated Passive Lead-Through Programming for Industrial Robots. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 3530-3537). IEEE - Institute of Electrical and Electronics Engineers Inc.. <https://doi.org/10.1109/IROS.2015.7353870>

*Total number of authors:*  
6

### General rights

Unless other specific re-use rights are stated the following general rights apply:  
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Read more about Creative commons licenses: <https://creativecommons.org/licenses/>

### Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LUND UNIVERSITY

PO Box 117  
221 00 Lund  
+46 46-222 00 00

# Sensorless Friction-Compensated Passive Lead-Through Programming for Industrial Robots

Andreas Stolt<sup>1</sup>, Fredrik Bagge Carlson<sup>1</sup>, M. Mahdi Ghazaei Ardakani<sup>1</sup>, Ivan Lundberg<sup>2</sup>,  
Anders Robertsson<sup>1</sup>, Rolf Johansson<sup>1</sup>

**Abstract**—Industrial robots are important when the degree of automation in industry is increased. To enable the use of robots also when the products change rapidly, the programming must be quick and easy to perform. One way to accomplish this is to use lead-through programming, i.e., the user manually guides the robot. This paper presents a sensorless approach, and thus avoids the need for a typically expensive sensor. The method is based on disabling low-level joint controllers combined with gravity compensation. It is reported how the performance can be improved by compensating for friction. Further, a method for detecting small external torques is described, based on the use of the low-level joint controllers with increased integral gain. The lead-through programming is experimentally evaluated using two different industrial robots.

## I. INTRODUCTION

Industrial robots have become indispensable in many places in industry today, such as the automotive industry. They have relieved human workers from hazardous and/or repetitive and monotone tasks, and they have increased the productivity and quality of the manufactured products due to their high speed and precision. The robots are usually placed in structured environments that are supposed to remain the same for a long time. This makes it worthwhile to put the required effort into performing the robot programming, which usually takes long time.

In other parts of the manufacturing industry, it is much more common with short-series production. For robots to be competitive here, the teaching phase must be quick and easy to perform, to minimize the down-time. One easy way to accomplish a straightforward teaching method is to manually guide the robot, which is usually called lead-through programming (LTP), or walk-through. This makes it possible for the robot to both learn positions and trajectories. It becomes especially convenient for the operator as no consideration of different coordinate frames etc. is needed. Lead-through programming for industrial robots is usually implemented by using force sensors. They are, however, often very expensive, and it would be preferable if LTP could be accomplished in a sensorless setting.

A survey of different methods for programming industrial robots was presented in [1]. It was concluded that although online programming, i.e., using the teach pendant to manually move the robot between positions to be used in the

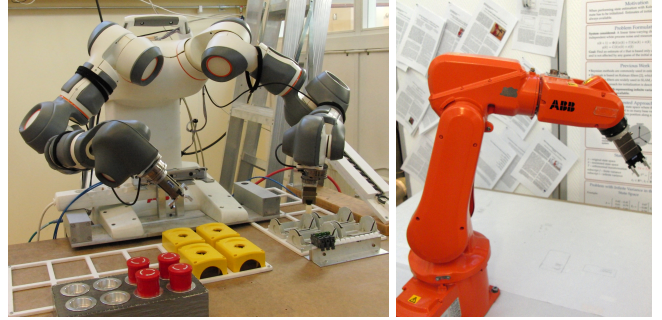


Fig. 1. The robots used in experiments. Left: ABB YuMi. Right: ABB IRB120

program, has several drawbacks, it is widely used. Further, several methods for incorporating sensors to simplify the online programming phase are described, e.g., using force sensors or vision systems. An application of LTP is presented in [2], where LTP was used to simplify the teaching of weld paths in a shipyard. To accomplish the LTP, the robot was equipped with a force/torque sensor.

Programming by demonstration is a field where LTP can be used for performing the demonstration, which sometimes is called kinesthetic teaching. One such example is presented in [3], where the authors focus on how to treat redundancy during kinesthetic teaching. The experiments were performed with a seven-degree-of-freedom robot, and a user study showed that it was beneficial to assist the operator by controlling the null-space of the robot, according to a redundancy resolution that was trained during an initialization phase. Another example of kinesthetic teaching is presented in [4], where skills were taught to a robot in two steps. First, motion sensors were used to record the demonstrated task. Then the robot tried to perform the task and the teacher could interact through kinesthetic teaching, which was accomplished by choosing which motors of the robot that should become passive.

One standard approach for performing sensorless force control has been based on using dynamical models of the robot together with the measured joint positions, where the applied force can be estimated with disturbance observers [5], [6]. Another approach is based on using the motor torques to estimate the applied external torques; two such examples are [7], [8]. An approach for detection of external torques without external sensing using the generalized momentum is [9], where the method relies on knowledge of a dynamical model of the robot. A method to decrease

<sup>1</sup>Andreas Stolt, andreas.stolt@control.lth.se, Fredrik Bagge Carlson, M. Mahdi Ghazaei Ardakani, Anders Robertsson and Rolf Johansson are with the Department of Automatic Control, Lund University, Sweden. The authors are members of the LCCC Linnaeus Center and the eLLIT Excellence Center at Lund University.

<sup>2</sup>Ivan Lundberg is with ABB Robotics, Västerås, Sweden.

