Observer-based ILC applied to the Gantry-Tau parallel kinematic robot — modelling, design and experiments

Johanna Wallén, Isolde Dressler, Anders Robertsson, Mikael Norrlöf, Svante Gunnarsson

Division of Automatic Control
E-mail: johanna@isy.liu.se, isolde.dressler@control.lth.se, anders.robertsson@control.lth.se, mino@isy.liu.se, svante@isy.liu.se

15th October 2010

Report no.: LiTH-ISY-R-2968

Address:
Department of Electrical Engineering
Linköpings universitet
SE-581 83 Linköping, Sweden

WWW: http://www.control.isy.liu.se
Abstract

Three different approaches of iterative learning control (ILC) applied to a parallel kinematic robot are studied. First, the ILC algorithm is based on measured motor angles only. Second, tool-position estimates are used in the ILC algorithm. For evaluation, the ILC algorithm finally is based on measured tool position. Model-based tuning of the ILC filters enables learning above the resonance frequencies of the system. The approaches are compared experimentally on a Gantry-Tau prototype, with the tool performance being evaluated by using external sensors. It is concluded that the tool performance can be improved by using tool-position estimates in the ILC algorithm, compared to when using motor-angle measurements. In the paper applying ILC algorithms to a system following trajectories with so-called lead-in/lead-out is also considered, as well as dynamic modelling of the Gantry-Tau prototype.

Keywords: Iterative methods, Learning control, Parallel, Robotic manipulators, Estimation algorithms, Performance evaluation.
Observer-based ILC applied to the Gantry-Tau parallel kinematic robot — modelling, design and experiments

Johanna Wallén* Isolde Dressler** Anders Robertsson**
Mikael Norrlöf* Svante Gunnarsson*

* Division of Automatic Control, Department of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden, e-mail: {johanna, mino, svante}@isy.liu.se
** Department of Automatic Control, Lund University, Box 118, SE-221 00 Lund, Sweden, e-mail: {isolde.dressler, anders.robertsson}@control.lth.se

Abstract: Three different approaches of iterative learning control (ILC) applied to a parallel kinematic robot are studied. First, the ILC algorithm is based on measured motor angles only. Second, tool-position estimates are used in the ILC algorithm. For evaluation, the ILC algorithm finally is based on measured tool position. Model-based tuning of the ILC filters enables learning above the resonance frequencies of the system. The approaches are compared experimentally on a Gantry-Tau prototype, with the tool performance being evaluated by using external sensors. It is concluded that the tool performance can be improved by using tool-position estimates in the ILC algorithm, compared to when using motor-angle measurements. In the paper applying ILC algorithms to a system following trajectories with so-called lead-in/lead-out is also considered, as well as dynamic modelling of the Gantry-Tau prototype.

Keywords: Iterative methods, Learning control, Parallel, Robotic manipulators, Estimation algorithms, Performance evaluation.

1. INTRODUCTION

Parallel kinematic machines (PKMs) have potential of high performance due to their stiffness compared to serial robots. Despite this fact, PKMs are still seldom used in industrial automation. Although the interest in parallel robot geometries was renewed by the Stewart-Gough platform [Merlet, 2000], it was not until the Delta structure was invented by Clavel [1991] that the PKMs were used in industrial applications. Several basic, but challenging problems still have to be solved.

The first challenge regarding parallel robots is to find a mechanical structure useful for high performance applications [Brogårdh, 2002]. The Delta structure has centralised placement of the actuators, which enables pick-and-place applications of accelerations up to 12g. However, the limitations regarding the Delta structure are its small payload (a few kilos) and limited working range. The so-called Gantry-Tau configuration [Johannesson et al., 2004] illustrated in Figs. 1 and 3 is designed to have a large workspace to machine footprint ratio compared to other parallel robots, while still being stiff compared to a serial robot. The Gantry-Tau robot can be used for example in aerospace industry, where there is a need of machining large components with very high accuracy require-
ments [Brogårdh, 2009, Crothers et al., 2009]. Another advantage of the Gantry-Tau prototype is the possibility of a modular construction, thus enabling reconfiguration for flexible manufacturing processes of small lot sizes in small-size enterprises [Dressler et al., 2007a].

The next important challenge is the price-performance ratio needed for applications where the present serial robot technology is unsatisfactory [Brogårdh, 2002]. Improved methods for measurement, modelling, identification and compensation of kinematic errors must be developed. To avoid the need of expensive high-precision components and use of high-precision assembly, calibration of the parallel

Fig. 1. Schematic picture of the Gantry-Tau parallel robot.
structure is important [Dressler et al., 2010]. Further development of the dynamic modelling of the Gantry-Tau prototype is however needed in order to achieve high accuracy at high velocities. A remedy for handling this situation is to use iterative learning control (ILC) to compensate for kinematic as well as dynamic errors. It is a control method that compensates for repetitive errors, and originates from the robotics field. Since the first publications in 1984, see Arimoto et al. [1984], Casalino and Bartolini [1984] and Craig [1984], it has developed into an intense research area as can be seen in the survey papers Bristow et al. [2006] and Ahn et al. [2007], among others.

The main ideas presented in this report are

- Applying ILC algorithms to the Gantry-Tau robot, where the ILC algorithms are based on estimates of the robot tool position.
- Enable learning up to and above the resonance frequencies of the robot system.
- Consider the case with an ILC algorithm applied to a system following a trajectory with lead-in/lead-out.

In commercial robot systems the motor angles are the measurements available, but the control objective is to follow a desired tool path. Since dynamic models suffer from model and parameter uncertainties, especially at higher frequencies, it is hard to improve the tool performance by ILC algorithms based on motor-angle measurements only. Another difficulty in industrial applications is measuring actual tool position. It is therefore preferable to use additional sensors, like accelerometers, in combination with signal processing algorithms to obtain accurate estimates of the relevant signals [Norrlöf and Karlsson, 2005]. These estimates can then be used in the ILC algorithm to improve the tool performance and hence be able to compensate for errors originating from the flexible robot structure, as well as kinematic errors. In this report experimental results with ILC algorithms using estimates of the tool position of a Gantry-Tau robot are compared to ILC algorithms using measured motor angles. The estimates are derived from motor-angle and tool acceleration measurements by using complementary filtering and Kalman filtering, respectively. In the experiments, the actual tool performance is evaluated by measurements from length gauges. However, these measurements cannot be assumed available in industrial applications, and are used only for evaluation purposes in this report. Experiments when the ILC algorithm is based on the measured tool position are therefore used as an illustration of what can be achieved. The work in this report also includes dynamic modelling of the Gantry-Tau prototype, as well as implementation issues, such as time weighting of the ILC signal when the trajectory has lead-in and lead-out parts.

To the best of our knowledge, there are only a few earlier contributions where ILC algorithms are applied to PKMs. In Abdellatif et al. [2006] and Abdellatif and Heimann [2010] linear ILC algorithms are applied to the direct-driven hexapod PaLiDa with arm slider length of 1 m. The main objectives in these works are on accuracy of high-velocity motions, as is also the focus in this report. In the papers, the ILC algorithms are based on the measured motor positions and the evaluation of the tool performance is performed by transforming the motor positions by the forward kinematics. In Cheung and Hung [2009] an ILC algorithm based on measured joint positions is applied to a planar parallel manipulator prototype with a small workspace (4×2 cm) intended for semiconductor packaging operations. The ILC algorithm in Cheung and Hung [2009] is intended to reduce the error at lower frequencies, and therefore the higher frequency and resonance frequency components are not learned in order not to excite the robot at those frequencies. A high-precision laser measurement system provides measurements of the resulting tool position. In Burdet et al. [2001] ILC algorithms based on measured motor angles are applied to a 3-DOF micro-Delta parallel robot and two sizes of 2-DOF parallel robots with arm lengths of 12 cm and 80 cm, respectively. To overcome the problem with a flexible robot structure, the learning algorithm has a low bandwidth to prevent instability. A 3-PRPS Stewart platform with arm lengths of approximately 0.5 m are used in Chuang and Chang [2001]. The performance is improved and evaluated experimentally by using an ILC algorithm based on measured motor angles. The Gantry robot in Freeman et al. [2010] can be treated as three separate SISO systems (one for each axis) and the axis position is measured by incremental encoders. First an ILC algorithm using the measured motor positions is applied to the system, followed by an ILC algorithm improving the synchronisation between the Gantry robot and the conveyor belt for a pick-and-place application. See also Ratcliffe et al. [2006], where the same Gantry robot is used, and the system states are estimated by means of an observer and used in a norm-optimal ILC algorithm. Some work related to this report has also been performed with ILC applied to a direct-driven 1D positioning table, see for example Qin and Cai [2001] and Cai and Huang [2000]. Although the much stiffer positioning table has a small working range compared to the much larger Gantry-Tau robot, which makes the questions to be studied quite different, it can be noted that the ILC algorithms are based on measurements of the position of the table.

