ISSN 0280–5316
ISRN LUTFD2/TFRT--5814--SE

Subspace-based Identification of a Parallel Kinematic Manipulator Dynamics

Marzia Cescon

Department of Automatic Control
Lund University
April 2008

Department of Automatic Control
Lund University
Box 118
SE-221 00 Lund Sweden

Document name
MASTER THESIS

Date of issue
April 2008

Document Number
ISRN LUTFD2/TFRT--5814--SE

Author(s)
Marzia Cescon

Supervisor
Anders Robertsson
Rolf Johansson

Sponsoring organisation

Title and subtitle
Subspace-based Identification of a Parallel Kinematic Manipulator Dynamics
(Identifiering med subspace-metoder av dynamiken hos en parallell-kinematisk manipulator)

Abstract

This thesis deals with the identification of the dynamics of a Parallel Kinematic Manipulator, namely the Gantry-Tau patented by ABB located in the Robotics lab at LTH, Lund. The approach consider for modelling is subspace-based identification of linear models, where measurements from the robot motion are used to estimate the unknown parameters in the models. Rigid body dynamics and flexible body dynamics are taken into account and a description of the system in terms of a network with spring-damper pairs at the edges, representing the clusters, and masses at the nodes representing the end-effector and the carts, is proposed.

Key words

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title
0280–5316

ISBN

Language
English

Number of pages
60

Recipient's notes

Security classification

"La filosofia e' scritta in questo grandissimo libro che continuamente ci sta aperto innanzi agli occhi (io dico l'universo), ma non si puo' intendere se prima non si impara a intender la lingua, e conoscer i caratteri, ne' quali e' scritto. Egli e' scritto in lingua matematica, e i caratteri son triangoli, cerchi, ed altre figure geometriche, senza i quali mezzi e' impossibile a intenderne umanamente parola; senza questi e' un aggirarsi vanamente per un oscuro laberinto" *Galileo Galilei, Il Saggiatore*

Acknowledgments

The work resulting in this thesis was carried out at the Department of Automatic Control, Lund Institute of Technology, Lund, Sweden during Fall 2007 thanks to an agreement between the University of Padova, Italy, and Lund University, Sweden.

My stay at the Department has meant a lot to me. It's been a pleasure, a honor and an invaluable experience. Many people are aknowledged for having made this possible. First of all my supervisor Rolf Johansson and co-supervisor Anders Robertsson, for their guidance throughout the project, the great enthusiasm shown and all the time they have never refused to give me. I'm grateful to Leif Andersson, always patient in helping me with LaTeX typesetting, and Anders Blomdell for answering my questions at the early stage of my work.

Agneta Tuszynski, Britt-Marie Mårtensson and Eva Schildt played an important role making me feel so welcome and so comfortable warming up the atmosphere during the breaks.

Special mention deserves Isolde for all her precious help, Brad, responsible of my initiation to innebandy, Aivar for listening to me all the time, Oskar for his comments and Erik, Maria, Toivo, Ather, Peter and all the others who made me feel home...at work! Min korridoren...hoppas att vi ses snart i Italien!

I owe a lot to prof. Giorgio Picci, my supervisor in Italy, who certainly was and still is a solid reference to look at: thanks to make me discover the wonderful word of the Identification!

Finally, on a personal note, I would like to express all my love and gratitude to my family, the only persons on earth that could handle me during these long, sometime painful, but never vain years of studies, for being next to me whenever I fell down and for not keeping track of how many times this occured: you guys rule!

Marzia

Contents

1. Introduction

This chapter gives a brief introduction to the subject and summarizes previous work. Later the contributions and the outline of the thesis are presented.

1.1 Background

Industrial robots are nowadays widely used by the manifacturing enterprises in order to accomplish highly repetitive tasks saving costs, augmenting productivity and eliminating dangerous works. The increasing demands on extreme accuracy, high speed and modularity is leading to the growth of a new trend in robotics, Parallel Kinematics Manipulator (PKM). Moreover, cost reduction, reduced power consumption and safety issues come up with a lighter mechanical structure.

This thesis deals with the identification of the dynamics of a parallel robot, namely the Gantry-Tau, where measurements from the robot motion are used to estimate unknown parameters in the model. Identifying robot models is a challenging task since industrial robot are multivariable, nonlinear and resonant systems. Furthermore, in the case examined in this work, elastic effects of the material in the link structure have to be taken into account in addition to joint flexibilities.

A mathematical model that can cope with the above problems is, thus, of interest for the purpose of simulation and control of future, i.e. not yet observed, data.

1.2 Previous work

In [Johannesson *et al.*, 2005] the solution of the inverse kinematics and direct kinematics problems for a 3-degrees-of-freedom (DOF) Gantry-Tau parallel kinematic robot patented by ABB were obtained. The article [Dressler *et al.*, 2007] presents a newer kinematic and a dynamic model for the Gantry-Tau, the new kinematic model cosidering rotations of actuator axes which do not affect the end-effector orientation and the dynamic model being based on the assumption of parallel actuator axes and constant end-effector orientation. Elasticities or friction, however, are not included in the previous analysis. The thesis [Lyzell, 2007] presents an early attempt for identification of the dynamic model of the 3-DOF prototype located at the University of Queensland (Australia) using the least-squares method. As a result the estimation was not successful, since the estimated parameters did not have physical values. Furthermore the paper [Murray *et al.*, 2006] presents the inverse kinematics for the 5-axis variant of the Gantry-Tau parallel manipulator, achieved by extending the 3-axis machine with two linear prismatic actuators.

1.3 Motivation

System identification is needed for the determination of transfer functions from actuators to end-effector; transfer function in structural mechanics from end-effector loads to position and other force-to-displacement responses.

The approach considered for modelling is the subspace-based identification of linear models.

1.4 Outline

The report is organized as follows. Chapter 2 serves as background to the field of robotic manipulation. Chapter 3 presents the Gantry-Tau machine located in the Robotics Lab at Lunds Tekniska Högskola (LTH), Lund, the experimental set-up with particular attention to the vision system implemented and the software tools used throughout the work. In Chapter 4, a brief review of subspace-based identification is provided, later the experiments carried out are presented, the data obtained are examined and the models are estimated. Chapter 5 shows the result of the identification and gives estimation of the stiffness. Chapter 6 deals with the discussion of the achievements. Chapter 7 concludes the report and provides insights for future work. Finally, in the Appendix matemathical details are provided, numerical values for the parameters estimated are given, as well as code examples.

2. Robotics

This chapter deals with an overview of robotic manipulation. A concise description of the different type of manipulators is given pointing out benefits and drawbacks.

2.1 Introduction

Robot manipulators consist of a sequence of rigid bodies (*links*) connected by articulations (*joints*) actuated by motors. In the structure it is possible to identify an *arm*, which ensures mobility, a *wrist* that confers dexterity, and an *end-effector* that accomplishes the desired task. The section of the environment that can be reached by the end-effector is called *workspace*.

2.2 Serial structure

Serial manipulators are open-loop kinematic chain mechanisms consisting of a sequence of links, coupled one with another by a one-DOF joint. The end-effector is not constrained to the base. The high generality offered by this structure makes it the most used in the existing manipulators. It suffers, though, of low transportable load and poor accuracy, because of the mechanical architecture. Each of the segments, indeed, has to deal with the weight of the following link plus the load. The links must be stiffened and thus become heavier in order to cope with large flexure torques. Moreover, accuracy is affected by the error magnifications throughout the chain ([Merlet, 2000]). Serial disposition of the links is, thus, not suitable requiring manipulation of heavy loads and good position accuracy.

Figure 2.1 IRB 2400 robot

2.3 Parallel structure

Parallel robots are closed-loop kinematic chain mechanisms consisting of a mobile end-effector and a fixed base linked together by multiple kinematic chains. The number of degrees of freedom corresponds to the number of actuators that move the kinematic chains. Mechanical architectures are various according to their different applications. In this work, the attention is focused on a 3-DOF in translation manipulator. The advantages offered by the close-chain structure over its open-chain counterpart are several, as stated in [Merlet, 2000]. First of all, it ensures higher stiffness and consequently larger bandwidth for its control. Secondly, it provides better accuracy, skipping errors that propagate from one joint to the next in a serial manner. Thirdly, the higher payload is due to that the load is distributed over the chains and generally located close to the base. In addition to these the inverse kinematic problem is less difficult than in the open chain case, being solved independently for each chain. On the other hand, however, the drawbacks are noticeable. Namely, the forward kinematic problem is more complicated and the workspace volume is generally lower, leading to more restrictive motions. In the light of the previous comparison one can conclude that the parallel structure is preferable for those tasks that require high performances in terms of speed, acceleration, accuracy and stiffness and not a too large workspace. The research on parallel robotics, however, is a relatively new field in robotics and lots of issues need to be solved.

Figure 2.2 IRB 340 FlexPicker robot

3. The Gantry-Tau at LTH

In this chapter the robot system used in the work is outlined. A quick presentation of the Gantry-Tau is given, as well as a short description of the experimental set-up and the software tools adopted. Particular attention is, then, retained to the vision system.