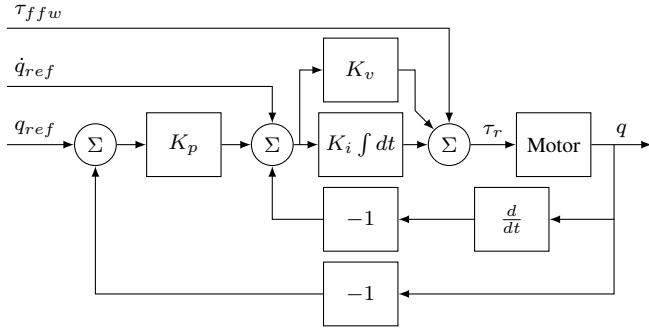


Fig. 2. Schematic block diagram of the low-level joint controller running at 2 kHz.

the low velocity friction uncertainties using dithering when doing force estimation was presented in [10].

This paper will present a method for sensorless LTP. It is based on disabling the low-level servo controllers in the joints and only feedforward the torques to balance gravity, i.e., the robot is in a passive mode with no position- or force-feedback loops running. This makes the interaction between the robot and any environment stable. It is further described how the LTP performance can be improved by adding friction compensation. Experimental results from implementations on the ABB YuMi and ABB IRB120 are also presented, see Fig. 1 for the experimental setups.

## II. METHOD

### A. Robot controller structure

The robot system considered in this paper has a control structure such that each joint is individually controlled, with a main computer that calculates references for each of the joints. A cascaded control structure is used for each joint, see Fig. 2 for a block diagram. The outer position loop has proportional feedback, while the inner velocity loop has both proportional and integral feedback. The controller parameters are the three gains  $K_p$ ,  $K_v$ , and  $K_i$ . There is further a current control loop executed in the robot controller. The current control loop ensures that the motor actuates the desired torque.

The research interface available to the robots [11], [12] makes it possible to alter the signals sent from the main computer, i.e., one can send position and velocity references and a torque feedforward signal. Whereas it is possible to modify the control gains ( $K_p$ ,  $K_v$ ,  $K_i$ ), it is not possible to change the controller structure. Available measurements include joint positions and velocities, and the torque reference sent to the motors, which will be close to the actual torque exerted by the motors as the current loop is tightly controlled. The low-level control loops run with a sampling frequency of 2 kHz, while the research interface for setting the references and reading measurements runs at 250 Hz.

### B. Passive lead-through programming

Lead-through programming of the robot was accomplished by disabling the low-level joint control loops, i.e., setting the control gains  $K_p$ ,  $K_v$ , and  $K_i$  to zero. To prevent the robot

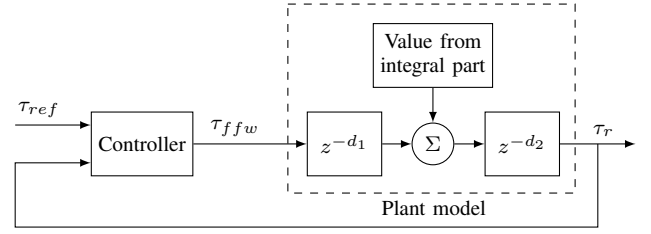


Fig. 3. Block diagram of the torque feedforward control loop, running at 250 Hz. The delays  $d_1$  and  $d_2$  are unknown, but it is known that  $d_1 + d_2 = 3$ .

from falling due to gravity forces, the torque feedforward signal was used to apply the torque needed to counteract gravity. As the joint torques commanded were purely based on feedforward, external forces applied to the robot could lead to movements, if they exceeded the friction forces in the joints. The friction forces, mainly Coulomb and viscous friction, were helpful, as they made sure that the robot did not move when no external forces were applied. The LTP worked in the same way as releasing the brakes of the robot, while maintaining feedforward gravity compensation. As no feedback loops were active, interaction with the environment became stable without the need of tuning any control parameters.

Note that the LTP was implemented independently for each joint, i.e., the implementation was made in joint space.

### C. Gravity compensation

The rigid body dynamic model of a robot is linear in the parameters [13]. The values of these parameters for the robots used in experiments were, however, not known a priori and therefore needed to be identified. Identifying all of the dynamic parameters is difficult, as the experiments to be performed must be chosen with care to be sufficiently exciting to make all parameters identifiable. For the intended application, namely lead-through programming, the robot is expected to move relatively slowly with low accelerations, which means that the dynamic torques will be small, and they were therefore neglected by setting all velocities and accelerations to zero when deriving the equations. The remainder of the dynamics model was therefore only related to gravity, with four parameters for each link of the robot. A simple friction model was also added

$$\tau_{fric} = \theta_C \text{sign}(\dot{q}) + \theta_v \dot{q} \quad (1)$$

where  $\dot{q}$  is the joint velocity,  $\theta_C$  is the Coulomb friction, and  $\theta_v$  the viscous friction parameter. Each link had in total six parameters, and as the model was linear in the parameters, they could be estimated using the least-squares procedure with data from an experiment where the robot slowly moved around in its workspace.

### D. Torque feedforward control

The gravity compensation torque was sent as a torque feedforward signal to each of the low-level joint control loops. Due to the fact that the control gain  $K_i$  is situated

before the integral in Fig. 2, setting it to zero will just stop the update of the integral state, but it will hold its value. This means that the feedforward torque signal will have to compensate for this. To be certain that the desired torque is actuated, a combined feedforward and feedback strategy was applied, see a block diagram of the control loop in Fig. 3. The reference signal, denoted by  $\tau_{ref}$ , is the desired torque, i.e., the torque due to gravity, the measurement signal is the torque reference sent to the motor,  $\tau_r$ , and the control signal is the feedforward signal,  $\tau_{ffw}$ .