The experiments in the publications mentioned above are performed on small parallel kinematic robots with a different mechanical design compared to the Gantry-Tau robot, which in many cases result in a stiffer robot structure than the Gantry-Tau robot. When having mechanical robot structures where problems with flexible dynamics occur, as in Burdet et al. [2001] or Cheung and Hung [2009], the problems are in those papers solved by simply using an ILC algorithm with a bandwidth below the resonance frequencies in order to prevent instability.

Estimation techniques and ILC have to our knowledge only been combined in a very few publications, as for example the work by Ratcliffe et al. [2006] mentioned above. Another example is Gunnarsson et al. [2007], where the ILC algorithm uses an arm-angle estimate calculated from measurements of motor angle and arm-angular acceleration of a flexible one-link robot arm. In Schöllig and D’Andrea [2009] the model error of a state-space model linearised along the desired trajectory is estimated by using a Kalman filter in iteration-domain. The ILC input signal at next iteration is given by minimising the deviation of the states from the desired trajectory. Another example is Tayebi and Xu [2003], where the estimated states for
a class of time-varying nonlinear systems are used in an ILC algorithm and asymptotic behaviour of the system is discussed. In Ratcliffe et al. [2006] a norm-optimal ILC algorithm is used and the system states are estimated by means of a full-state observer. The focus in these papers is on specific estimation and/or ILC algorithm techniques, while the principal experimental result of ILC based on estimates is discussed in this report, and can be seen as an experimental illustration of the results presented in Wallén et al. [2009] and Wallén et al. [2011].

The report is organised as follows: Section 2 describes the Gantry-Tau robot and the experimental setup, and the nominal properties of the system is discussed in Section 3. Robot models used for stability analysis of the ILC algorithms and for estimation of the robot tool position are given in Section 4, followed by a short introduction to ILC in Section 5 and a discussion in Section 6 about the robot tool position estimation. Section 7 presents the experimental conditions, and in Section 8 the experimental results are discussed in detail. Finally, Section 9 concludes the report.

2. GANTRY-TAU ROBOT AND EXPERIMENTAL SETUP

In this section the Gantry-Tau robot is presented in more detail, as well as the external sensors used in the experiments.

2.1 Robot

The three degrees-of-freedom (DOF) Gantry-Tau parallel robot, shown in Figs. 1 and 3, is a Gantry variant of the Tau parallel robot [Crothers et al., 2009]. It consists of three kinematic chains, where each chain is driven by a linear actuator consisting of a cart moving on a linear 5 m long guideway. The three carts are connected to the end-effector plate via three link clusters, grouped in a 3-2-1 configuration. Arm lengths of of 1.8–2 m, gives a cross-section of the workspace in the \( yz \)-plane of \( 1.1 \times 0.95 \) m, see Fig. 2. The links are connected to the carts and end-effector plate by spherical joints. The so-called Tau configuration, named after a specific case of the Tau joint placement reminding of the greek letter \( \tau \), is such that the links belonging to one cluster forms parallelograms. This results in a constant end-effector orientation, and the Gantry-Tau robot therefore has three purely translational DOF. There exist several possibilities of extending the basic Gantry-Tau configuration to obtain a robot with higher DOF. One, which is implemented on the prototype used in this work, is to add a 2-axis serial wrist on the PKM end-effector plate to give altogether a 5-DOF motion, see Fig. 3. The serial wrist is however fixed in the experiments.

The static end-effector positioning accuracy is also experimentally identified. For the prototype used here and the calibrated nominal kinematic model, the mean end-effector positioning error is 140 \( \mu m \). In Dressler et al. [2010] it is showed for a different prototype with similar static positioning accuracy that the mean error can be decreased to 90 \( \mu m \) using an error kinematic model. For this different Gantry-Tau prototype Crothers et al. [2009] states the static cartesian stiffness to vary mostly between 2–4 N/\( \mu m \) and the mean omnidirectional end-effector positioning repeatability to be 13 \( \mu m \).

2.2 Control system

The Gantry-Tau prototype is controlled by a standard industrial ABB IRC 5 system, where the trajectory generator is based on the kinematic model in Dressler et al. [2007b]. The motor-angle references are generated from the trajectory generator and then have to be followed in order to achieve the desired tool trajectory. The system is stabilised by controlling each motor independently with a standard cascade P/PI-controller consisting of an inner velocity and an outer position loop.

Fig. 2. Cross-section of the workspace of the Gantry-Tau parallel robot in the \( yz \)-plane, where the three small squares are the cross-sections of the tracks. In the yellow area the inverse kinematics has a solution, and the area that can be reached without risking broken joints is marked blue. The allowed workspace area implemented in the IRC5 system, marked by the rectangle, is \( 1.1 \times 0.95 \) m.

Fig. 3. Experimental set-up of the Gantry-Tau robot. The robot structure is shown to the left. To the right it is shown how the 3-DOF accelerometer (grey box mounted on the end-effector plate) and the length gauges (measuring the relative motion to the plate mounted on the serial wrist) are mounted.
model for feedforward control or any other compensation of the coupling effects between the motors is implemented. These effects can therefore be interpreted as disturbances.

The implementation of customised control solutions and integration of external sensors is enabled by an extension to the IRC 5 system described in Blomdell et al. [2005]. This architecture enables shared memory access via a PCI system bus and Ethernet communication. Data from the external sensors, accelerometer and length gauges, are obtained directly in the shared memory and are thus synchronised with the robot system. The sampling time is $T_s = 4$ ms. Signals that are sent from the IRC 5 main controller to the axis controllers of each motor can be read and modified by the external system and thus customised controllers modelled in Simulink and translated to C code using RTW code generation can be integrated into the system. The control signal which is acting most directly on each motor and can be changed with the external controller is the motor torque reference. For the ILC experiments however, the ILC input signal is added to the motor-angle reference. The derivative of the ILC input signal is added to the motor angular velocity reference.

### 2.3 External sensors

**3-DOF accelerometer** A Freescale MMA7361L accelerometer [Freescale, 2010] is mounted on the 3-DOF end-effector, see Fig. 3(b). The accelerometer can measure accelerations up to a bandwidth of 300Hz for all three degrees of freedom, within a range of ±1.5 g. The standard deviation of the measurement noise for the different channels is measured to be between 0.06 and 0.07 m/s². The calibrated accelerometer measures the specific force, also known as proper acceleration, that is, both the linear acceleration of the accelerometer and the acceleration due to gravity, as in

$$y_s^a(t) = a^s(t) - g^x + \delta^x_s(t) + e^x_s(t)$$

The low-frequency offset is described by $\delta^x_s$ (for example measurement errors caused by temperature changes and small changes in orientation of the end-effector due to non-ideal mechanical robot construction) and $e^x_s$ is measurement noise. The measurement is given with respect to the coordinate system of the accelerometer, denoted s (sensor). Since the acceleration due to gravity is constant in the earth coordinate system, the accelerometer measurement (1) can be used to measure the acceleration when the orientation of the accelerometer is known relative to the robot base coordinate frame fixed to the earth coordinate system.