3.1 Introduction

The Gantry-Tau robot is a parallel kinematic manipulator first presented in [Brogårdh, 2002], [Brogårdh *et al.*, 2005], [Johannesson *et al.*, 2005]. The structure located at LTH (see Fig.3.1) is based on an ABB patent and provides 3-DOF translational motion. The robot, thus, possesses three independent kinematic chains, each of these attached on one side to the ground and on the other side on the mobile tool-center-point (*TCP*) platform, actuated by three actuators. Its configuration overcomes the limitation of the workspace common to the parallel stucture. Constant end-effector orientation is guarateed by the tau-configuration, so that links belonging to the same cluster form parallelograms [Dressler *et al.*, 2007]. It is straight forward, then, that the benefits of the Gantry-Tau robot are many. First, the tool forces and moments are translated into pure axial forces in the manipulation links, which allows for a stiff and lightweight mechanical design. The lightweight and stiff design allows for high speeds, accelerations and tool forces. No redundancy in the structure brings to simple assembly and disassembly. Third, the Gantry-Tau has a workspace-to-footprint ratio larger or comparable to serial-type Gantry robots in contrast to other parallel kinematic machines ([Murray *et al.*, 2006]). The above characteristics made the Gantry-Tau structure particularly suitable for application with demands on high accelerations, accuracy, structural stiffness and modularity.

3.2 Experimental set-up

Each of the three clusters is actuated by an AC motor through a gearbox. The linear actuators are controlled via an ATMEL interface, which also takes care of the sensor interface, with measurement from resolvers on the arm-side, by a standard ABB IRC5 industrial robot controller (further details in [ABB, 2004]). The robot can be run directly from Matlab with the real-time extension developed at the Department of Automatic Control at LTH. A more detailed description of the structure of the ABB IRC5 control system is given in [Olsson, 2007]. Using the interface it is possible to access input-output signal of the controller and data from sensors mounted on the structure. Logging signals from the Simulink model can be done by an external program (further information in [Dressler *et al.*, 2007]).

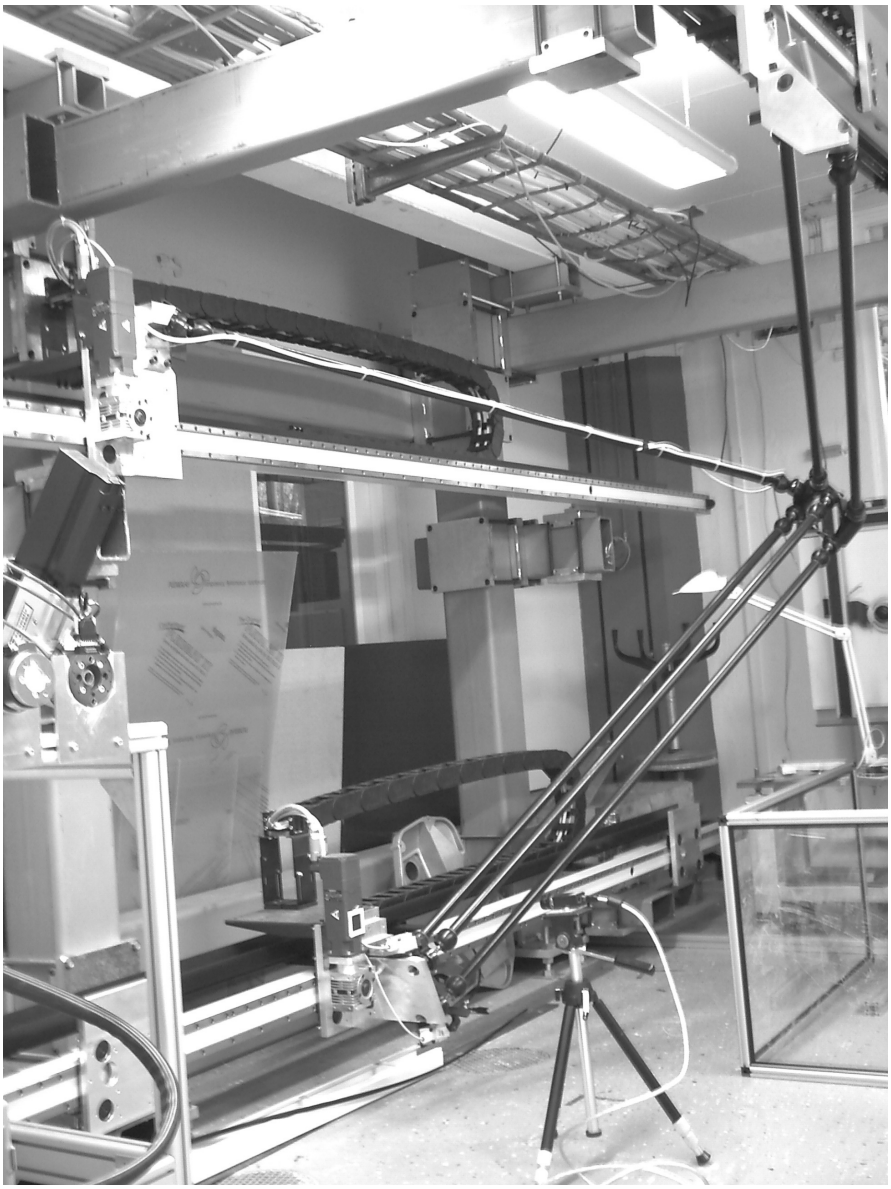

Figure 3.1 The prototype of a Gantry-Tau PKM in the Robotics Lab at LTH, Lund University

3.3 Software tools

The *Matlab* environment together with *SMI toolbox* by Haverkamp (see [Haverkamp and Verhaegen, 1997]), *LQG toolbox* by Gustafsson and *ID toolbox* by Chiuso, have been used extensively throughtout this work for data analysis, model estimation and all the simulations. The vision system running on the camera, instead, exploits the Open Source Computer Vision (*OpenCV*) written in *C++*.

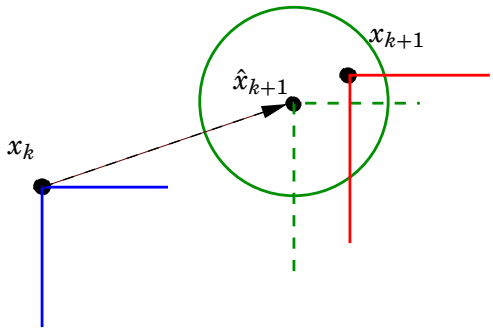

Figure 3.2 Given the position x_k of the cornerpoint in the current frame (blue) the search for the position x_{k+1} in the next frame (red) is done in a circle centered in the predicted $\hat{x}_{k+1}$ new position

3.4 The vision system

In order to record the motion of the robot a camera is used. It is a single isochronous *Basler A602fc* camera, running in *FORMAT7* with *MONO8* color mode, frame rate 250 Hz, and resolution 512 × 240 pixel. It makes use of *libraw1394*/*libdc1394* and the OpenCV library [Intel Corporation, 2008]. A simple paper-made square is attached to the object to keep track of (see e.g. Fig. 4.1) and it's traced by the camera during the motion with a sampling time of 4 ms. The corner tracker application written in C++ requires, during the initialization step, to choose the significant corners to be followed manually with the mouse and quit once the desired motion has been obtained. The feature position data are, then, logged in a Matlab m-file. Experiments undertaken at an early stage moving cart 3 back and forth along the x-axis keeping the y- and z-coordinates constant revealed the inaccuracy of the tracking in case of fast motion, relevant to the thesis purposes (see Fig. 3.4). Consequently, the algorithm needed to be improved. The core of the problem encountered stands in the OpenCV function *FindCornerSubPix*. That iterates to find the sub-pixel accurate location of each of the marked corners from the previous sequence of pictures. The search is performed for each point in a circle of radius 7 pixels centered in the corner's position on the previous picture. The process of corner position refinement stops either after a certain number of iterations or when a required accuracy specified by the user is achieved. As a result, in case of high speed, the routine worked poorly, missing the point's location and, hence, becoming useless. So, instead of looking for the new cornerpoint position in a neighbourhood of the cornerpoint position in the previous frame, a prediction of where the corner will be in the next frame based on the data available up to that point seems to be promising. Hence, the "search window" was centered in the predicted corner location (see Fig. 3.2). A standard one-step-ahead predictor for the estimation of the position was, thus, derived (full details are given in A.1) and implemented in C++ (code example can be found in C.2).