In the research interface used, there are delays both when references are set and when measurements are received. These individual delays are unknown, whereas their sum was estimated to be three sampling periods. A model with these properties are given as the plant model in Fig. 3. To account for the delays, the following modified measurement signal was used

$$y_{mod}(t) = \tau_r(t) + \sum_{k=1}^2 \Delta u(t - kh) \quad (2)$$

where  $\Delta u(t)$  denotes the the update of the control signal at time  $t$ , and  $h$  the sampling period. Using  $y_{mod}(t)$  for feedback is the same as using a Smith predictor [14], where the process model is a time delay. An integral controller was used to close the loop, i.e., the torque feedforward signal was calculated as

$$\tau_{ffw}(t) = \tau_{ffw}(t - h) + \Delta u(t) \quad (3)$$

where  $\Delta u(t)$ , the update of the control signal, was

$$\Delta u(t) = \underbrace{(\tau_{ref}(t) - \tau_{ref}(t - h))}_{\text{feedforward}} + \underbrace{K(\tau_{ref}(t) - y_{mod}(t))}_{\text{feedback}} \quad (4)$$

where  $K$  is the feedback gain of the controller, and  $\tau_{ref} = \tau_G$ , i.e., the reference is the torque due to gravity,  $\tau_G$ . The feedforward part handles most of the reference changes, and the feedback part handles disturbances, i.e., the value held in the integral part as shown in Fig. 3. Note that this control loop is active only when the low-level control loop (Fig. 2) is inactive, and vice versa.

#### E. Friction compensated passive lead-through programming

Friction torques in the joints are helpful as they prevent the robot from moving except for when external torques are applied. At the same time, the friction torques make it heavy to move the robot, especially if the friction torques are large. To make it easier to move the robot, additional torque feedforward can be used to compensate for friction based on movement of the robot. Ideally, it should be possible to use the torque due to gravity plus the Coulomb friction torque. Then the robot would be truly free-floating. In reality, neither the torques due to gravity nor the Coulomb friction will be exactly known, and to feedforward the entire Coulomb friction estimate might result in an accelerating joint without any external torque applied. There should still exist some friction that stops the robot when no external forces are

present and the amount depends on the viscous friction level and the quality of the Coulomb friction estimate.

The velocity measurement in the robot is based on numerical differentiation of the position measurement, and consequently the standard deviation of the noise of the speed signal will be a factor  $1/h$  larger than the position measurement noise, where the sampling period  $h = 0.004$  s in our case. The friction compensation is therefore added once the velocity exceeds a threshold, which must be chosen with respect to the noise level. To make a smooth transition when the torque feedforward is added, the amount of friction compensation is made proportional to the velocity for small velocities. Specifically, the friction compensation torque,  $\tau_{FC}$ , was calculated as

$$\tau_{FC} = \begin{cases} 0 & , \quad |\dot{q}| < \dot{q}_0 \\ \frac{|\dot{q}| - \dot{q}_0}{\dot{q}_1 - \dot{q}_0} \text{sign}(\dot{q}) a \hat{\theta}_C & , \quad \dot{q}_0 \leq |\dot{q}| < \dot{q}_1 \\ \text{sign}(\dot{q}) a \hat{\theta}_C & , \quad |\dot{q}| \geq \dot{q}_1 \end{cases} \quad (5)$$

where  $\dot{q}_0$  and  $\dot{q}_1$  are the velocity thresholds defining the proportional region, and  $a$  is the percentage of the Coulomb friction estimate  $\hat{\theta}_C$  that is used as friction compensation. The velocity may increase fast, and the friction compensation will then almost be in the form of a step, which could be unpleasant for the operator since it would feel like an abrupt change in the resistance of the joint in question. This possibility was avoided by also limiting the rate of change of  $\tau_{FC}$ .

The feedback control loop for the torque feedforward signal generation (Sec. II-D) is useful now, as the friction compensation torque can be handled by just adding it to the reference, i.e., use  $\tau_{ref} = \tau_G + \tau_{FC}$  in Eqs. (3)-(4).

#### F. Increased sensitivity to external torques when a joint is not moving

A major difficulty with sensorless LTP is that the Coulomb friction (or stiction) must be overcome to start moving the robot. This effect can be experienced both when the LTP is started from rest, and when one wants to move the end-effector of the robot while keeping the orientation fixed. In the latter scenario, many joints will have to move simultaneously, and it will take quite some force to overcome the friction in the joints where the lever arm from the end-effector is short.

An approach to achieve increased sensitivity to external torques is to activate the low-level joint controller when the joint is not moving, and significantly increase the integral gain (100 times the nominal value was used in the experiments). Then it becomes possible to detect external torques that are significantly smaller than the friction band. Whereas the reason why this is possible is not fully determined, a hypothesis is that the effect is coupled with the fact that the investigated robots have harmonic drive gears, as the effect has not been observed for robots with other types of gears. The large integral gain increases the bandwidth of the control loop. The hypothesis is that the increased bandwidth together with the measurement noise functions as a dithering signal,

which would mean that the motor is constantly moving and thereby has a significantly lower resulting friction torque.

The idea is to activate the controller with high integral gain only when a joint is not moving, and when an external torque is detected it should be turned off again. If the detected torque is small, it will not overcome the friction torque. Therefore, a short torque feedforward pulse is commanded, and if the operator really intended to move the joint, i.e., keeps applying a force, the pulse will help the joint start moving. Once the joint is moving, the friction compensation, according to (5), will be active and help the operator.