From experiments when the accelerometer is at rest in 14 different orientations, the offset $a_{\text{offset}}$ and scaling factor $a_{\text{factor}}$ of the accelerometer measurements can be determined according to the following optimisation problem

$$\min_{a_{\text{factor}}, a_{\text{offset}}} \sum_{i=1}^{n} \left( g^2 - (a_{\text{factor}} y_a^s + a_{\text{offset}})^2 \right)^2$$

for the $n = 14$ number of measured points, where $y_a^s$ denotes the mean value of the accelerometer output. The mean error $[g - (a_{\text{factor}} y_a^s + a_{\text{offset}})]$ for the 14 calibration measurements after optimisation is 0.047 m/s².

The accelerometer is mounted on the 3-DOF plate of the robot, see Fig. 3(b), which means that it practically cannot change orientation during the experiments due to the mechanical design of the robot. Therefore the orientation of the accelerometer with respect to the robot tool frame is hard to determine with a good accuracy.

**Length gauges** Two Heidenhain length gauges ST 3078 [Heidenhain, 2010] mounted on a stand are used for measuring the tool position, see Fig. 3(b). As only two gauges are available, the motion in $xy$-direction (horizontal motion) is measured to capture the difference in dynamic properties of the $yz$-plane compared to the $x$-direction, which is the actuator direction. The gauges have a range of 30 mm and accuracy of ±1 μm. The measurement setup allows for motions in a small square in the $xy$-plane of size $30 \times 30$ mm, placed in the middle of the robot workspace, as is shown in Fig. 3.

### 3. SYSTEM PROPERTIES

In this section the nominal performance of the robot is discussed. The dynamical behaviour as well as the dynamical repeatability is investigated when following a square-like motion used in the experiments.

#### 3.1 Trajectory

A square-like path with a side of 10 mm is chosen for the experiments, seen in Fig. 4, where the motion starts at the point and is directed downwards. The square-like motion has sharp edges ($v_{\text{ref}} = 0$ in the corners) and high programmed velocity, which will excite the flexibilities of the robot and make the dynamical characteristics of the robot even more pronounced. In the experiment the motion is performed in two consecutive rounds, which enables to have a lead-in and lead-out part of the movement.

A programmed tool velocity of $v = 100$ mm/s is used in the ILC experiments presented in Section 8. The resulting motor-angle references and motor angular velocity references from the path planning algorithm in the ABB IRC5 control system are depicted in Fig. 5, when divided by the gear ratio. The corresponding tool position reference and tool velocity reference, calculated from motor-side values transformed to the tool side by the forward kinematics are shown in Fig. 6. Due to limited workspace and acceleration, the highest possible velocity of the robot tool trajectory is slightly below $v = 100$ mm/s.

#### 3.2 Nominal performance

In order to illustrate the dynamical properties of the Gantry-Tau robot, the square-like path is programmed with a low tool velocity, $v = 10$ mm/s. This result is then compared with an experiment following the same path with a high programmed tool velocity, $v = 100$ mm/s. The results are depicted in Fig. 4, where the actual tool behaviour measured by the length gauges in $xy$-direction is compared to the reference path. From Fig. 4(a), it can be concluded that although the path is to be followed at low velocity, there are still some control errors resulting in overshoot after passing each corner. Since the robot is moving at a low velocity, the influence of the dynamical
Fig. 4. Nominal performance of the robot for low velocity \((v = 10 \text{ mm/s})\) and high velocity \((v = 100 \text{ mm/s})\). The motion starts at the point and is directed downwards. The tool reference path (reference) is drawn together with the actual tool behaviour measured by length gauges (measured).

Effects in the robot structure can be considered as small. One possible explanation of the errors may therefore be friction in the motors and drivelines. The oscillatory behaviour especially seen in the lower left corner could possibly be explained by static friction when measuring the robot tool position by the length gauges onto the tool plate, seen in Fig. 3(b).

For the experiment performed at \(v = 100 \text{ mm/s}\), see Fig. 4(b), the deviation from the tool path can be explained both by larger control errors on the motor side compared to the performance for \(v = 10 \text{ mm/s}\) and dynamical effects in the robot structure. As is discussed in more detail in Section 4, the robot is stiff in the \(x\)-direction, while it suffers from a distinct resonance frequency at around 11.5 Hz in the \(y\)-direction in this operating point. One possible explanation of the overshoot after each corner could therefore be the flexible structure.
3.3 Repeatability

Repeatability of a system is a key property for success when performing ILC experiments, since only the repeatable part of the error can be corrected in the ILC algorithm. In order to investigate the dynamical repeatability of the robot, five experiments are made under as identical conditions as possible and with the same reference signal as input to the robot.

The result is evaluated by the difference

\[ e_{\text{rep}}(t) = \frac{q_i^j(t) - q_j^j(t)}{\eta}, \quad i = 1, \ldots, 3, \quad j = 2, \ldots, 5 \]  

where \( q_i^j \) is the measured motor angle for cart \( i \) and experiment \( j \). The values are divided by the gear ratio \( \eta \) to get the corresponding tool-side values. The measure (2) is illustrated in Fig. 7. First, it can be seen that the non-repeatable elements of the errors are small. It can also be noted that the first part of the movement (from approximately 0.3 s to 0.6 s) has non-repeatable elements, especially for motor 3. The non-repeatability can probably be explained by static friction, which requires a varying amount of time to integrate the torques to overcome the static friction for different experiments. The elements could however be larger due to varying experimental conditions, for instance, temperature. Experiments have shown that it is beneficial to divide the trajectory into an introductory motion, called lead-in, followed by the actual trajectory to be learned, called learning part, and a lead-out. Lead-in is useful in practical applications, one example is laser cutting where constant tool velocity along the path is important. The full potential of lead-in is however not utilised in this work, since the trajectory is programmed with zero velocity reference in the corners. It can still be motivated by the new combination of ILC and lead-in/lead-out (see Section 5.1).

It can be concluded from Fig. 7 that although a large part of the errors can be reduced when applying an ILC algorithm to the Gantry-Tau prototype, some small errors will still remain due to the non-repeatable elements.

4. ROBOT MODELS

Earlier work on dynamic modelling of the Gantry-Tau robot includes rigid body dynamics [Dressler et al., 2007b] as well as black-box identification of compliance dynamics [Cescon, 2008], [Cescon et al., 2009]. More dynamic modelling and identification of a slightly different Gantry-Tau prototype is presented in for example Tyapin et al. [2002] and Hovland et al. [2007].

While Tyapin et al. [2002] calculated the resonance frequency of the Gantry-Tau robot to be larger than 50 Hz, it was measured to lie around 14 Hz in [Cescon et al., 2009]. The experiments presented in this work show as well notable compliant behaviour, so that a rigid body model is not sufficient. Most of the flexibility is assumed to lie in the carbon fibre links and the framework, as the newly developed spherical joints are extremely stiff and a comparison of motor-side and arm-side measurements on the motors shows considerably less flexibility than the tool motion. A new modelling compared to Cescon et al. [2009] needs not only to be done to fit more specifically the needs for ILC, but as well because the serial wrist is added since then and has changed the dynamic properties.

Below, different models are presented. As the ILC trajectory is confined to a small part of the robot workspace, linear models are identified in the center of this workspace part (compare Fig. 3). In Section 4.1, SISO models for each of the controlled motors are presented, that is, models with motor-angle reference as input and motor-angle measurement as output. The models are used for tuning the ILC algorithm based on motor angles, see Section 8.1. Section 4.2 summarizes the modelling of the complete robot structure. First, the impulse response experiments from Cescon et al. [2009] are repeated to update information about the resonance frequency. Furthermore, models used for tuning the tool-side ILC algorithm and for estimating the tool motion are described.

4.1 Motor models

First, identification experiments are performed in order to derive local, linear SISO models for the three motors with motor-angle reference as input and measured motor angle as output. As the tuning of the very simple ILC algorithm used for motor-side ILC only requires the system’s stationary gain and delay, and experiment results prove that first order systems model the controlled motors adequately, three step response experiments are performed. In order to disregard the non-linear coupling effects between the three motors, all step response experiments were performed with all three motors moving simultaneously and the tool moving in the x-direction seen in Fig. 1. This procedure facilitates obtaining linear SISO models, but it limits as well the identified models’ validity for other movements involving coupling effects. The amplitude of the steps are
chosen as 1 mm, 1.5 mm and 2 mm, respectively, and are motivated by the ILC experiments described in Section 7, where the robot makes small movements within a few millimeters from the working point.