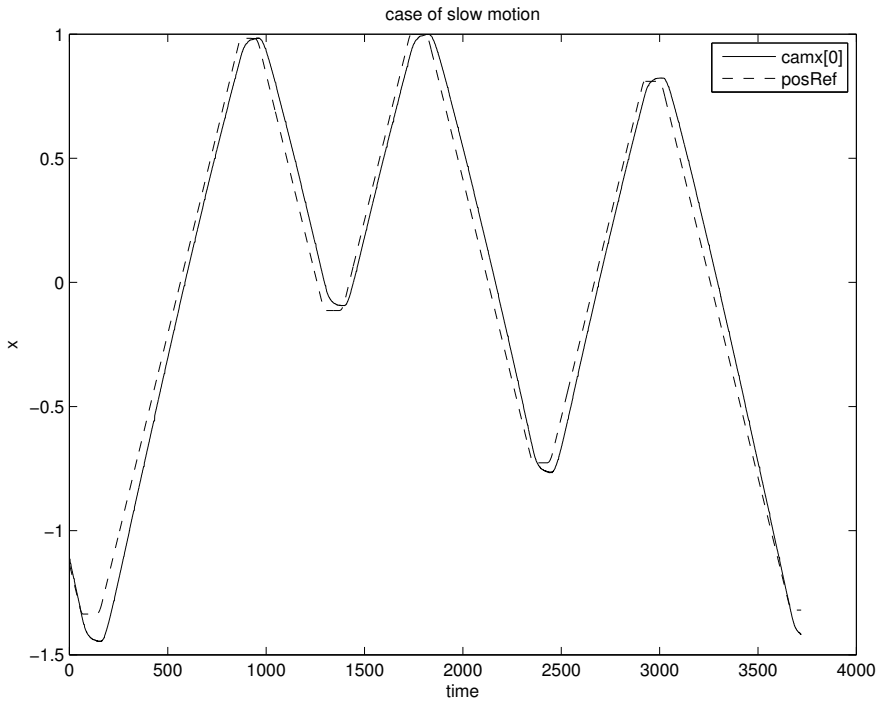

Figure 3.3 cart position from camera (solid line), reference to controller (dashed line) before the improvement

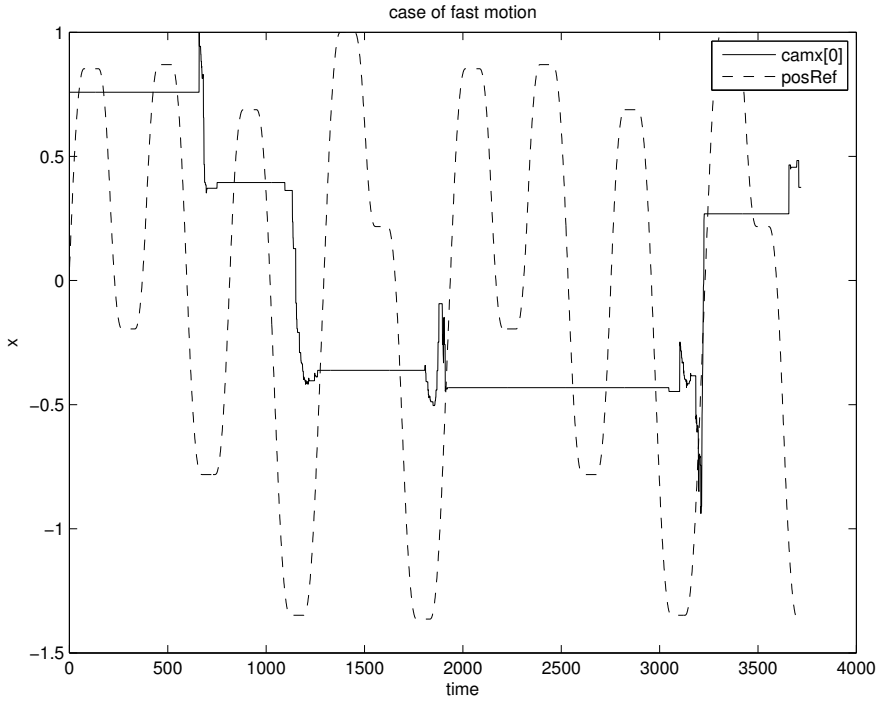

Figure 3.4 cart position from camera (solid line), reference to controller (solid line) before the improvement

4. Subspace-based model identification

This chapter deals with the subspace-based identification of the robot dynamics model. An early overview of system identification, in particular in the presence of input signals, is given. Experiments are, then, presented and different techniques are applied for identification.

4.1 Introduction

In general terms, system identification is a way of estimating mathematical models of dynamic systems from experimental data. The observable variables, namely inputs and outputs, are obtained performing experiments on the system and are measured at discrete instants of time. The procedure consists in finding a model that describes the measurement data sufficiently well. Two main attitudes exist towards the problem of fitting a model to the data:

♦ the *optimization approach*, i.e. parameters of a given model structure are obtained by minimization of a suitable cost function. These methods have been widely used and shown reasonably successful in modeling single-input single-output systems. A description of the above can be found in [Söderström and Stoica, 1989], [Johansson, 1993], [Picci, 2006].

♦ the *geometric, or "subspace" or realization-based approach*, in which the basic objects constructed in the algorithms are subspaces generated by the data, and geometric operations such as orthogonal and oblique projections are all what is needed to compute estimates of the parameters. For further information see e.g. [van Overschee and De Moor, 1994], [Haverkamp, 2000], [Chiuso and Picci, 2004c], [Picci, 2005].

This thesis considers the second class of methods, the main advantages being that the problem of finding local minima solving iterative nonlinear optimization and selecting the model structure and model order inherent in the classical parameter optimization approach are not required, all the operations are performed numerically in a reliable way using QR, SVD, QSVD from linear algebra, making these tecniques able to handle multivariate and complex systems.

An interesting approach to the identification of dynamic robot model mainly in the frequency domain is introduced in [Wernholt, 2007].

4.2 Preliminaries

This and the following section are meant to recall briefly some useful notations and important results to be used in the further analysis. Henceforth, boldface symbols will denote stochastic processes.

Given $\{ u_0, u_1, \cdots, u_N \}$, $u_t \in \mathbb{R}^p$ and $\{ y_0, y_1, \cdots, y_N \}$, $y_t \in \mathbb{R}^m$ observable input-output variables it will be assumed that they are sample paths generated by discrete-time zero-mean random signals in $t \in [t_0, +\infty)$,

namely $\mathbf{u} = \{\mathbf{u}(t,\omega)\}$ $p-$dimensional and $\mathbf{y} = \{\mathbf{y}(t,\omega)\}$ $m-$dimensional. A stochastic state-space realization of $\mathbf{y}$ with input $\mathbf{u}$ in absence of feedback from $\mathbf{y}$ to $\mathbf{u}$ can be given either in the form:

$$\begin{cases} \mathbf{x}(t+1) = A\mathbf{x}(t) + B\mathbf{u}(t) + G\mathbf{w}(t) \\ \mathbf{y}(t) = C\mathbf{x}(t) + D\mathbf{u}(t) + J\mathbf{w}(t), \end{cases} \quad \mathbf{x}(t_0) = \mathbf{x}_0, t \geq t_0 \tag{4.1}$$

where A, B, C, D, J are constant matrices, $\{\mathbf{x}(t)\}$ is the state process of dimension n, and $\{\mathbf{w}(t)\}$ is a normalized white noise process, such linear representations are infinite so it seems more convenient considering the so-called "innovation representation" which is unique up to change of basis:

$$\begin{cases} \mathbf{x}(t+1) = A\mathbf{x}(t) + B\mathbf{u}(t) + K\mathbf{e}(t) \\ \mathbf{y}(t) = C\mathbf{x}(t) + D\mathbf{u}(t) + \mathbf{e}(t), \end{cases} \tag{4.2}$$

where the white noise $\{\mathbf{e}(t)\}$ has the meaning of one step prediction error of $\{\mathbf{y}(t)\}$, given the infinite past history of $\{\mathbf{y}(t)\}$ and $\{\mathbf{u}(t)\}$ up to time $t-1$; or combining in parallel a stochastic model for the noise component

$$\begin{cases} \mathbf{x}_s(t+1) = A\mathbf{x}_s(t) + G\mathbf{w}(t) \\ \mathbf{y}_s(t) = C\mathbf{x}_s(t) + J\mathbf{w}(t), \end{cases} \tag{4.3}$$

and a deterministic model for the dynamic part

$$\begin{cases} \mathbf{x}_d(t+1) = A\mathbf{x}_d(t) + B\mathbf{u}(t) \\ \mathbf{y}_d(t) = C\mathbf{x}_d(t) + D\mathbf{u}(t), \end{cases} \tag{4.4}$$

so that $\mathbf{y}(t) = \mathbf{y}_s(t) + \mathbf{y}_d(t) = C[\mathbf{x}_s + \mathbf{x}_d] + D\mathbf{u}(t) + J\mathbf{w}(t)$, as proposed in [Picci and Katayama, 1996], [Katayama and Picci, 1999].

4.3 Identification in the presence of exogenous inputs

In the light of the above considerations it emerges clearly that identification in the presence of exogenous inputs can be done, in principle, following two different approaches, which essentially correspond to different choices of "model structures". On one hand, one could use stochastic realizations of $\mathbf{y}$ driven by $\mathbf{u}$ of the general form

$$\begin{cases} \mathbf{x}(t+1) = A\mathbf{x}(t) + B\mathbf{u}(t) + K\mathbf{e}(t) \\ \mathbf{y}(t) = C\mathbf{x}(t) + D\mathbf{u}(t) + \mathbf{e}(t) \end{cases} \tag{4.5}$$

Identification procedures based on this model will be referred to as "joint identification" and have been widely studied in the literature. Milestones in this area are the so-called *N4SID* ([van Overschee and De Moor, 1994]) and *PO-MOESP* ([Haverkamp, 2000]) families of algorithms. On the other hand one could, instead, consider models in block diagonal form such as

$$\begin{cases} \begin{bmatrix} \mathbf{x}_d(t+1) \\ \mathbf{x}_s(t+1) \end{bmatrix} = \begin{bmatrix} A_d & 0 \\ 0 & A_s \end{bmatrix} \begin{bmatrix} \mathbf{x}_d(t) \\ \mathbf{x}_s(t) \end{bmatrix} + \begin{bmatrix} B_d & 0 \\ 0 & K_s \end{bmatrix} \begin{bmatrix} \mathbf{u}(t) \\ \mathbf{e}(t) \end{bmatrix} \\ \mathbf{y}(t) = \begin{bmatrix} C_d & C_s \end{bmatrix} \begin{bmatrix} \mathbf{x}_d(t) \\ \mathbf{x}_s(t) \end{bmatrix} + D_d\mathbf{u}(t) + \mathbf{e}(t) \end{cases} \tag{4.6}$$