The detection of external torques is performed by using detection thresholds. The torque measurement,  $\tau_r$  in Fig. 2, might end up in other places than in the middle of the friction band when the controller with high integral gain is activated. To account for this, the thresholds were centered around a delayed and filtered version of  $\tau_r$ . To handle the initialization phase, when there was uncertainty of where inside the friction band  $\tau_r$  would end up, the thresholds were ramped down from the Coulomb friction estimates to the final thresholds. The upper threshold,  $\Lambda_{up}$ , was calculated as

$$\Lambda_{up}(t) = \begin{cases} \hat{\tau}_{center} + \frac{t\lambda + (T-t)(\hat{\tau}_G + \hat{\theta}_C)}{T} & , \quad t < T \\ \hat{\tau}_{center} + \lambda & , \quad t \geq T \end{cases} \quad (6)$$

where the time  $t$  is assumed to be zero when the initialization starts,  $T$  is the length of the initialization phase,  $\hat{\tau}_{center}$  is a low-pass filtered and delayed version of  $\tau_r$ ,  $\lambda$  is the final threshold,  $\hat{\tau}_G$  the estimated gravity torque, and  $\hat{\theta}_C$  the estimated Coulomb friction. The lower threshold was calculated analogously.

#### G. Small movements

A LTP based on the features described earlier in this section works very well for large movements. However, if the operator is interested in doing small adjustments of the end-effector, e.g., learning a position for gripping an object, the friction compensation scheme becomes counterproductive. An attempt to move the end-effector a small distance might result in a larger movement when the friction compensation torques are activated. The reason is that at first, the operator applies a force to start the movement, and as this force is sensed, extra help in the form of friction compensation is activated, and then the applied force is too large and the movement becomes larger than intended.

For small movements, it was found out in experiments that it is in practice better to disable all friction compensation. This will make it harder to move the robot, as the operator will have to overcome all friction forces. On the other hand, the constant friction resistance makes the friction predictable and manageable.

The LTP implementation will have to switch between friction compensation on and off to work well for both large and small movements. One way to do this is to investigate the maximum distance moved by the end-effector during a fixed time window. Small movements will almost always be fine adjustments of the end-effector, e.g., fine tuning a gripping

position, and the end-effector movement is therefore the relevant measure to use. To be precise, the measure used for determining when to switch between friction compensation on and off is defined as

$$\text{maxDist} = \max_{t_0 \in [t-\Delta t; t]} \|p(t) - p(t_0)\|_2 \quad (7)$$

where  $p(t)$  denotes the Cartesian position of the end-effector at time  $t$ , and  $\Delta t$  is the time window used. To make the transition smooth, a linear region where the amount of friction compensation is proportional to the value of maxDist was also introduced, i.e., using the friction compensation torque  $\tau_{FC}^{new} = b\tau_{FC}$ , where  $\tau_{FC}$  is defined in (5) and the factor  $b$  as

$$b = \begin{cases} 0 & , \quad \text{maxDist} < d_1 \\ \frac{d_2 - \text{maxDist}}{d_2 - d_1} & , \quad d_1 \leq \text{maxDist} < d_2 \\ 1 & , \quad \text{maxDist} \geq d_2 \end{cases} \quad (8)$$

where  $d_1$  and  $d_2$  defines the linear region.

### III. IMPLEMENTATION

Two robots were used for experiments, see photos in Fig. 1. The first one was the ABB YuMi [15] (previously known as FRIDA). It is a dual-arm manipulator, where each of the arms has 7 joints. The robot is made with light-weight materials, and it is safe to be used in proximity to humans, due to power and speed limitations and soft paddings to cover all sharp edges. The other robot was the ABB IRB120 [16]. It is a small, traditional industrial robot with 6 joints. Both robots were controlled with the ABB IRC5 control system. The YuMi robot was equipped with wrist-mounted ATI Mini 40, 6 degree-of-freedom force/torque sensors, which made it possible to collect validation data.

The LTP was implemented in two different versions. The first one was such that the operator teaches a number of positions, and when the taught program is being replayed, the trajectory between the positions are planned by the native robot controller. The other demonstrator records the actual LTP trajectory, by saving positions with a frequency of 10 Hz. When the learned program is being replayed, the native controller performs joint moves between the recorded positions, with possibilities to both increase and decrease the velocity of the motion.

### IV. EXPERIMENTAL RESULTS

The experiments described in this section were made with the ABB YuMi, unless otherwise stated.

#### A. Parameters

The parameters for the gravity compensation and for the friction model was estimated from an experiment where the robot slowly moved around in its workspace without any interaction with the environment. As the friction model (1) only is valid for velocities different from zero, data from the experiment where the velocity was close to zero were excluded when performing the parameter identification. The resulting compensation gave a mean absolute error ranging

from 0.3 Nm for the base joints to 0.03 Nm for the wrist joints.

The friction compensation parameters in (5) were tuned manually. The lower velocity level,  $\dot{q}_0$ , was chosen such that the noise in the velocity measurement did not trigger any torque feedforward, and the upper level,  $\dot{q}_1$ , was chosen such that the transition from no torque feedforward to full torque feedforward felt smooth when manually guiding the robot. The parameter  $a$ , the percentage of the estimated Coulomb friction to use as feedforward, was chosen as high as possible without getting any drifting joints. This magnitude depended on the accuracy of the gravity compensation and the estimated Coulomb friction, for the YuMi robot 60–80 % of the estimated Coulomb friction was used.

The parameters for switching the friction compensation on and off described in Sec. II-G were chosen as  $\Delta t = 0.5$  s, and the start of the proportional region as  $d_1 = 2$  cm, and full friction compensation was active at  $d_2 = 5$  cm.

The other parameters used are described in the following subsections.

### B. The use of large integral gain

The integral gain,  $K_i$ , was increased when the velocity became low. To get rid of the noise in the velocity measurement, it was low-pass filtered, such that a low threshold could be used. Filtering, however, delays detection of the start of a joint movement, and therefore also the unfiltered velocity measurement was thresholded, but with a significantly higher threshold. To increase  $K_i$ , the velocity had to be below both of the above mentioned thresholds. Further, to get some robustness towards when the velocity changes sign, the velocity had to be below the thresholds for 0.1 seconds for the controller to be activated. The controller with large integral gain resulted in a stable system as long as the joint remained at rest. At rare occasions, though, the noise triggered a motion that led to instability of the system. Therefore, the velocity was supervised in this phase, and the large integral gain was turned off if the velocity exceeded any of the thresholds previously mentioned. When  $K_i$  was increased, the other controller parameters,  $K_p$  and  $K_v$ , had their nominal values.