A model type with few parameters is sufficiently good for the purpose with an ILC algorithm applied to the robot using motor-angle measurements, therefore an ARX model structure (AutoRegressive with eXternal input) with \( na = 1, nb = 1 \) and \( nk = 5 \) is used in the identification, see Ljung [1999] and Ljung [2010]. The three motor models are derived using the System Identification Toolbox in MATLAB [Ljung, 2010] using data from all step response experiments with the different amplitudes, resulting in

\[
T_{r,1}(q) = q^{-5} \frac{0.03527}{1 - 0.9647q^{-1}}
\]

\[
T_{r,2}(q) = q^{-5} \frac{0.03425}{1 - 0.9656q^{-1}}
\]

\[
T_{r,3}(q) = q^{-5} \frac{0.02748}{1 - 0.9728q^{-1}}
\]

(3)

From the relations above, it can be seen that models for motor 1 and motor 2 are close to each other, while the model for motor 3 is somewhat different. As the above SISO models are a simplification of a complex, non-linear dynamic system, no good explanation can be given for that. The control loop of the motor-angle is fairly slow and can be approximated by a low-pass filter. The 5 samples delay are presumably caused by internal data communication in the IRC 5 system.

The models are validated using the part of the data from the 1.5 mm step response experiment (Fig. 8) which has not been used for identification. The step response experiments (Fig. 8) show a good approximation of the controlled motors with a delayed first-order system. However, even if different data is used for validation, still the same type of movement is performed. A validation with the rectangular ILC trajectory (Section 3.1) shows that the identified models are too slow for modeling the robot performing this different type of movement, where stronger coupling effects between the actuators appear. Linear models that perform well for all kinds of movements are extremely difficult to find, and the identified models still fulfill their purpose for tuning the motor-side ILC.

4.2 Models of the complete robot structure

Resonance frequency To determine the new resonance frequencies (of the robot including the serial wrist), impulse response experiments are performed. A short force impulse is applied on the 3-DOF end-effector plate and the position is measured with the linear gauges in \( x \)- and \( y \)-direction. Four different experiments are carried out, with the brakes applied and released, and with a force in \( x \)- and \( y \)-direction, respectively.

The behaviour observed does not differ for applied or released brakes. When the force is applied in \( x \)-direction, a resonance of 7.4 Hz is observed in \( x \)-direction, while the end-effector starts a vibration with a frequency between 11.4 and 11.9 Hz in \( y \)-direction, which after a few oscillations turns into a 7.3 – 7.5 Hz vibration. When the force is applied in \( y \)-direction, the resonance frequency in \( y \)-direction is between 11.4 and 11.5 Hz, and the resonance frequency in \( x \)-direction is between 10.4 and 10.7 Hz.

Linear black-box models Different linear black-box models are identified. They can be distinguished by

- the data used for identification: Two kinds of experiment data are used for identification. A different pseudo random binary sequence (PRBS) signal with an amplitude of 1 mm and adjustable frequency content is added to each of the motor references. As it is difficult to obtain a model performing well enough for estimation, identification is as well attempted on the square-like reference trajectory (with \( v = 100 \) mm/s).
- the input and output signals chosen: The essential output signal is certainly the tool position, which is measured by the linear gauges. For a Kalman filter estimating the tool position based on accelerometer data, the tool acceleration has to be an output signal as well. However, better results are obtained when the model is manually extended with states corresponding to the acceleration compared to fitting a black-box model to measured accelerometer data. A reason for this might be the rather noisy accelerometer signal. As input signal, the reference for the tool position is used. Another possibility is to use the motor-angle measurements transformed to the tool side by the forward kinematics. This signal has no direct physical correspondence, but can be considered closer connected to the output signal and the resulting model performs well when used for estimation.
• the intended usage: Besides knowledge about the robot properties in general, the robot models are needed for tuning the ILC filters and designing a Kalman filter for estimating of the tool position.

The diagonal elements of the model (5) from the tool position reference to the tool position measurement are used for tuning the ILC filters. The model is identified from PRBS data.

For the Kalman estimator, the model (6) from the tool position calculated by the kinematics to the measured tool position is used. The model is identified form the square-like reference trajectory \(v = 100\text{mm/s}\).

Figure 9 shows the Bode plots of the two models which are finally used for the ILC tuning and the Kalman estimator. The resonances around 10 Hz can clearly be seen.

5. ILC ALGORITHMS

For analysis purposes, the following discrete-time LTI system is studied with system description at iteration \(k\) given by

\[
y_k(t) = T_{ry}(q)r(t) + T_{uy}(q)u_k(t)
\]

\[
z_k(t) = T_{rz}(q)r(t) + T_{uz}(q)u_k(t)
\]

with measured variable \(y_k\) and controlled variable \(z_k\). Throughout the report it is assumed that the system is stable, which is guaranteed since the description (7) contains the closed-loop dynamics of the system.

A linear discrete-time first-order ILC algorithm of the following structure is used,

\[
u_{k+1}(t) = Q(q)\{u_k(t) + L\epsilon_k(t)\}
\]

with possibly noncausal ILC filters \(Q\) and \(L\) and the error \(\epsilon_k\) in the ILC update equation is given by

\[
\epsilon_k(t) = r(t) - \hat{z}_k(t)
\]

using the estimate \(\hat{z}_k\) given by

\[
\hat{z}_k(t) = F_r(q)r(t) + F_u(q)u_k(t) + F_y(q)y_k(t)
\]

with the stable filters \(F_r\), \(F_u\) and \(F_y\).

The system (7) and ILC algorithm (8) can be expressed in matrix form, which is closely related to the descriptions in for example Phan and Longman [1988] and Moore [1998]. First, define the vector \(r\) of the \(N\)-sample sequence of the reference \(r\) as

\[
r = (r(0) \ldots r((N-1)T_s))^T
\]

Next, define the vectors \(u_k, y_k, z_k\) and \(\hat{z}_k\) similarly. The matrix \(T_{ry}\) is formed by the pulse response coefficients \(g_{T_{ry}}, t \in \{0, \ldots, (N-1)T_s\}\) of the transfer operator \(T_{ry}\) in (7). This results in the Toeplitz matrix

\[
T_{ry} = \begin{pmatrix}
    g_{T_{ry}}(0) & 0 & \cdots & 0 \\
    g_{T_{ry}}(1) & g_{T_{ry}}(0) & \cdots & 0 \\
    \vdots & \ddots & \ddots & \ddots \\
    g_{T_{ry}}(N-1) & g_{T_{ry}}(N-2) & \cdots & g_{T_{ry}}(0)
\end{pmatrix}
\]

The system matrices \(T_{uy}, T_{rz}, T_{uz}, F_r, F_u\) and \(F_y\) are defined similarly. Finally, the system description (7) is rewritten in matrix form as

\[
y_k = T_{ry}r + T_{uy}u_k
\]

\[
z_k = T_{rz}r + T_{uz}u_k
\]

The ILC algorithm (8) is in matrix form given by

\[
u_{k+1} = Q(\epsilon_k + L\epsilon_k)
\]

\[
\epsilon_k = r - \hat{z}_k
\]

using the estimate

\[
\hat{z}_k = F_r r + F_u u_k + F_y y_k
\]

How to derive the matrices \(Q\) and \(L\) for the analysis of the ILC system is discussed in next section. Stability and convergence can be analysed using the results presented in Wallén et al. [2011]. First, the system (13) controlled by the ILC algorithm (14) using the estimate (15) is stable if and only if

\[
\rho\left(\frac{Q(I - L(F_u + F_y T_{uy})})\right) < 1
\]

where \(\rho(\cdot)\) denotes the spectral radius of the matrix. Second, if the system (13) is controlled by the ILC algorithm (14) and

\[
\sigma\left(\frac{Q(I - L(F_u + F_y T_{uy})})\right) < 1
\]

that is, the largest singular value is smaller than one, then the system is stable and \(u_k\) converges to the limit value \(u_\infty\) with monotone exponential convergence.