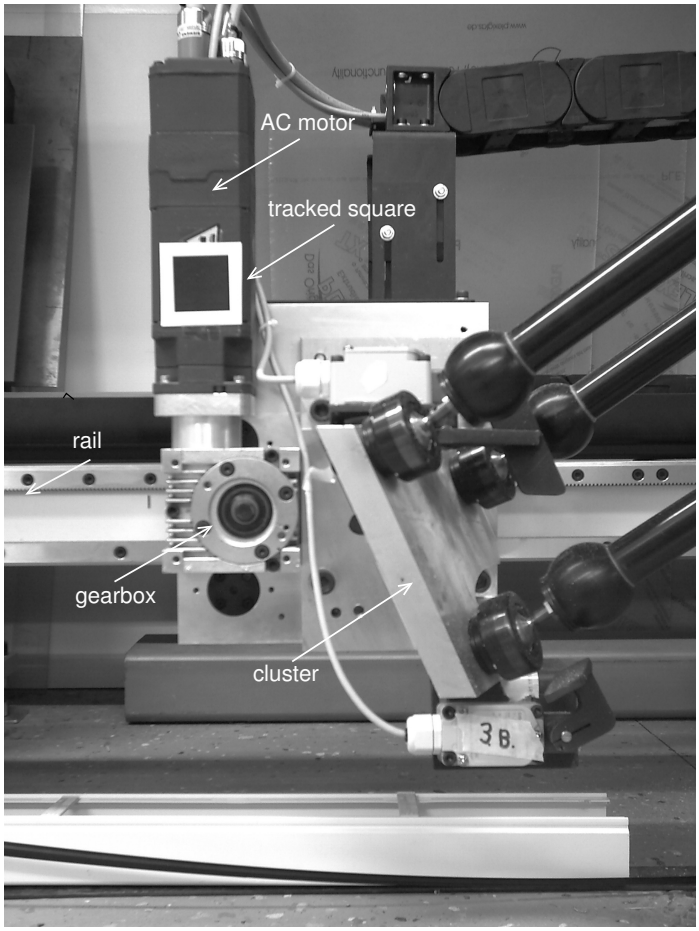

Figure 4.1 Cart 3

which is based on the preliminary decomposition of the state and output processes into the component lying in the input space (the "deterministic component") and its orthogonal complement (the "stochastic component"). For full details see [Picci and Katayama, 1996], [Katayama and Picci, 1999], [Chiuso and Picci, 2004c]). Identification based on models of this structure will be referred to as "orthogonal decomposition" approach.
In this thesis both of the approaches have been considered and evaluated.

4.4 Experiments

All the results in this thesis are based on experimental data. In order to obtain the best model as possible of the unknown system, the estimation problem needs to be well-conditioned, meaning that the signals in play have to contain "enough" information. In particular, substantial importance have the input signals, not only because they affect the consistence of the estimation but also because they determine which parts and modes of the system will be excited, thus identified.

The goal of an experiment during the whole work was having a set of measured positions (the output signals) covered by the square during the motion of the robot and the series of the references (the input signals) that

generated the motion. Hence, keeping in mind the above considerations, two different kinds of experiments were conducted: on the arm-side, having the markers on cart 3 as shown in Fig. 4.1, and on the TCP, recording the position of the end-effector (the generated input-output time series are shown in Appendix C).
From now on, the analysis will be conducted separately.

On the arm-side

Every measurement was carried out accomplishing the following consecutive steps:

- connect the simulink model of the controller (see Fig.4.2) to the PKM
- trajectory planning using the teach pendant
- submit ("listening", parameters values can be read but nothing is sent back)
- enable the vision system
- move the robot
- data logging in Matlab format

The trajectory was known beforehand and both cart position reference to the controller and cart position measured from the resolver on the arm were available. The schematic of the value reference path from main to axis computer used is presented in Fig. 4.3. One experiment has been selected for the identification procedures presented in Section 4.6 and named *Arm1*. It amounts to 2480 samples of the four marked cornerpoints x-positions, of the reference to the controller and of the measurement from the sensor.

On the TCP

The main intent of the second type of experiments was investigating the behavior of vibrations and resonances of the structure. In order to achieve that goal, markers were put on the end-effector and traced by the camera. Firstly, a motion was generated with the teach pendant for cart 3 in experiment *Arm2* making the end-effector move in both the y- and z-direction (in world coordinate frame). The experiment was the longest carried out throughout this work, providing 9920 samples of data, namely the cart position reference to the controller, the cart position measured by the sensor on the actuator and the x- and y- TCP coordinates in the TCP coordinate frame tracked by the camera. Secondly, with steady motors, sequences of impulses have been generated beating the end-effector. Two batches of data have been chosen for the estimation: *Pulse1*, which consists of a $[2480 \times 8]$ matrix, where the columns contain the x- and y-positions of the TCP, and *Pulse2* that contains in addition to the afore-mentioned positions also 2 columns of data representing the impulses applied along the z- and y- axis in world coordinate frame measured by the force sensor attached to the end-effector.

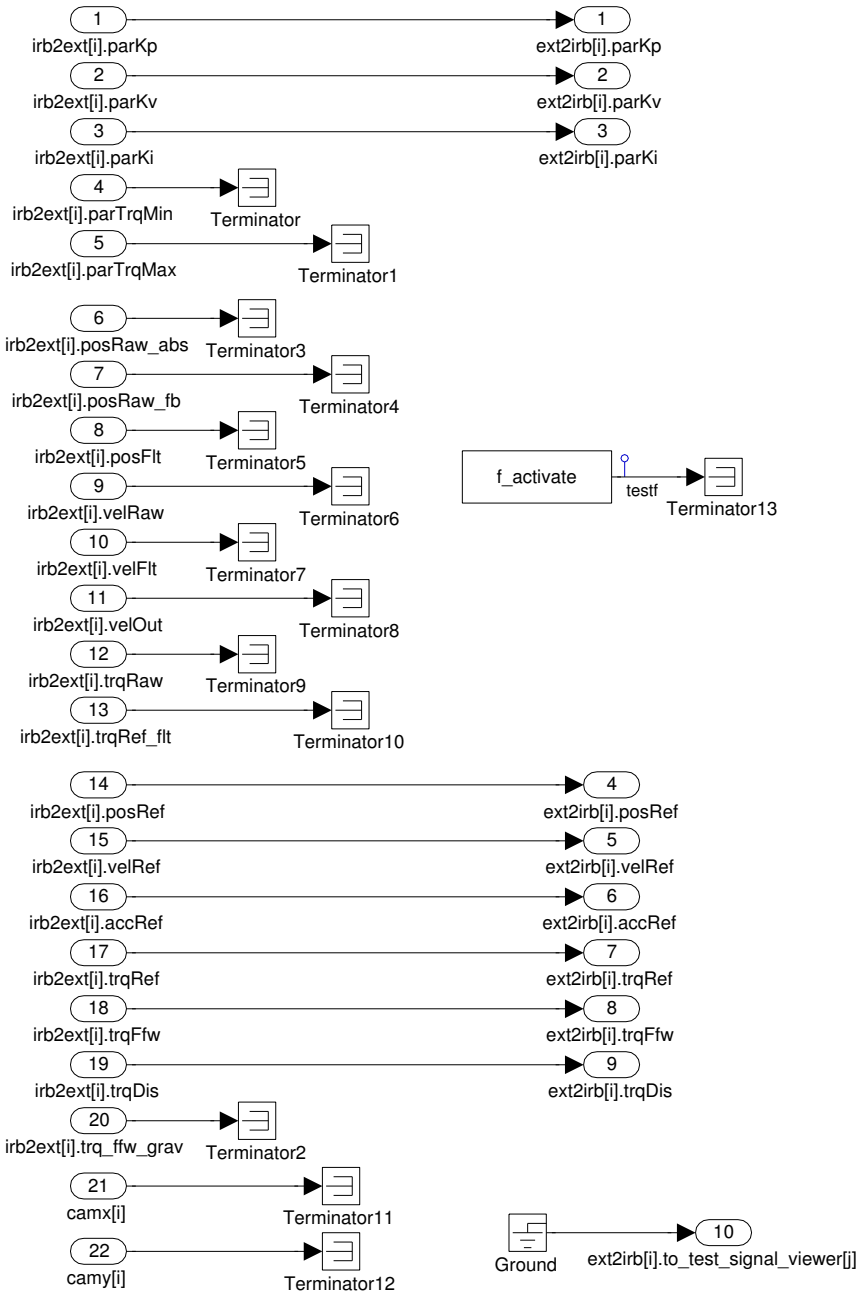

Figure 4.2 Simulink model (*GTvision*) of the robot system used for the experiments

4.5 Data Examination

As mentioned above, in system identification, a crucial role is performed by the input signal. A way to measure the excitation properties of a signal, is represented by its second-order statistical properties. A general but necessary requirement is the condition of persistent excitation of the input, so that, roughly speaking, all the modes of the system are excited during the identification experiment (for the definition see e.g. [Söderström and Stoica, 1989]). Depending on the methods applied to the data batch, several