The benefit of using the controller with a large integral gain is displayed in Fig. 4. It shows an experiment where forces were applied to the end-effector of the robot. The experiment was carried out twice; the left subplots show the case with nominal controller values and the right subplots the case where  $K_i$  is 100 times larger than its nominal value. The upper plots show the response in the first joint of the robot, and the lower plots the response for the sixth joint. A wrist-mounted force/torque sensor was used to give the validation measurement. The torque due to gravity has been compensated for in the plots. It can clearly be seen that using the large integral gain is beneficial, as external torques within the estimated Coulomb friction band are clearly visible in the raw torque data, i.e., the signal denoted  $\tau_r$  in Fig. 2. Without the use of high integral gain (the left subplots), the raw torque is unpredictable with drifting curves, and it would be very

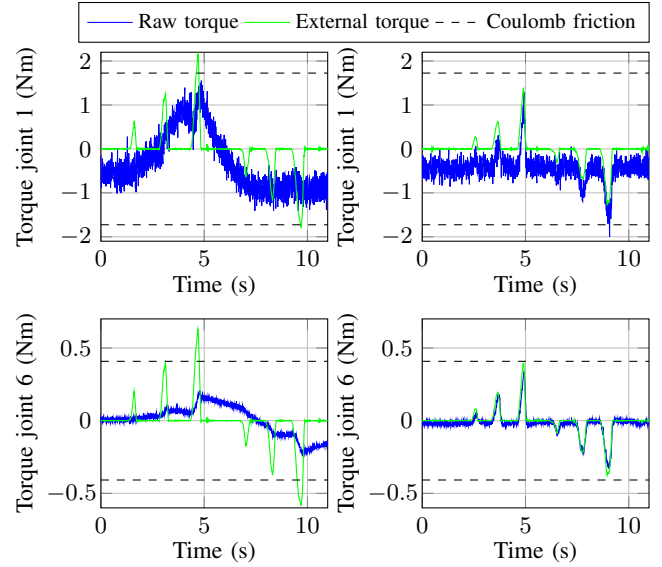


Fig. 4. Experiment where forces were applied to the end-effector of YuMi. Results from two joints are shown; the top row shows the result for the first joint, and the bottom row shows the results for the sixth joint. The left subplots show the raw torque response for the nominal controller parameters and the right subplots the response for the case with high integral gain. The external torque has been measured with a wrist-mounted force/torque sensor.

difficult to detect the applied forces without the validation data. It can further be noted that the noise level is much lower for the sixth joint, and also the other wrist joints, making it possible to detect lower torques for these joints. The reason for this is probably that the base joints support a larger part of the robot structure, with more noise due to mechanical resonances.

An experiment that illustrates the behavior of the detection thresholds used for detecting external torques is displayed in Fig. 5, where the thresholds are displayed in red and the Coulomb friction estimate in the dashed black lines. In the beginning of the experiment, the joint was moving and a torque feedforward of 65% of the friction band was applied, which can be seen in the raw torque curve (blue). The applied torque can also be seen in the external torque signal (validation data from a force sensor). When the external torque disappeared (at  $t = 0.4$  s), the joint stopped moving and the torque feedforward was stopped as well. After the velocity had been below the thresholds earlier mentioned, the joint controller with the high integral gain was activated, as indicated in the top of the plot. The raw torque shows a slight positive drift, which was captured by the thresholds. At  $t = 2.7$  s, an external torque appeared. The motor first counteracted the external torque, i.e., tried to keep the position of the joint, but when the detection was made, a helping torque to compensate for friction was sent as feedforward. Initially, the feedforward was in the form of a pulse with a length of 0.2 s, as described in Sec. II-F. The length of the pulse was chosen such that the robot had time to start moving in case the operator would keep applying a force, i.e., such that the velocity based friction compensation



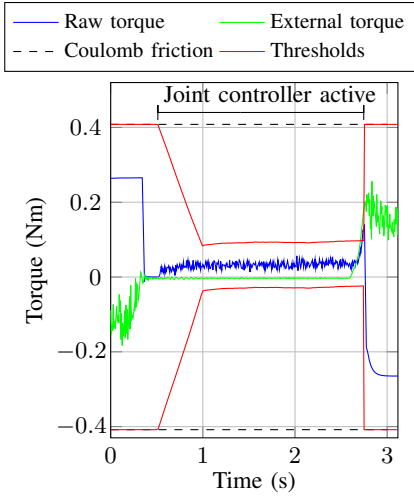


Fig. 5. An illustration of the behavior of the detection thresholds used when the joint controllers with high integral gain were active. The experiment was performed with the sixth joint of YuMi.

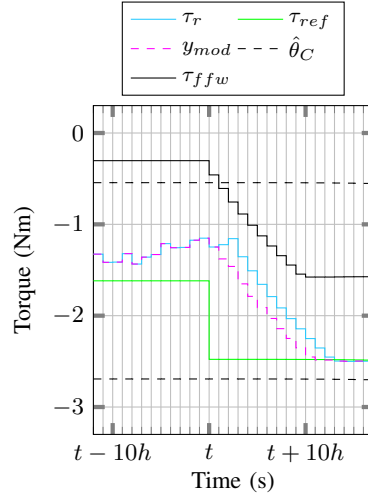


Fig. 6. Experiment that illustrates the behavior of the feedforward control loop for one joint (the third joint of YuMi). An external torque that generated a feedforward torque was detected at time  $t$ . The sample period  $h = 4$  ms.