5.1 Implementation

The filter \(L\) is chosen as \(L = \gamma q^t\), which means a learning gain \(\gamma\) and a time shift of \(\delta\) samples. This noncausal filter is implemented by letting

\[
\epsilon_k(t) = \epsilon_k(T_s(N-1)), \quad t > T_s(N-1)
\]

which corresponds to

\[
L = \begin{pmatrix}
    0 & \cdots & 0 & \gamma & 0 & \cdots & 0 \\
    0 & \cdots & 0 & \gamma & 0 & \cdots & 0 \\
    \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\
    0 & \cdots & 0 & \gamma & 0 & \cdots & 0 \\
\end{pmatrix}
\]

in the matrix form of the implementation of the \(L\) filter, as is discussed in more detail in Wallén et al. [2010].

The chosen structure of the filter \(Q\) is a non-causal filter with zero-phase characteristics. A standard way to carry out such filtering is to use a causal filter \(Q\), and perform forward-backward filtering in order to obtain a zero-phase filter. The lower-triangular Toeplitz matrix \(Q\) is created similarly as in (12) from the pulse response coefficients of the filter \(Q\). The forward-backward filtering is in the matrix description interpreted as

\[
\hat{Q} = Q^T Q
\]

See Wallén et al. [2010] for the details.

The ILC algorithm is applied to a system where the trajectory to be followed has so-called lead-in and lead-out with the actual trajectory of interest in between. The part of the trajectory to be learned by the ILC algorithm is hereafter called the learning part of the trajectory, while the remaining parts of the trajectory are called lead parts. Combining ILC and trajectories with lead parts implies that the ILC input signal needs to be weighted in time-domain when applied to the system. This is performed by
\[
\begin{bmatrix}
X_{\text{kin}} \\
Y_{\text{kin}}
\end{bmatrix} = \\
\begin{bmatrix}
-0.01063 q^5 + 0.04867 q^4 - 0.08307 q^3 + 0.06915 q^2 - 0.02756 q + 0.003539 \\
-0.008403 q^3 + 0.04354 q^2 - 0.0971 q^1 + 0.1089 q^0 - 0.06019 q - 0.01133 \\
q^6 - 4.672 q^5 + 9.591 q^4 - 11.12 q^3 + 7.702 q^2 - 3.035 q + 0.536 \\
0.009528 q^5 - 0.04158 q^4 + 0.07683 q^3 - 0.07497 q^2 + 0.03834 q - 0.008148 \\
q^6 - 4.672 q^5 + 9.591 q^4 - 11.12 q^3 + 7.702 q^2 - 3.035 q + 0.536 \\
-0.01087 q^3 + 0.04841 q^2 - 0.07341 q^1 + 0.04793 q^0 - 0.01027 q - 0.00177 \\
q^6 - 4.672 q^5 + 9.591 q^4 - 11.12 q^3 + 7.702 q^2 - 3.035 q + 0.536
\end{bmatrix}
\begin{bmatrix}
X_{\text{ref}} \\
Y_{\text{ref}}
\end{bmatrix}
\tag{4}
\]

\[
\begin{bmatrix}
X_m \\
Y_m
\end{bmatrix} = \\
\begin{bmatrix}
-0.001999 q^6 + 0.01819 q^5 - 0.06378 q^4 + 0.1244 q^3 - 0.1415 q^2 + 0.08666 q - 0.02193 \\
0.01666 q^6 - 0.06287 q^5 + 0.07712 q^4 - 0.006658 q^3 - 0.06611 q^2 + 0.05771 q - 0.01586 \\
q^7 - 5.072 q^6 + 11.23 q^5 - 14.13 q^4 + 10.83 q^3 - 4.925 q^2 + 1.137 q - 0.07708 \\
0.004358 q^6 - 0.02509 q^5 + 0.06064 q^4 - 0.08211 q^3 + 0.063 q^2 - 0.03346 q + 0.007357 \\
q^7 - 5.072 q^6 + 11.23 q^5 - 14.13 q^4 + 10.83 q^3 - 4.925 q^2 + 1.137 q - 0.07708 \\
-0.009283 q^6 + 0.03787 q^5 - 0.05919 q^4 + 0.04155 q^3 - 0.005226 q^2 - 0.01013 q + 0.00441 \\
q^7 - 5.072 q^6 + 11.23 q^5 - 14.13 q^4 + 10.83 q^3 - 4.925 q^2 + 1.137 q - 0.07708
\end{bmatrix}
\begin{bmatrix}
X_{\text{ref}} \\
Y_{\text{ref}}
\end{bmatrix}
\tag{5}
\]

\[
\begin{bmatrix}
X_m \\
Y_m
\end{bmatrix} = \\
\begin{bmatrix}
q^{10} - 8.491 q^9 + 33.26 q^8 - 79.34 q^7 + 127.9 q^6 - 145.9 q^5 + 119.1 q^4 - 68.66 q^3 - 26.72 q^2 - 6.327 q + 0.6903 \\
0.7653 q^9 - 6.759 q^8 + 26.76 q^7 - 62.45 q^6 + 94.78 q^5 - 97.11 q^4 + 67.3 q^3 - 30.28 q^2 + 8.055 q - 0.9631 \\
q^{10} - 8.491 q^9 + 33.26 q^8 - 79.34 q^7 + 127.9 q^6 - 145.9 q^5 + 119.1 q^4 - 68.66 q^3 - 26.72 q^2 - 6.327 q + 0.6903 \\
-0.3056 q^9 + 2.45 q^8 - 8.854 q^7 + 18.97 q^6 - 26.58 q^5 + 25.29 q^4 - 16.33 q^3 + 6.897 q^2 - 1.726 q + 0.1946 \\
q^{10} - 8.491 q^9 + 33.26 q^8 - 79.34 q^7 + 127.9 q^6 - 145.9 q^5 + 119.1 q^4 - 68.66 q^3 + 26.72 q^2 - 6.327 q + 0.6903 \\
-0.135 q^9 + 1.46 q^8 - 5.689 q^7 + 16.84 q^6 - 27.31 q^5 + 29.42 q^4 - 21.18 q^3 + 9.858 q^2 + 2.697 q + 0.3309 \\
q^{10} - 8.491 q^9 + 33.26 q^8 - 79.34 q^7 + 127.9 q^6 - 145.9 q^5 + 119.1 q^4 - 68.66 q^3 + 26.72 q^2 - 6.327 q + 0.6903
\end{bmatrix}
\begin{bmatrix}
X_{\text{kin}} \\
Y_{\text{kin}}
\end{bmatrix}
\]

Fig. 9. Bode diagrams: model from tool position reference to measured tool position (solid, model (5)), model from tool position calculated by kinematics to measured tool position (dashed, model (6)).
multiplying the corresponding ILC input signal vector by the weighting vector
\[ \mathbf{w} = (w(1) \ldots w(r))^T \] (20)
with \( r \) number of samples. The total length of the ILC input signal \( \mathbf{u}_k \) is \( N \) samples, where the learning starts after \( s \) samples and ends \( e \) samples before the end of the signal. This is interpreted in matrix form by a total weighting matrix
\[ \mathbf{Q}_w = \text{diag}(0, \ldots, 0, w_1, \ldots, w_s, 1, \ldots, 1, w_r, \ldots, w_1, 0, \ldots, 0) \] (21)
for the whole trajectory. Finally, the matrix \( \mathbf{Q} \) in (14) is given by
\[ \mathbf{Q} = \mathbf{Q}_w \tilde{\mathbf{Q}} \] (22)
with \( \tilde{\mathbf{Q}} \) from (19).

6. ESTIMATION OF ROBOT TOOL POSITION

The first estimate of the robot tool position in \( xy \)-direction is derived by using complementary filtering, which is a simple estimation technique easy to implement. The second estimation approach is to use a stationary Kalman filter based on the model (6). The estimate is only derived in \( xy \)-direction, because of the tool position measurements available for evaluation, see Section 2.3.