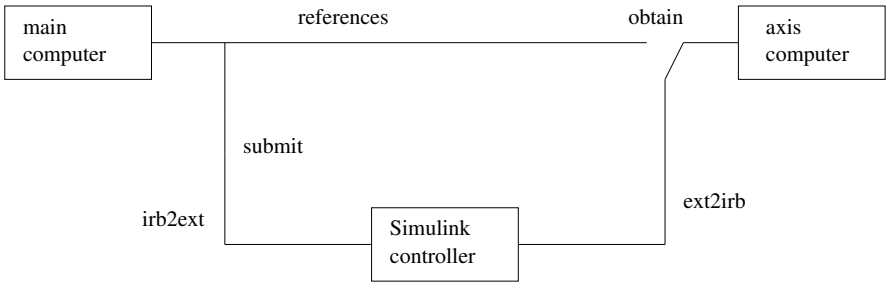

Figure 4.3 Sketch of reference value path from main to axis computer used in the experiments

terms measuring how “good” the experimental conditions are, have been derived, see the literature. In addition to this, as far as subspace-based identification is concerned, the geometric nature of the problem has to be taken into account. Indeed, it has been shown in [Jansson and Wahlberg, 1997] and [Jansson and Wahlberg, 1998] that the critical relation for consistency of 4SID methods involves the extended observability matrix so that for systems with process noise a persistence of excitation even of any order is not sufficient. Recently, it has been argued that the bottleneck for the performances of the standard subspace methods is how “parallel” are the rowspaces of the past signals (past input and output) and future input ([Chiuso and Picci, 2004b], [Chiuso and Picci, 2004a]). According to this, large errors on the computation of the parameters of the model occur when some (canonical) angles are nearly zero. These facts are to be considered in Chapter 6.

In this section the attention will be focused on the second-order statistical properties of the generated time-series, namely power spectral density to investigate the "richness” of the signals and coherence spectrum, to test the signal-to-noise ratio and linearity between input-output.

On the arm-side

Concerning *Arm1*, it was decided to use as input the cart position reference signal from the ABB controller and as outputs the cart position from the vision system represented by one of the corners x-coordinate and the motor position measured by the encoder on the arm. Spectral analysis was done on detrended input-output data and gave promise of a successful identification at low frequency. Indeed, the analysis of the coherence function suggested that a noise source affected the measurements at high frequency, thus the parameter estimation might be poor (see Fig. 4.4). However, regardless the lack of optimal experimental condition, an attempt of identification will be done.

On the TCP

Dealing with *Arm2* data batch, the input signal chosen, namely the reference value from IRC5, and the output signals, namely cart position reference from the sensor on the arm, x- and y-coordinates of the first corner marked on the end-effector, might not be properly described by a linear model, as can be noticed from the coherence spectra in Fig 4.5.
As for *Pulse1* only one over the four tracked cornerpoints was chosen for

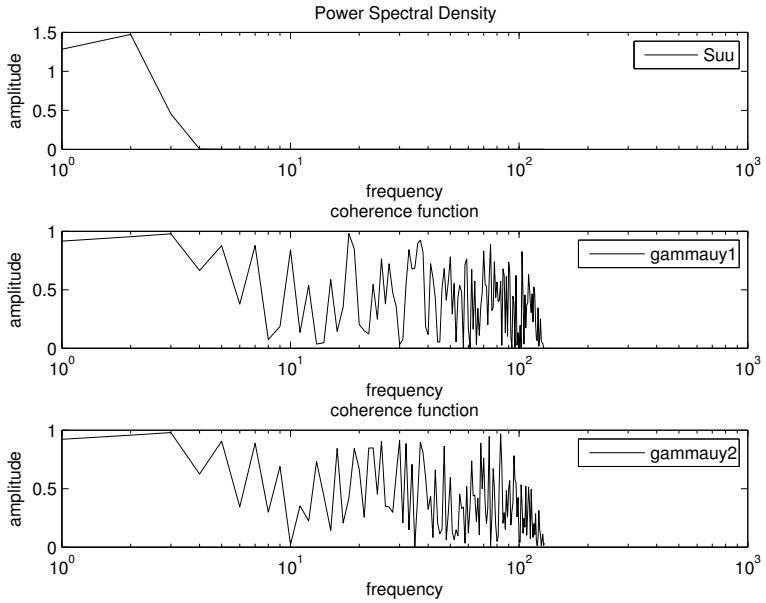

Figure 4.4 *Arm1*: spectral analysis on data

the further analysis. There are no available techniques that could have handled more than one output for the identification, therefore it was decided to continue only with the TCP x-position, and specifically with the samples 1569 : 2010. The mean value and all the other linear trends were removed from raw data obtained from the vision system. Besides, as can be noticed in Fig. C.3, the first 230 samples might be regarded as a noise term to be considered later on in the analysis. The spectrum highlighted a strong component in a neighborhood of 90 Hz (see Fig.4.7). A method of spectrum estimation based on the Fourier transform will have to be used, hence problems of spectral leakage might arise from the finite observation interval. In order to prevent a possible spectral distortion a window function, namely the Hamming window, was applied to the already detrended data. Referring, instead, to experiment *Pulse2*, it was possible to study the spectral properties of the forces applied to the TCP thanks to the force sensor. As can be seen in Fig. 4.10 and Fig. 4.11 the input signals generated seem to be sufficiently exciting.

4.6 Model determination

On the arm-side

A first attempt of model determination was done applying the N4SID algorithm, using the canonical variable algorithm by Larimore for the determination of the weighting matrices in the singular value decomposition. A fourth-order model was obtained. However, although it has been argued in [Jansson and Wahlberg, 1997] that in certain conditions resembling those encountered herein the estimated parameters cannot be considered reli-

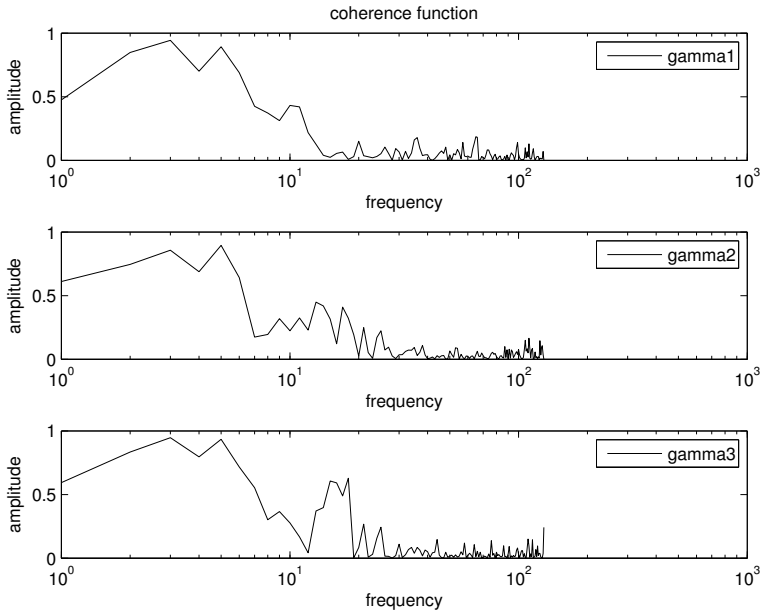

Figure 4.5 *Arm2*: coherence spectra: between refence and x-position (gamma1), reference and y-position (gamma2), reference and value from sensor (gamma3)

able, this effort could give an insight of the state variable order. Next, a second model with the same order but different properties (see Chapter 5.1) has been derived with PO-MOESP. Concluding, due to the observation on the coherence function of input-output made in Section 4.5 it was considered relevant providing a description of the noise component. Thus, the orthogonal decomposition approach was followed, and a seventh-order model for the disturbance was determined.

On the TCP

Concerning experiment *Arm2* the best realization obtained was a 5-th order model achieved with PO-MOESP algorithm. However, even if the orthogonal decomposition approach was not successful, the investigation of the stochastic submodel gave valuable intuition on the disturbance characteristics (see for instance Fig.5.4).
As already pointed out in Section 4.5, in case of experiment *Pulse1* a non-parametric method was applied leading to a first crude model obtained in the form of a Bode plot (see Fig. 4.8). Making use of the discrete Fourier transforms of input and output data, the following formula brought to the estimate:

$$\hat{H}(e^{j\omega h}) = \frac{Y_N(j\omega)}{U_N(j\omega)} \tag{4.7}$$

The second attempt was done with the Ho and Kalman realization algorithm, in its modified version by Juang and Pappa. This is a special case of a subspace technique, namely one in which the input is an impulse. A 9-th order system was estimated. Afterwards, a reduction to order 6 was done with balanced truncation. When the force sensor was mounted and calibrated in *Pulse2* it was possible to apply subspace algorithms, namely or-