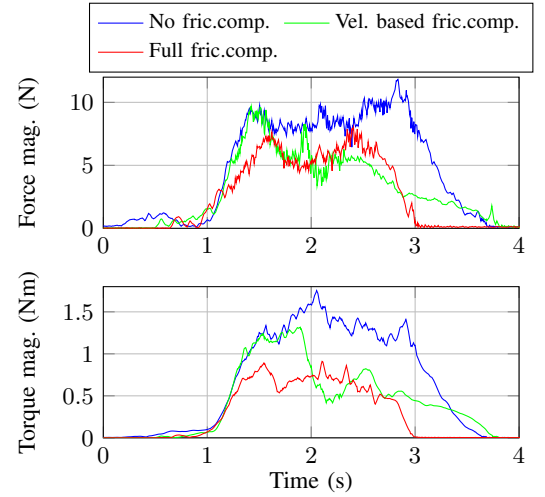


Fig. 7. Results from an experiment where the end-effector of YuMi was moved linearly while trying to keep the orientation fixed. The top diagram shows the measured force magnitude from the wrist-mounted force/torque sensor, and the bottom plot shows the torque magnitude. The experiment was performed three times; first with no friction compensation, then with friction compensation based on velocity only according to (5), and finally with the full friction compensation.

(5) was activated. In Fig. 5, this is exactly what happens. The amplitude of the pulse was chosen to be the same as the velocity based friction compensation, which explains why the pulse can not be separated from the velocity-based friction compensation. When the detection of the external torque was made, the joint controller was deactivated and the thresholds were reset to the Coulomb friction estimate.

The parameters used for the detection thresholds (6) were chosen such that the initialization phase was  $T = 0.5$  s, and the final threshold  $\lambda$  somewhat larger than the noise level as can be seen in Fig. 5 between  $t = 1$  s and  $t = 2.7$  s. The center of the threshold levels was taken as the 40 samples (0.16 s) delayed mean of 80 samples (0.32 s) of the raw torque signal.

### C. Torque feedforward control

An example of the torque feedforward control performance is displayed in Fig. 6, which displays the behavior of the third joint of YuMi when an external torque was detected and a friction compensation torque was commanded. In the beginning of the experiment, for times less than  $t$ , the joint controller with high integral gain was active, i.e., the torque feedforward controller was inactive. At time  $t$ , an external force was detected and a torque feedforward of 80% of the estimated friction band was commanded, as can be seen in the reference signal  $\tau_{ref}$ . As was described in Sec. II-D, the feedforward part of the controller handles most of the set-point change, as can be seen in the fast response in  $\tau_{ffw}$ , the control signal. The limited rate-of-change of  $\tau_{ffw}$  can also clearly be seen, as  $\tau_{ffw}$  was ramped down instead of being changed in a step. The velocity increased fast in this experiment, and the region where the friction compensation torque was proportional to the velocity as defined in (5) is

not visible. By comparing  $\tau_r$  and  $\tau_{ffw}$ , the delay can be noticed, and it can further be seen that  $y_{mod}$  works as an approximation of  $\tau_r$  without the delay.

The feedback component of the controller takes care of the deviation of  $\tau_r$  from the reference, while the feedforward part handles the reference change. The feedback gain,  $K$  in (4), was manually tuned such that the controller was fast but without getting any overshoot when the controller was activated, e.g., which happened at time  $t$  in Fig. 6.

### D. Lead-through programming performance

Experiments were carried out to investigate the LTP performance. The first experiment was performed to show how much external torque was required to start moving a joint by manually applying an increasing force until the investigated joint started to move. The experiment was carried out 30 times without friction compensation, and another 30 times with friction compensation. Two different joints were investigated, the first joint to represent the base joints, and the sixth joint to represent the wrist joints. For the first joint, on average only 55% of the applied torque was needed when friction compensation was active as compared to when it was not. The standard deviation was 9 percentage points. For the sixth joint, the corresponding result was that only 40% of the torque was needed, with a standard deviation of 15 percentage points. The main reason for the difference in benefit was the larger noise level for the base joints. When moving the end-effector, however, the lever arm is longer for the base joints, and this compensates the fact that larger torques are needed.

Another experiment analyzed friction compensation for linear end-effector motion. The experiment was performed by manually moving the end-effector while trying to keep the

orientation fixed. Results from this experiment are displayed in Fig. 7, which shows the force and torque magnitude from the wrist-mounted force/torque sensor. The experiment was performed three times; first with no friction compensation, then with friction compensation based on the measured velocity, i.e., according to (5), and finally with the full friction compensation, where also the method with large integral gain was used. Without friction compensation, the largest applied force and torque was needed, as can be seen in the diagrams. Using friction compensation based on measured velocity initially required the same force and torque to start the robot movement, but after the initial transient when the friction compensation torques were applied, lower external forces and torques were needed. For the full friction compensation, moving the robot was easier, which can be seen in Fig. 7 as lower applied force and torque. Especially, the lower applied torque feels pleasant when moving the robot.

The LTP was compared to using the joystick on the teach pendant to teach a simple task, namely to pick an object. Three positions needed to be taught; a position above the object to pick, a position where the gripper can close around the object, and a position where the robot can safely move away to with the object. Using LTP, teaching these three positions took 25 s, but using the teach pendant took over 2 minutes. The difficult part using the teach pendant was to move the robot to the correct orientation to be able to pick the object. Using LTP can thus substantially decrease the amount of time needed for teaching a robot program.