6.1 Complementary filtering

A pair of filters is called a complementary filter if the sum of their transfer functions is one over all frequencies. It is an estimation technique often used in flight control industry [Higgins, 1975], and is popular because of its simplicity. In Higgins [1975] its relation to the steady-state Kalman filter is shown for a certain class of filtering problems. No details about the noise processes are considered in complementary filtering, and the filters are derived from a simple analysis in the frequency domain. A similar kind of filter pairs is widely used in communication systems, see for example Vaidyanathan [1993], however there the sum of the transfer functions does not have zero phase since many communication systems can handle time delays.

Complementary filtering is used to fuse noisy measurements of the same physical variable from two sensors having different frequency characteristics. One example is when measurements \( s_1 \) from the first sensor have mostly high-frequency noise and measurements \( s_2 \) from the second sensor suffers from noise mostly at low frequencies. First, the measurements \( s_1 \) are filtered by using a low-pass filter \( G \). The high-pass filter \( 1 - G \), the complement to \( G \), is then used to filter the measurements \( s_2 \), before the filtered signals are summed together to form an estimate of the physical variable. [Higgins, 1975]

In the robotic case in this report, the estimate \( \hat{z}_{\text{kin}} \) derived from motor-angle measurements transformed by the forward kinematics is a good tool-position estimate for low frequencies. The estimate \( \hat{z}_{\text{acc}} \) obtained by double integration of the accelerometer output is on the other hand a good tool-position estimate for higher frequencies. These two estimates are then fused together by a complementary low-pass filter \( G \) to form the estimate \( \hat{z} \) of the robot tool position as in
\[ \hat{z}(t) = G(q)\hat{z}_{\text{kin}}(t) + (1 - G(q))\hat{z}_{\text{acc}}(t) \] (23)
The filter \( G \) is chosen as a Butterworth filter of second order, where the cutoff frequency \( f_n \) is tuned experimentally in each \( xy \)-direction for the specific square-like motion to give as good estimate as possible, resulting in
\[ f_n = (4.500 \ 2.625)^T \text{ Hz} \] (24)
The filter \( G \) is applied to give zero-phase characteristics by using the MATLAB function \texttt{filtfilt}. It can be noted that a similar approach with double integration and high-pass filtering of the accelerometer signal is used in Nordström [2006]. Simulation examples and experimental evaluation concludes that the industrial robot control is improved by using the accelerometer as an additional sensor.

In Fig. 10 the resulting estimation error \( e = z - \hat{z} \) of the robot tool position is shown for the square-like motion, where the actual tool position \( z \) is measured by the length gauges. The estimation error when using the estimate (23) is compared to the corresponding estimation error when using the estimate \( \hat{z}_{\text{kin}} \). It can be seen that when using (23), the estimation error is decreased, especially in the \( y \)-direction. This can be explained by the fact that the robot is more flexible in this direction and thereby the accelerometer measurements provide more information than is the case for the stiffer \( x \)-direction.

6.2 Kalman filtering

The robot tool position is also estimated by using a stationary Kalman filter, based on the model (6), described in state-space form as
\[ x(t + T_s) = Ax(t) + Bu(t) \]
\[ z(t) = Cx(t) \] (25)
Input to the model is measured motor angles transformed by the forward kinematics to the tool side, and model output is the robot tool position measured by the length gauges in \( xy \)-direction. The state vector \( x \) in the model (25) is thereafter extended to also incorporate states for the tool velocity, \( x_v \), and acceleration, \( x_a \), by differentiation of the tool-position states according to
\[ x_v(t + T_s) \approx \frac{z(t + T_s) - z(t)}{T_s} = \frac{C(Ax(t) + Bu(t)) - Cx(t)}{T_s} \]
and similarly for the tool acceleration. This results in the extended state-space model
\[
\begin{pmatrix}
  x_v(t + T_s) \\
  x_v(t + T_s) \\
  x_a(t + T_s)
\end{pmatrix} =
\begin{pmatrix}
  1 & 0 & 0 \\
  \frac{1}{T_s}A & 1 & 0 \\
  \frac{1}{T_s}C(A - I) & 1 & 0
\end{pmatrix} \begin{pmatrix}
  x_v(t) \\
  x_a(t)
\end{pmatrix}
\] + \begin{pmatrix}
  B \\
  \frac{1}{T_s}C \\
  \frac{1}{T_s}CB
\end{pmatrix} u(t)
\]
\[ y(t) = (0 \ 0 \ I) \begin{pmatrix}
  x_v(t) \\
  x_a(t)
\end{pmatrix} \]
(27)
where the tool acceleration is measured. Denoting
\[ \bar{x}(t) = (x(t)\ x_v(t)\ x_a(t))^T \]
the model (27) is written more compactly on the form (25) with the extended state-space matrices \( A, B, C, D \). Using measured motor angles transformed to the tool side by the forward kinematics as input \( u \) and tool acceleration \( y_a \), as measurement, the tool position \( z \) can be estimated by a linear observer of the form
\[
\ddot{x}(t + T_s) = A\dot{x}(t) + Bu(t) + K(y_a(t) - \dot{C}\dot{x}(t)) \\
\dot{z}(t) = M\dot{x}(t)
\]
(28)
The observer gain \( K \) is determined by using the procedure for the stationary Kalman filter, with tuning parameters \( R_1 \) and \( R_2 \). The covariance matrix \( R_2 \) for the measurement noise is determined from the covariance of the error when comparing the simulated accelerometer output from the model (27) to the measured acceleration in \( xy \)-direction for the square-like motion used in the experiments. The covariance matrix for the process noise is given as \( R_1 = r I \) with the factor \( r \) determined by minimising the estimation error of the square-like motion by inspection. The value \( r \) has to be rather small in order to avoid problems with drift in the estimate. The resulting estimation error for the square-like motion in Fig. 10 when using the estimate (28) is compared to the estimation errors when using the estimates \( \hat{z}_{kin} \) and \( \hat{z} \) in (23). It can be seen that the Kalman filter estimate (28) performs slightly better than the complementary filter estimate (23). It is likely that the quality of the estimate can be further increased by using a better (nonlinear) model for the specific working point in the estimation procedure.

In the ILC experiments in Section 8 the resulting tool performance is compared for the cases when all three estimates are used in different ILC algorithms.

7. EXPERIMENTAL CONDITIONS

Some notes are given in this section on how the ILC experiments are performed and evaluated.

In the ILC experiments, the square-like trajectory is used with a programmed robot tool velocity of \( v = 100 \text{ mm/s} \). The experimental results in Section 8 should be compared to the nominal performance shown in Fig. 4(b). The experiments are categorised into the following cases with the ILC algorithm applied to the robot based on:

1) Motor-angle measurements, \( \epsilon_k = r_y - y_k \)
2) Estimates of tool position, \( \epsilon_k = r - \hat{z}_k \)
3) Measurements of tool position, \( \epsilon_k = r - \hat{z}_k \)

where \( \epsilon_k \) denotes the error used in the ILC update equation (8). The measured motor angles at iteration \( k \) are denoted \( y_k \), while the robot tool estimate and measured robot tool are given by \( \hat{z}_k \) and \( z_k \), respectively. The corresponding references are denoted \( r_y \) and \( r \). Case 2 is thereafter divided into:

A) The estimate \( \hat{z}_k \) is derived by using complementary filter, see (23).
B) The estimate \( \hat{z}_k \) is given from the Kalman filter, see (28).

Due to the experimental setup with the two length gauges, only information in the \( xy \)-direction is available. This means that ILC experiments based on measurements or estimates of tool position only corrects for errors in \( xy \)-direction, and it is assumed that the error in \( z \)-direction is zero. For the same reason, the actual tool performance is only evaluated in the \( xy \)-plane.

7.1 Experimental considerations

When performing the experiments, there is a delay between each iteration when the current error is analysed and the next ILC input signal to the system is calculated. An effect of the delay is that vibrations induced in the structure die out between the iterations, thus preventing vibrations from being propagated between the iterations. Before each iteration the linear motors are also reset to their initial positions to minimise the effect of initial state errors.