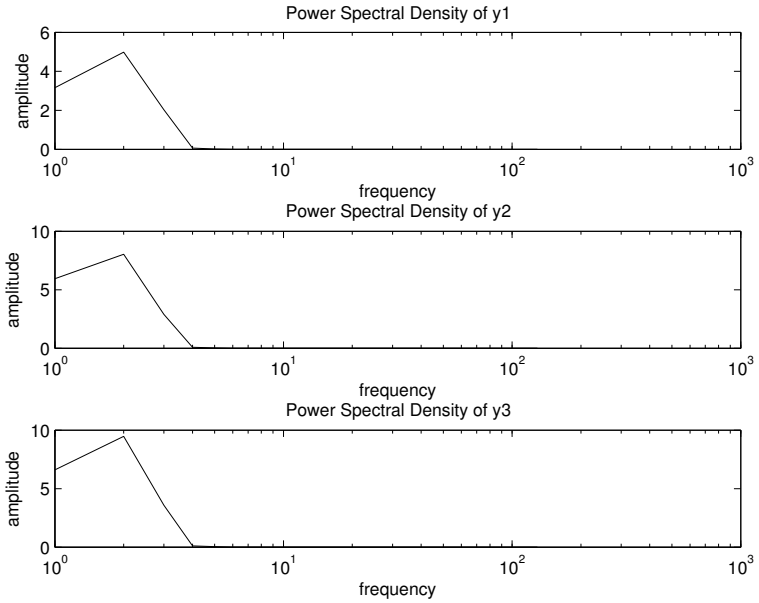

Figure 4.6 *Arm2*: input-output spectra

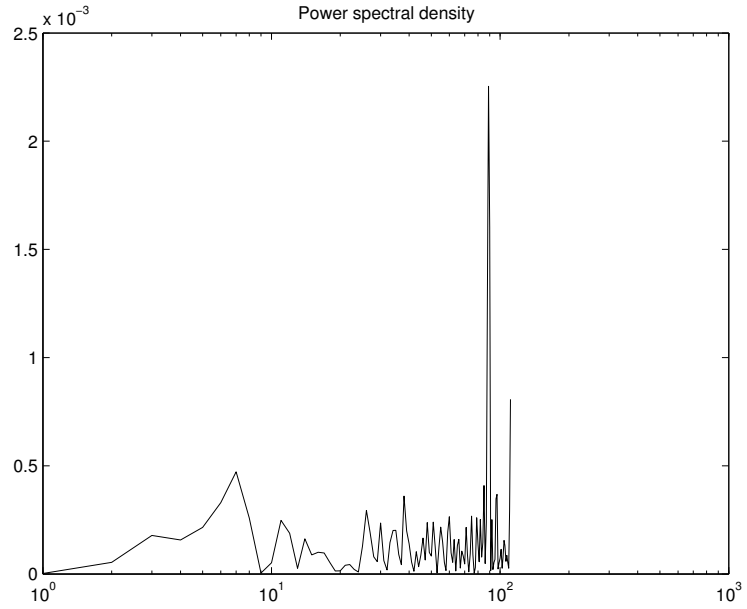

Figure 4.7 *Pulse1*: spectrum of the first 230 x-position samples. Notice the spectral peak at 90 Hz

thogonal decomposition approach and joint approach, considering the forces applied to the end-effector along the y- and z-direction as input, and the selected TCP cornerpoint's x- and y-positions as output.

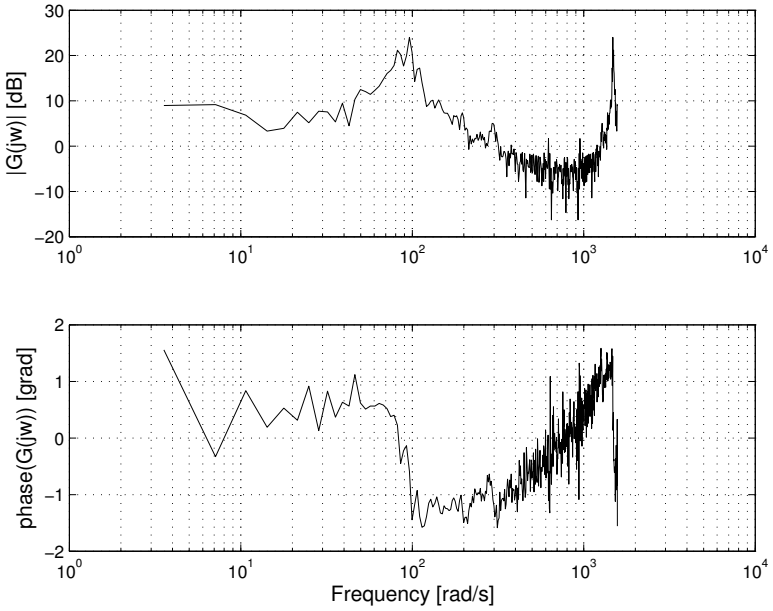

Figure 4.8 *Pulse1*: Bode diagram obtained using spectral estimation

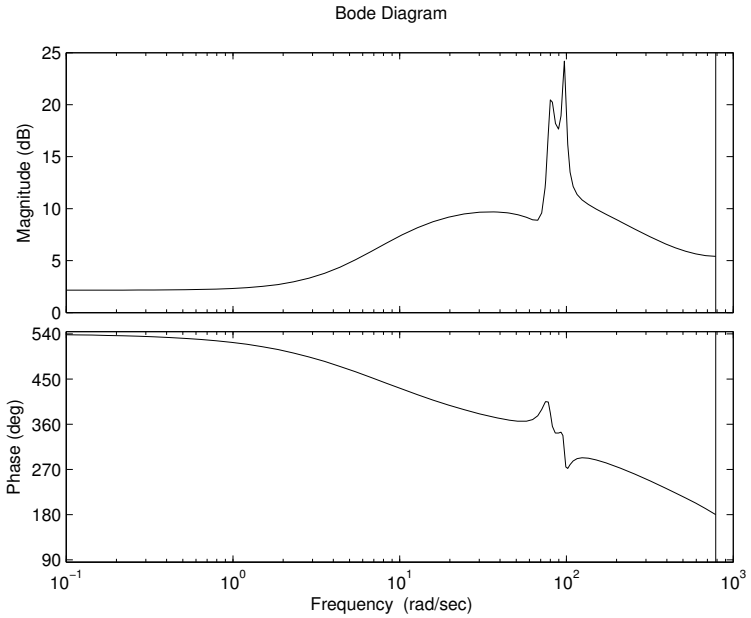

Figure 4.9 *Pulse1*: Bode diagram of the model obtained using Ho-Kalman realization

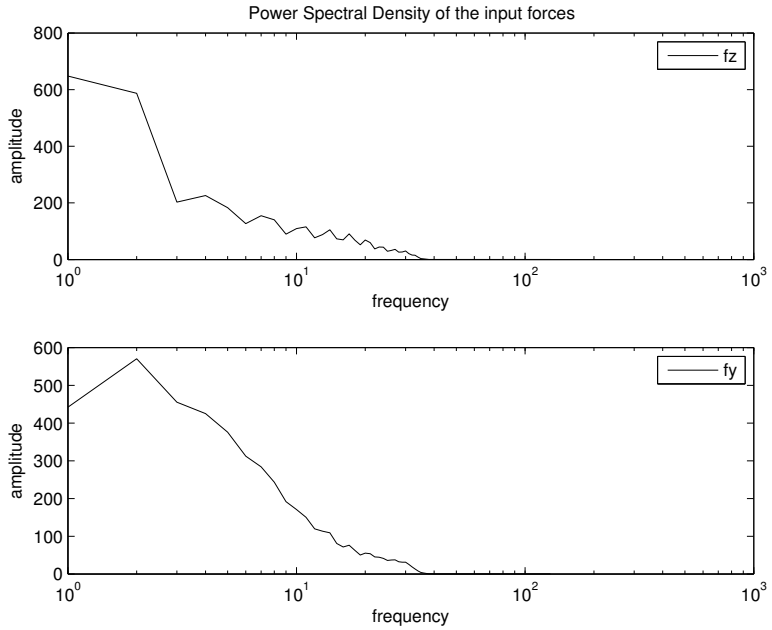

Figure 4.10 *Pulse2*: spectra of input signals

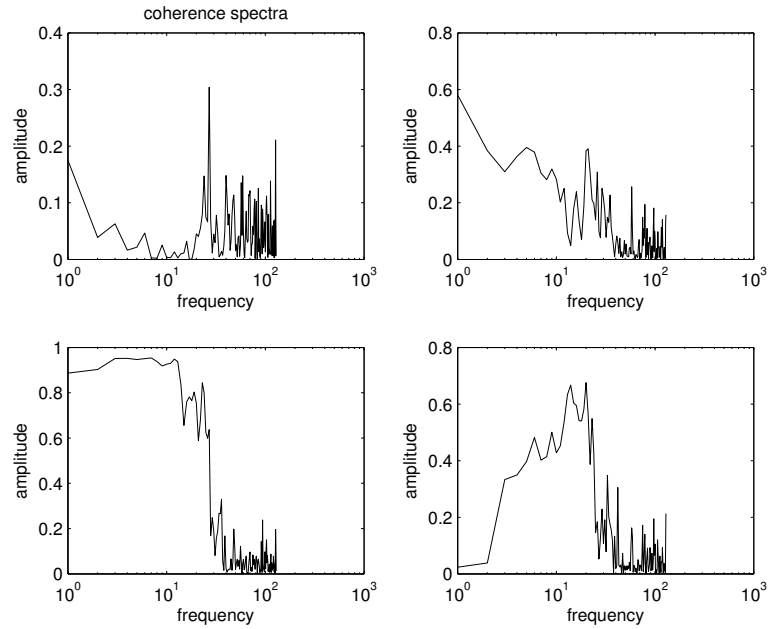

Figure 4.11 *Pulse2*: in clockwise order coherence spectra between force along z-direction and TCP x-, y-position, force along y- direction and TCP x-, y-position

5. Results

The purpose of this chapter is to present the models obtained in Section 4.6. The numerical values of the parameters are shown in Appendix B. In addition, an attempt of physical parametrization will be made.