#### E. Lead-through programming of IRB120

The LTP was also implemented on the ABB IRB120, which has significantly more friction than YuMi. The gravity compensation resulted in a mean absolute error that was 1 Nm for the base joints and 0.1 Nm for the wrist joints. In contrast to YuMi, the IRB120 has a significant amount of viscous friction in the joints, and this makes it possible to feedforward a larger amount of the Coulomb friction torque, and 80–100 % of the estimated Coulomb friction was used for feedforward. Unfortunately, the IRB120 was not equipped with a force sensor and, therefore, no validation data are available. An experiment was performed to show that the benefit of using high integral gain was valid also for this robot, see Fig. 8. Forces were manually applied to the end-effector; the left subplots show the result for the nominal controller gains and the right subplots for when high integral gain was used. The qualitative behavior is the same as the experiment performed with YuMi presented in Fig. 4, but without validation it is difficult to say more.

The LTP was implemented in the same way as for YuMi, and it works well joint by joint. Moving the robot in Cartesian directions becomes quite hard, as a lot of force is required to move all the joints that need to move, also without the strategy for turning off the friction compensation for small movements described in Sec. II-G. The main reason for this was the viscous friction. As compared to the YuMi implementation, the LTP with IRB120 is not as good, but it would still be useful for teaching robot programs.

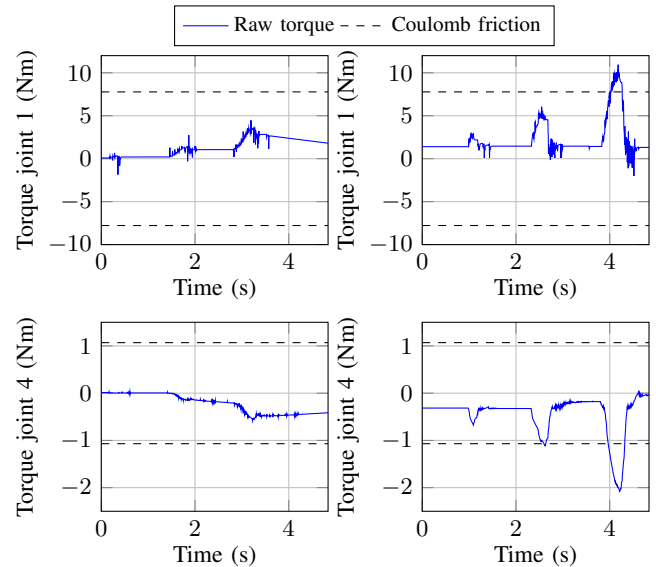


Fig. 8. Experiment where forces were applied to the end-effector of IRB120. The results are shown for two of the joints; for the first joint in the top row, and for the fourth joint in the bottom row. The left subplots show the raw torque response for the nominal controller parameters and the right subplots the response for the case with high integral gain. No validation data are available as no force sensor was mounted on the robot. The manually applied forces were intended to be equally large in both the left and the right subplots.

## V. DISCUSSION

The LTP described in this paper was passive, in the sense that no force-feedback control loops were used. An alternative implementation would be to make it active, i.e., such that each joint is actively controlled. In the active approach, the problem is about estimating the external forces in the presence of the friction disturbances, rather than compensating for friction. A drawback with that method is that it is difficult to get good performance in both free-space motions and when the robot is in contact with the environment. A controller that performs well in free-space motion may become unstable in contacts with stiff environments. Forces may build up quickly in contact with a stiff environment, and the dynamics of the robot and small time delays that exist in the control system become problematic for the feedback interconnection [17]. Stable controllers for stiff environments can be designed, e.g., using the notion of passivity, and the performance can be increased by also modeling the environment [18]. On the other hand, a controller performing well during stiff contacts can be sluggish and hard to move during free-space motions. A controller that switches between two different parameter settings can solve this problem, but it would be difficult to automate the switching and making the switching manually would decrease the user-friendliness. The passive LTP contains no feedback loops and hence does not suffer from this drawback.

The presented LTP implementation contained a lot of different switches, i.e., it is based on a hybrid control approach [19]. A number of parameters has to be tuned, which may make it difficult to use the method. On the other



hand, as each joint can be tuned individually, it is quite easy to perform the tuning procedure. The implementation in this paper was based on the ABB research interface, but the implementation should be possible to do with other robots and interfaces with similar performance, such as the KUKA-FRI and Comau C5G Open.

The LTP was implemented in joint space, i.e., each joint of the robot moved independently. One benefit of doing this is that no problems with singularities of the robot will occur. A disadvantage is that in a purely passive approach it is not possible to make the robot keep the end-effector orientation and only move linearly. On the other hand, with the friction compensation, it is fairly easy to manually fix the orientation of the end-effector while moving the robot. For a redundant robot like YuMi, LTP implemented in task space would further have to control the redundant degrees of freedom. With the presented joint-space LTP, it is quite intuitive to use two hands, one to take the end-effector to the desired position, and the other to move the elbow of the robot accordingly.

Using a force sensor for implementing LTP has the advantage that problems with friction can be avoided, provided that the friction in the joints is handled by the low-level joint controllers. The sensor is usually attached to the wrist of the robot and it can be used to perform LTP of the end-effector. Any redundant degrees of freedom will have to be taken care of by the controller, and only forces applied that the force sensor can measure will give rise to motions of the robot, i.e., applied forces with contact points on the robot arm inside the mounting position of the force sensor will not lead to any motions. As the implementation of LTP with a force sensor most commonly is the active version, the problem with instability in contact with stiff environments is also present.

The proposed LTP was evaluated experimentally on two different robots. Lead-through programming with YuMi worked better, mainly due to significantly lower amount of friction present in the joints. For LTP purposes in particular, the benefit of a force sensor increases with the amount of friction in the robot, which is usually correlated with the size of the robot. Overcoming the friction with a sensorless approach for a large robot may demand too large forces, but with a force sensor, the operator only has to apply a force that is larger than the noise level of the sensor.