The ILC algorithm is applied to a system where the trajectory to be followed has lead-in and lead-out parts, as is discussed in Section 5.1, and only the learning part of the trajectory is fully learned. The time-domain weighting of the ILC input signal is performed with a weighting vector \( (20) \) consisting of \( r = 100 \) elements, generated by \texttt{tukeywin}(2*\texttt{r}, 0.95) in MATLAB. The weighting coefficients are given in Fig. 11.

7.2 Evaluation measures

The evaluation of the experimental results is based on the error when no ILC algorithm is applied (0th iteration), called nominal error. The nominal error for each of the three motors is given by
\[
\epsilon_{y,i,0} = r_y - y_{i,0}, \quad i = 1, 2, 3
\] (29)
with the vector of measured motor angles \( \mathbf{y} \) and the corresponding motor-angle reference vector \( \mathbf{r} \). Similarly, the nominal tool position error in \( x \) and \( y \)-direction is

\[
e_{z,i,0} = \mathbf{r} - \mathbf{z}, \quad i = x, y
\]

with the tool position reference vector \( \mathbf{r} \) and the vector of measured tool position \( \mathbf{z} \) from the length gauges.

The reduction of the 2-norm of the error at iteration \( k \) is given in percentage of the nominal error, as in

\[
\bar{e}_{n,i,k} = 100 \cdot \frac{\|e_{n,i,k}\|}{\|e_{n,i,0}\|} \% \tag{31}
\]

where \( n \) symbolises motor side (\( y \)) and tool side (\( z \)), respectively. The error measure \( \bar{e}_{y,i,k} \) on the motor side is given for each motor \( i = 1, 2, 3 \) separately. The error measure of the tool, \( \bar{e}_{z,i,k} \), is given for the two directions \( i = x, y \). The quantity (31) is based on the part of the trajectory which is fully learned by the ILC algorithm, that is, the lead parts and the part where the ILC input signal is weighted are not included.

8. EXPERIMENTAL RESULTS

The results for the ILC experiments for the cases 1, 2A-B, and 3, categorised in Section 7, are discussed and compared qualitatively.

First it is investigated how far one can reach in case 1, that is, the resulting tool performance when the motor angles are close to the references. This result is thereafter compared to the experiment when the ILC algorithm is based on an estimate of the tool position, case 2, with estimation alternative A or B. Finally, the result when measurements of tool position is used in the ILC algorithm in case 3 is discussed. This last approach should be seen as an illustration of what can be achieved for a specific tuning in the case with measurements for tool position available, since this case is hard to choose in practice because of difficulties in measuring tool position in industrial applications.

In all experiments the filter \( \mathbf{L} \) in (8) is given as \( \mathbf{L} = \gamma \mathbf{I} \), with a time shift of \( \delta = 5 \) and a learning gain of \( \gamma = 0.9 \), with the parameters chosen based on knowledge of time delay and static gain of the system. The choice of filter \( \mathbf{Q} \) is also model-based and discussed in more detail for the different cases.

8.1 Case 1 — ILC based on motor-angle measurements

For low velocities, most of the error components of the robot tool position can be corrected by using an ILC algorithm based on the motor-angle errors, as can be understood from Fig. 4(a), since the influence of the dynamical effects in the robot structure is small. This can be regarded as if using the estimate \( \hat{\mathbf{z}}_{\text{kin}} \), the motor angles transformed by the forward kinematics to the tool side as described in Section 6.1, which is a good estimate at low velocities. When performing ILC experiments at high velocity, a smaller error reduction of the robot tool is expected than for the low-velocity experiments. This is because of the weak structure and dynamical effects thereby introduced in the system, as can be seen in Fig. 4(b), which are not seen from the motor-angle measurements. Neither can kinematic errors be compensated by this approach.

For case 1, an ILC algorithm is implemented on each of the three motors independently, based on the individual motor-angle errors. The same ILC design variables are applied to all motors, since the models (3) for the three motors are close to each other. The filter \( \mathbf{Q} \) is chosen as a second-order Butterworth filter with cutoff frequency \( f_n = 10 \text{Hz} \) for all motors, and is applied to give zero-phase characteristics, see Section 5.1. From the implementation of the filters \( \mathbf{Q} \) and \( \mathbf{L} \), as described in Section 5.1, it results in the corresponding matrices \( \mathbf{Q} \) and \( \mathbf{L} \). Since the ILC input signal is added to the motor-angle reference, it means \( T_{uy,i} = T_{ry,i} \) for the three motors \( i = 1, 2, 3 \), and \( F_r = F_u = 0, F_y = 1 \). The criterion (17) for stability and monotone convergence for motor \( i \), given by

\[
\bar{\sigma}_i = \bar{\sigma}(\mathbf{Q}(\mathbf{L} - \mathbf{L}T_{uy,i})) < 1 \quad \tag{32}
\]

results in the following maximum singular values

\[
\bar{\sigma}_1 \approx 0.90, \quad \bar{\sigma}_2 \approx 0.91, \quad \bar{\sigma}_3 \approx 0.93 \quad \tag{33}
\]

when calculated using the number of samples of the trajectory to be fully learned plus twice the length of the weighting vector. The stability analysis results in stability and monotone convergence of the three motor models (3) when controlled independently by the ILC algorithm.

The result for an experimental investigation of the properties of the algorithm is presented in Fig. 12. The reduction (31) is shown for the three motors, and the error measure (31) is reduced to 2% and the resulting level of error reduction is practically reached after five iterations. This is important from industrial applications point of view, where there is little time for algorithm tuning and a small effort giving a substantial error reduction after only a few iterations often is sufficient. The resulting motor-side error after 10 iterations is compared to the nominal error (29) in Fig. 13. It is illustrated how the lead part of the trajectory is not learned, and that the learning is smoothly increased until the time where the actual motion to be fully learned starts (\( t = 1.6 \text{s} \)), and then is smoothly decreased at the start of the lead-out part of the motion (\( t = 4.1 \text{s} \)).

The corresponding tool performance is shown in Fig. 14. From this figure it can be seen that the error reduction relative to the nominal error is smaller on the tool side than on the motor side seen in Fig. 13. This can also be concluded from Fig. 15, showing the tool performance for the
learning part of the trajectory. The measured tool position in \(xy\)-direction is compared to the tool reference and to the motor angles transformed by the forward kinematics to the tool side. Since the ILC algorithm is based on the motor-angle errors and not tool-side errors, the kinematics curve compared to the tool reference represents the remaining control error, which is this case is very small as seen in Figs. 12 – 13. The measured tool position shows larger deviation from the reference trajectory compared to the kinematics curve, due to dynamical effects in the robot structure that cannot be compensated for from the motor side. However, the tool performance is improved compared to the nominal tool performance shown in Fig. 4(b). In order to be able to further reduce the actual tool-
side errors, either measurements or estimates of the tool position could be included in the ILC algorithm, which is also discussed in Wallén et al. [2011].

**Repeatability** Now, this ILC experiment is repeated five times by applying the same updated reference values as inputs to the first iteration under as identical conditions as possible. The spread of the error measure (31) on motor side for the first five iterations can be seen in Fig. 17. In Fig. 18 the corresponding spread of the error measure (31) on tool side is shown. Comparing the spread shown in Fig. 17 with the result illustrated in Fig. 12, it can be concluded that the behaviour with slightly non-monotone convergence may be explained by this spread. It should also be pointed out that the spread shown in Figs. 17 – 18 can be even larger due to varying experimental conditions, for example, temperature.

**8.2 Case 2 — ILC based on estimates of tool position**

In this section the ILC algorithm is based on an estimate of the tool position. The result for case 2A with an estimate based on the complementary filter technique and case 2B when using an estimate from a Kalman filter is compared to the result for case 1 in previous section.