5.1 Models obtained

On the arm-side

Regarding the joint model identification of *Arm1*, clearly visible flexibilities of the link emerge from the analysis of the poles of both systems derived in Section 4.6. Moreover, pleasing properties are shown by the stochastic model of the noise component: poles appear at the frequency of nearly 100 Hz (626 rad/sec) and 60 Hz (383 rad/sec). It will be shown in Section 5.1 that the first disturbance contribution is recurring.

On the TCP

Referring to the model estimated for *Pulse2* this may provide some information on the natural frequencies of the beam's oscillations during the motion. The rather poor experimental conditions, however, will affect the estimation, so that the results of the identification have to be considered with caution. As for experiment *Pulse1*, looking at the Bode plot of the transfer function (4.7) obtained via spectral analysis (Fig. 4.8) it might be inferred that the system has some dynamics at 15 Hz. The behavior at higher frequencies is not clear, in particular the phase plot is meaningless. This is not surprising, if related to the noise observation made in Section 4.5. Furthermore, it can be recalled that such a way of proceeding might be affected considerably by the disturbance contribution to $Y_N(j\omega)$ and thus $\hat{H}(e^{j\omega h})$.

As for the Ho-Kalman realization a 6-th order system can be considered reasonable from a mechanical point of view, as each of the three clusters may be described by two states. The same dynamics guessed before emerge from the analysis of the Bode diagram (see Fig. 4.9). Moreover, the resulting system is stable.

As far as experiment *Pulse2* is concerned, the orthogonal decomposition approach gives a second order system for the deterministic part, with components at 155 Hz, and a fourth-order system for the stochastic one, with the disturbance acting at 14 Hz (86 rad/sec) and at 100 Hz (633 rad/sec). The total order of the system is 6. It can be noticed quite clearly, that this constitutes a double-check of what has been argued previously. Dealing with joint model estimation, instead, different results have been obtained. The most reasonable one seems to be a fourth-order system, obtained by means of the joint approach applying subspace id, PO-MOESP and N4SID algorithms. Although the dynamics present another frequency range, the noise is captured.

5.2 Validation

The models presented in Section 5.1 were validated according to the following criterions:

Pole-zero plot A pole-zero plot indicates if the model order is adequate. There may be poles and zeros located close together, suggesting that model reduction should be performed. Furthermore, if the disturbance seems to affect the process at certain frequencies, there should appear resonant poles in the noise model with corresponding frequencies.

Validation by test on the residuals Residual tests can be done to find remaining correlations which point out whether the model is appropriate or not. Ideally the residuals should look like white noise of sufficiently small magnitude.

Validation by simulation Purpose of the validation by simulation is testing whether a model can reproduce the observed output when fed by the actual measured input.

On the arm-side

Pole-zero plot Each of the joint models estimated is characterized by a pair of complex poles within the unit circle, corresponding to a frequency of, respectively, 13 Hz and 14 Hz. These can be considered the frequencies of the beam's oscillations during the motion. The low-frequency range can be explained by the rather low velocity planned for the experiment.

Validation by test on the residuals Residuals can be calculated from (A.10) using the innovations model (A.9) of the systems obtained with N4SID and PO-MOESP methods. The covariance function of the estimate $\hat{\mathbf{e}}(k)$ of the innovation $\mathbf{e}(k)$ i.e. the one-step-ahead prediction error, occured to be good enough in case of the PO-MOESP model (see Fig. 5.1). As for the N4SID, instead, it suggested that measured position data need to be better exploited.

Validation by simulation Cross validation, where the validation data are used for simulation, was used to compare real and estimated data obtained from the joint models. Results are shown in Fig. 5.2 and Fig. 5.3. The behavior was slightly different and apparently unexplained. For the simulation of the model obtained with the orthogonal decomposition, the state-space realization given in (A.17) was considered for the implementation. Simulation on validation data was comparable to the one depicted in Fig. 5.3.

On the TCP

Pole-zero plot The system describing the data from *Arm2* presented a pole placed close to the unit circle and a pair of complex poles with frequency 20 Hz. As already noticed in 5.2 these poles might be related to flexibilities and oscillations of the beams. The disturbance system shows poles at the frequency of 102 Hz (see Fig. 5.4).

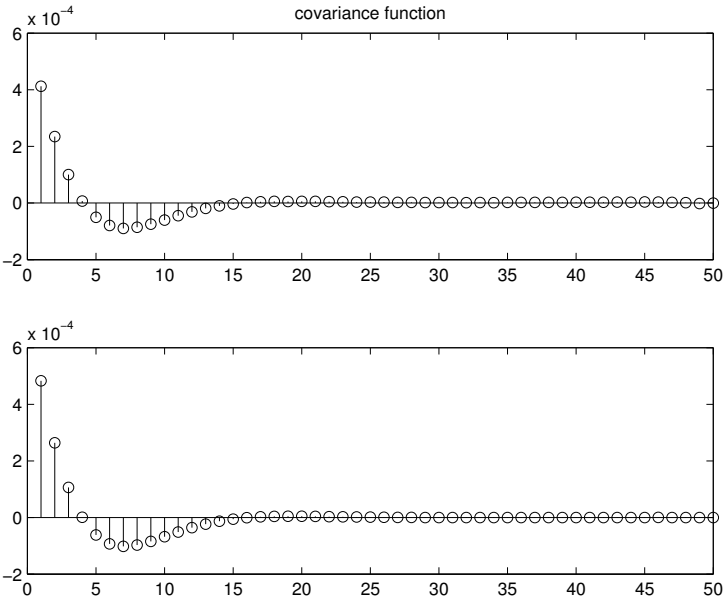

Figure 5.1 *Arm1*: covariance function of innovation's estimate, of the x-position on top and on the feedback from sensor on bottom

As for *Pulse1*, the pole-zero map of the 9-th order system (Fig. 5.5) showed possible cancellations. This suggested it might be appropriate performing a model reduction. Thus, a reduced-order approximation was computed. Simulations were carried out with order 6 revealing an acceptable behavior (see Fig. 5.2).

As for the model of the noise component of *Pulse2* poles were located at the frequencies 100 Hz, 14 Hz, as expected. Furthermore, dealing with joint system identification, the choice of a fourth-order model seemed to be suitable for the description of both the dynamic and the disturbance (see Fig. 5.7).

Validation by test on the residuals As far as *Arm2* is concerned, a residual sequence has been computed from (A.10) using the innovations model (A.9) of the system obtained with the joint system identification. Whereas it seems that the evolution of the signal corresponding to the x-position is well explained, the same can not be said for the one describing y. Indeed, the end-effector was oscillating along the y-axis rather independently from the reference signal (Fig. 5.8).

The covariance function of the one-step-ahead prediction error of the model obtained with PO-MOESP method in *Pulse2* turned out to be acceptable (see Fig. 5.9).

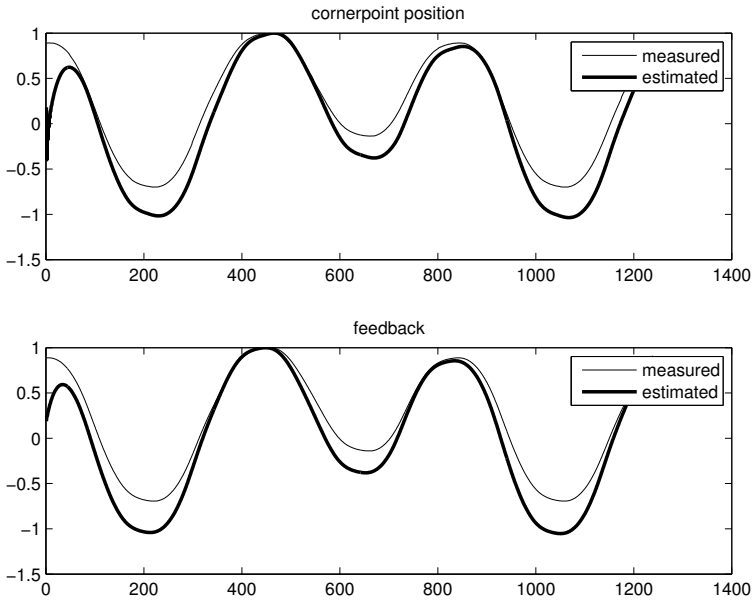

Figure 5.2 *Arm1*: simulation on validation data N4SID 4th order

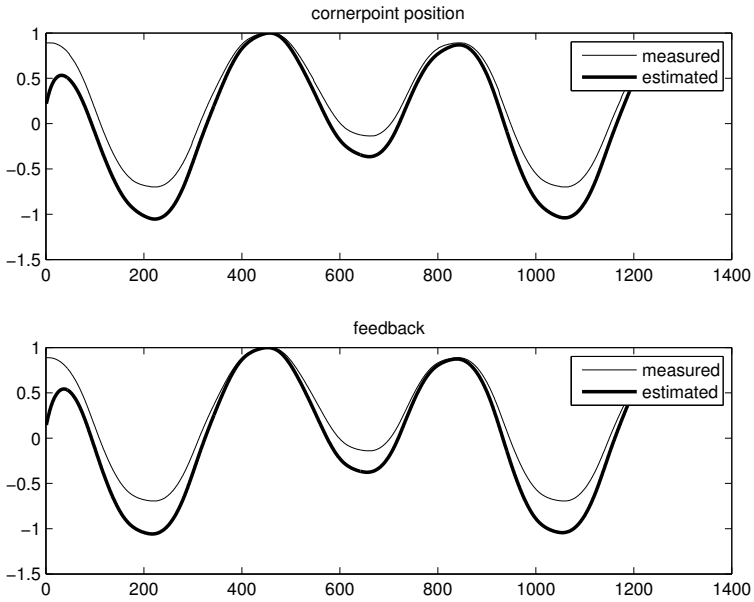

Figure 5.3 *Arm1*: simulation on validation data MOESP 4th order

Validation by simulation The cross validation made for the 5-th order model that describes *Arm2* is shown in Fig. 5.10. As for the model of data, the set *Pulse1* estimated with the Ho-Kalman technique the behavior is satisfactory (see Fig. 5.11). However, as already mentioned, a reduced-order model with good performance was obtained, see Fig. 5.12. Concerning, instead, the simulation of the decoupled model of *Pulse2* it is clearly noticed that the dominating dynamics of the true system were well captured (see