## VI. CONCLUSIONS

A method for performing sensorless LTP with industrial robots was presented. The method works by disabling the low-level joint controllers and only feedforward the torque due to gravity. It was reported how friction compensation could be added based on the measured velocity, which was shown to decrease the external force needed from the operator to move the robot. The sensitivity to external torques when the robot was not moving was further shown to be improved significantly by using the joint controllers with increased integral gain. The LTP was implemented on two different robots and experimentally evaluated, and it was

shown that the time for the programming phase can be substantially decreased compared to using the teach pendant. A version of this LTP for ABB YuMi is now commercially available [15].

## REFERENCES

- [1] Z. Pan, J. Polden, N. Larkin, S. Van Duin, and J. Norrish, "Recent progress on programming methods for industrial robots," *Robotics and Computer-Integrated Manufacturing*, vol. 28, no. 2, pp. 87–94, 2012.
- [2] V. Ang, L. Wei, and S. Y. Lim, "An industrial application of control of dynamic behavior of robots - a walk-through programmed welding robot," in *Proc. Int. Conf. Robotics and Automation (ICRA)*, vol. 3. San Francisco, USA, April 2000, pp. 2352–2357.
- [3] S. Wrede, C. Emmerich, R. Grünberg, A. Nordmann, A. Swadzba, and J. Steil, "A user study on kinesthetic teaching of redundant robots in task and configuration space," *Journal of Human-Robot Interaction*, vol. 2, no. 1, pp. 56–81, 2013.
- [4] S. Calinon and A. Billard, "Active teaching in robot programming by demonstration," in *Proc. Int. Symp. Robot and Human interactive Communication (RO-MAN)*. Jeju Island, Korea, Aug. 2007, pp. 702–707.
- [5] K. Eom, I. Suh, W. Chung, and S. Oh, "Disturbance observer based force control of robot manipulator without force sensor," in *Proc. Int. Conf. Robotics and Automation (ICRA)*, Leuven, Belgium, May 1998, pp. 3012–3017.
- [6] A. Alcocer, A. Robertsson, A. Valera, and R. Johansson, "Force estimation and control in robot manipulators," in *Proc. 7th IFAC Symp. Robot control (SYROCO)*. Wrocław, Poland, Sep. 2003, pp. 31–36.
- [7] M. Van Damme, B. Beyl, V. Vanderborght, V. Grosu, R. Van Ham, I. Vanderniepen, A. Matthys, and D. Lefeber, "Estimating robot end-effector force from noisy actuator torque measurements," in *Proc. Int. Conf. Robotics and Automation (ICRA)*. Shanghai, China, May 2011, pp. 1108–1113.
- [8] M. Linderöth, A. Stolt, A. Robertsson, and R. Johansson, "Robotic force estimation using motor torques and modeling of low velocity friction disturbances," in *Proc. Int. Conf. Intelligent Robots and Systems (IROS)*. Tokyo, Japan, Nov. 2013, pp. 3550–3556.
- [9] A. De Luca, A. Albu-Schäffer, S. Haddadin, and G. Hirzinger, "Collision detection and safe reaction with the DLR-III lightweight manipulator arm," in *Proc. Int. Conf. Intelligent Robots and Systems (IROS)*. Beijing, China, Oct. 2006, pp. 1623–1630.
- [10] A. Stolt, A. Robertsson, and R. Johansson, "Robotic force estimation using dithering to decrease the low velocity friction uncertainties," in *Proc. Int. Conf. Robotics and Automation (ICRA)*. Seattle, USA, May 2015, pp. 3896–3902.
- [11] A. Blomdell, G. Bolmsjö, T. Brogårdh, P. Cederberg, M. Isaksson, R. Johansson, M. Haage, K. Nilsson, M. Olsson, T. Olsson, A. Robertsson, and J. Wang, "Extending an industrial robot controller—Implementation and applications of a fast open sensor interface," *IEEE Robot. Automat. Mag.*, vol. 12, no. 3, pp. 85–94, 2005.
- [12] A. Blomdell, I. Dressler, K. Nilsson, and A. Robertsson, "Flexible application development and high-performance motion control based on external sensing and reconfiguration of ABB industrial robot controllers," in *Proc. ICRA 2010 Workshop on Innovative Robot Control Architectures for Demanding (Research) Applications*, Anchorage, Alaska, USA, May 2010, pp. 62–66.
- [13] B. Siciliano, L. Sciacivco, L. Villani, and G. Oriolo, *Robotics: modelling, planning and control*. London, U.K.: Springer-Verlag, 2009.
- [14] O. Smith, "Closer control of loops with dead time," *Chemical Engineering Progress*, vol. 53, no. 5, pp. 217–219, 1957.
- [15] ABB Robotics, "ABB YuMi," <http://new.abb.com/products/robotics/yumi>, 2015.
- [16] —, "ABB IRB120," <http://new.abb.com/products/robotics/industrial-robots/irb-120>, 2015.
- [17] C. An and J. Hollerbach, "Dynamic stability issues in force control of manipulators," in *American Control Conf. (ACC)*. Minneapolis, USA, Jan. 1987, pp. 821–827.
- [18] S. Buerger and N. Hogan, "Relaxing passivity for human-robot interaction," in *Proc. Int. Conf. Intelligent Robots and Systems (IROS)*. Beijing, China, Oct. 2006, pp. 4570–4575.
- [19] J. Lunze and F. Lamnabhi-Lagarrigue, Eds., *Handbook of hybrid systems control: theory, tools, applications*. Cambridge, U.K.: Cambridge University Press, 2009.