From the identification experiments described in Section 4.2 it is known that the robot has a distinct resonance frequency at around 11.5 Hz in the y-direction and a not so pronounced resonance frequency at 7.4 Hz in the x-direction. When choosing the filter Q according to the approach used in Section 8.1, the cutoff frequency of the low-pass filter Q has to be chosen below the resonance frequencies of the closed-loop system to give a stable algorithm, which can be realised from analysis of the convergence in frequency domain, see for example Norrlöf and Gunnarsson [2002]. To be able to learn also the resonance frequencies and higher frequencies, another design approach is therefore desirable. The filter Q is tuned to be robust with respect to large model errors especially around the resonance frequencies. Due to high-frequency measurement noise, learning up to 30 Hz is chosen, which is above the resonance frequencies of the system. The complementary filter (23) in case 2A has input signals \( \dot{z}_{\text{kin}} \) and \( \dot{z}_{\text{acc}} \). For tuning of the filter Q and stability analysis, the identified model (5) extended with acceleration states and the model (4) are used to obtain a model from the tool-position reference \( r \) to the filter output in (10). Case 2B including the Kalman filter is similarly treated. The choice of filter Q is experimentally evaluated for cases 2A and 2B to give a good error reduction with the filter Q implemented as described in Section 5.1. In Fig. 19 the magnitude \( |Q^{-1}| \) is shown together with the magnitude \( |1 - L(e_{\omega}) (F_r(e_{\omega}) + F_y(e_{\omega}) T_{uy}(e_{\omega}))| \) for the x- and y-direction, respectively. The robustness around the resonance frequencies of the system can be seen, together with the low-pass characteristics of the filter Q for higher frequencies. The criterion (17) results in

\[
\sigma_{\text{case 2A}} \approx 0.86 \\
\sigma_{\text{case 2B}} \approx 0.93
\]

In Figs. 20 – 21 the tool performance can be seen for case 2A. Similar results are achieved for case 2B, and therefore these figures are omitted. The results for case 2A and 2B are compared to the result for case 1 in Fig. (22).
The mean of the error measure (31) of the tool for iterations 5 to 10 is also given in Table 1. It can be seen that the tool-side error measure (31) is now smaller than for case 1, seen in Figs. 14 – 15. Since different ILC algorithms are tuned and applied to different systems, it is difficult to compare the results for the cases quantitatively. Another tuning could for example give a slightly smaller error reduction in x-direction, maybe resulting in slightly larger error reduction in the weaker y-direction. However, the tool performance is improved for cases 2A and 2B compared to case 1, especially in the y-direction. Since the x-direction is stiff compared to the y-direction, most of the errors in the x-direction can be compensated for already by the motor-angle errors, thereby it is hard to further improve the performance in x-direction. On the other hand, the accelerometer signal can give more information about the tool position in the y-direction than from the motor-angle measurements, which makes it possible to be able to improve the performance in that direction by using estimates of the tool position in the ILC algorithm.

Fig. 20. Case 2A: Tool performance after 10 iterations, with reference path (reference) compared to motor angles transformed by forward kinematics (kinematics) and tool behaviour measured by length gauges (measured).

Table 1. Comparison of mean value for iteration 5 to 10 of error measure (31) on tool side for the cases 1, 2A, 2B, 3.

<table>
<thead>
<tr>
<th></th>
<th>Case 1</th>
<th>Case 2A</th>
<th>Case 2B</th>
<th>Case 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>13.3 %</td>
<td>12.6 %</td>
<td>13.3 %</td>
<td>10.3 %</td>
</tr>
<tr>
<td>y</td>
<td>36.1 %</td>
<td>23.9 %</td>
<td>25.2 %</td>
<td>24.9 %</td>
</tr>
</tbody>
</table>

8.3 Case 3 — ILC based on measurements of tool position

Finally, in case 3 the ILC algorithm is based on measurements of the robot tool position in x/z-direction, and this case is an illustration of what can be achieved for a specific tuning.

The mean of the error measure (31) of the tool for iterations 5 to 10 is also given in Table 1. It can be seen that the tool-side error measure (31) is now smaller than for case 1, seen in Figs. 14 – 15. Since different ILC algorithms are tuned and applied to different systems, it is difficult to compare the results for the cases quantitatively. Another tuning could for example give a slightly smaller error reduction in x-direction, maybe resulting in slightly larger error reduction in the weaker y-direction. However, the tool performance is improved for cases 2A and 2B compared to case 1, especially in the y-direction. Since the x-direction is stiff compared to the y-direction, most of the errors in the x-direction can be compensated for already by the motor-angle errors, thereby it is hard to further improve the performance in x-direction. On the other hand, the accelerometer signal can give more information about the tool position in the y-direction than from the motor-angle measurements, which makes it possible to be able to improve the performance in that direction by using estimates of the tool position in the ILC algorithm.

Fig. 21. Case 2A: Error measure (31) on tool side.

Fig. 22. Comparison of error measure (31) on tool side for the cases 1, 2A, 2B, 3.

The choice of filter $Q$ follows the reasoning for case 2 with a need for a filter robust with respect to large model errors especially at the resonance frequencies, and having low-pass characteristics with learning up to 30 Hz to avoid learning of the high-frequency measurement noise. The tuning is based on the model (5) of the closed-loop system from tool reference to tool position. The choice of filter is then experimentally evaluated to give a good error reduction. In Fig. 23 the inverse $|Q^{-1}|$ is illustrated together with the relation $|1 - L(F_x + F_y T_{uv})|$ for the x- and y-direction, respectively, with $F_x = T_{xz}$, $F_u = T_{uz}$, and $F_y = 0$. The criterion (17) for monotone convergence for the ILC algorithm applied to the model (5), with the matrices $Q$ and $L$ given according to the implementation in Section 5.1, results in

$$\sigma(Q(I - LT_{uz})) \approx 0.95$$

when calculated using the number of samples in the learning trajectory and weighting of the ILC input signal.

The resulting performance of the robot tool is shown in Fig. 24, which can be compared to the nominal performance illustrated in Fig. 4(b), the result for case 1 depicted in Fig. 15 and to the result for case 2A illustrated in Fig. 20. The tool-side error measure (31) is shown and compared to the other cases in Fig. 22 and the mean of the error measure (31) of the tool for iterations 5 to 10 is given in Table 1. From these illustrations and figures, it can be seen that the tool position is slightly closer to the tool reference by a slightly larger tool-error reduction for
is based on measured motor angles, which are usually the only position measurements available by default for industrial robots. Second, the ILC algorithm is based on tool position estimates. The estimates are derived by using complementary or Kalman filtering, based on motor-angle measurements transformed by the forward kinematics and tool-acceleration measurements. Third, the ILC algorithm is based on the measured tool position. This approach is tested for evaluation purposes only, as it is difficult to measure the tool position in industrial applications. The tuning of the ILC filters includes learning above the resonance frequencies of the robot system. The stability is analysed with the aid of linear models identified in the specific working point chosen for all experiments.

The different ILC approaches are evaluated experimentally on a rectangular trajectory with a high-velocity reference. The ILC input signal is added with so-called lead-in/lead-out parts to the trajectory. The experiments show that ILC based on an estimate of the robot tool position, the controlled variable, improves the performance compared to ILC based on the motor-angles, usually the only measurements available by default.

9. CONCLUSIONS

In the paper three different approaches to ILC have been experimentally evaluated when applied to the Gantry-Tau parallel kinematic robot. First, the ILC algorithm

---

REFERENCES


T. Brogårdh. PKM research – important issues, as seen from a product development perspective at ABB Robotics. In PKM02.


M. Cescon, I. Dressler, R. Johansson, and A. Robertson. Subspace-based identification of compliance dynamics of parallel kinematic


PKM02. Workshop on Fundamental issues and future research directions to parallel mechanisms and manipulators, Quebec City, Canada, October 2002.


### Abstract

Three different approaches of iterative learning control (ILC) applied to a parallel kinematic robot are studied. First, the ILC algorithm is based on measured motor angles only. Second, tool-position estimates are used in the ILC algorithm. For evaluation, the ILC algorithm finally is based on measured tool position. Model-based tuning of the ILC filters enables learning above the resonance frequencies of the system. The approaches are compared experimentally on a Gantry-Tau prototype, with the tool performance being evaluated by using external sensors. It is concluded that the tool performance can be improved by using tool-position estimates in the ILC algorithm, compared to when using motor-angle measurements. In the paper applying ILC algorithms to a system following trajectories with so-called lead-in/lead-out is also considered, as well as dynamic modelling of the Gantry-Tau prototype.