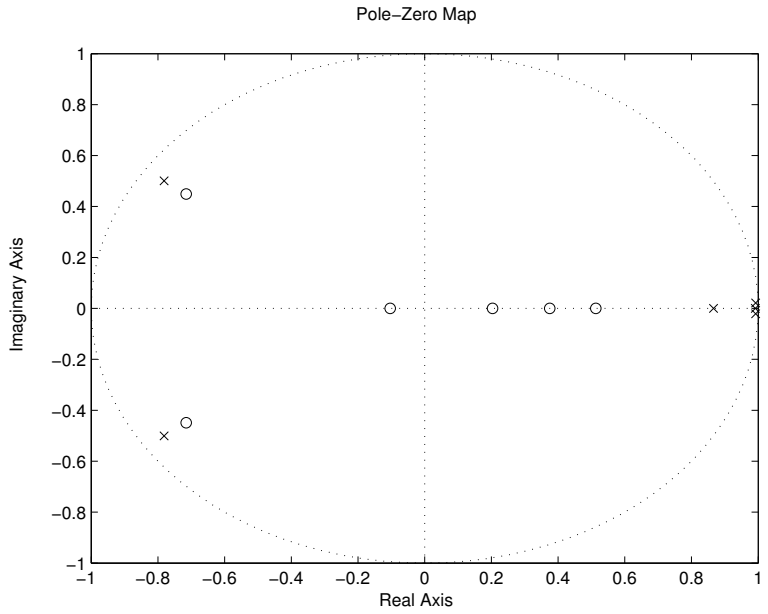

Figure 5.4 *Arm2*: pole-zero map of the stochastic submodel

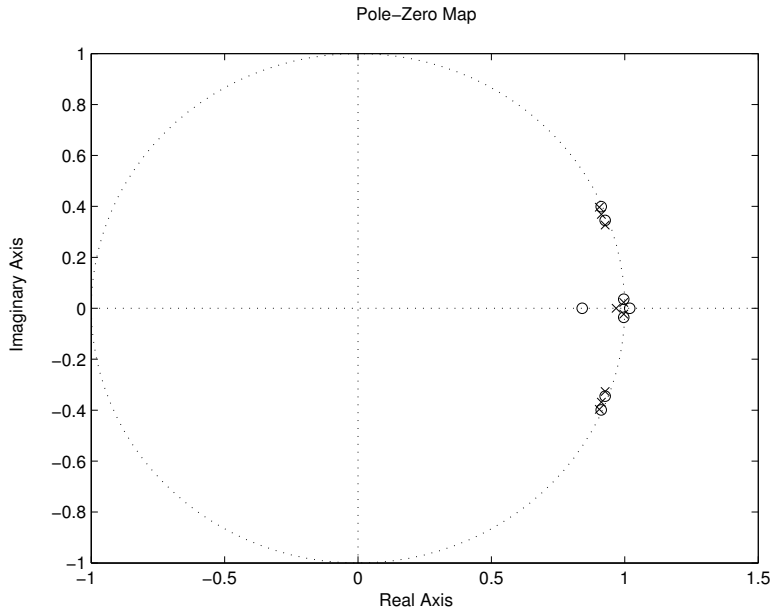

Figure 5.5 *Pulse1*: pole-zero map of the system obtained with the Ho-Kalman realization tecnique

Fig. 5.13), except for samples 2500 : 2600, where the identified system has opposite behavior compared to the true one with respect of the x-position. Regarding this, joint identification seems to perform better, see Fig.5.14.

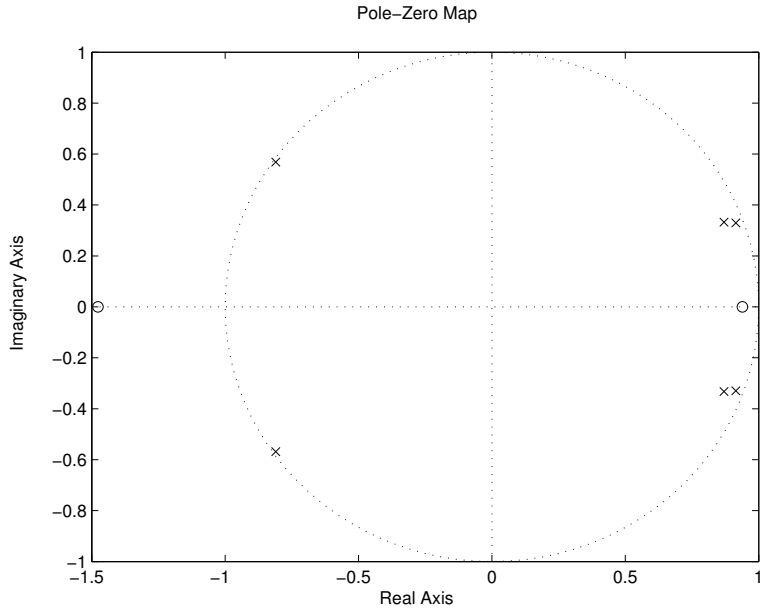

Figure 5.6 *Pulse2*: pole-zero map of the system obtained with the orthogonal decomposition

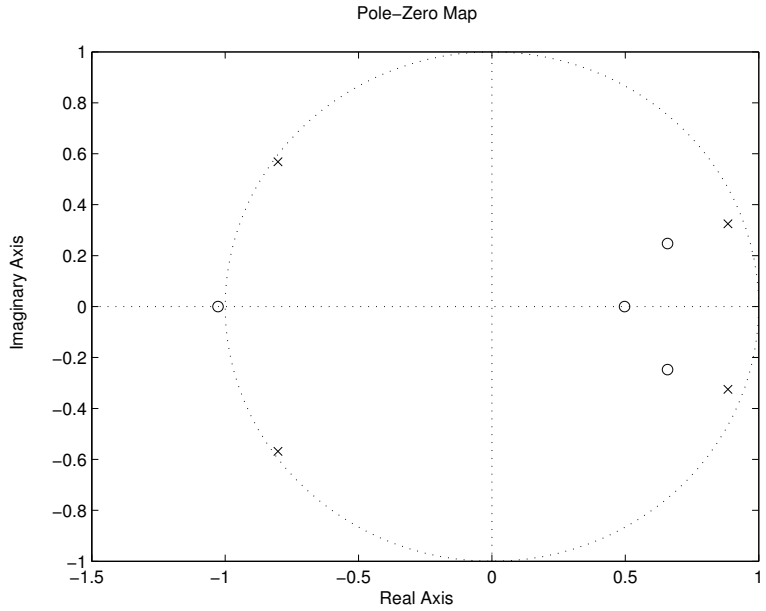

Figure 5.7 *Pulse2*: pole-zero map of the system obtained with the joint identification, namely *subspace id*

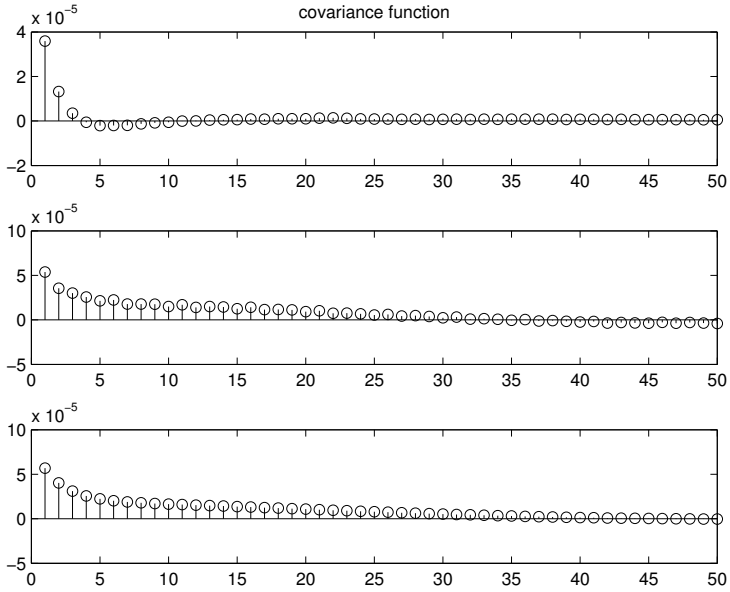

Figure 5.8 *Arm2*: covariance function of the one-step-ahead prediction error: for TCP x-position (top), y- position (center), and cart position from resolver (bottom)

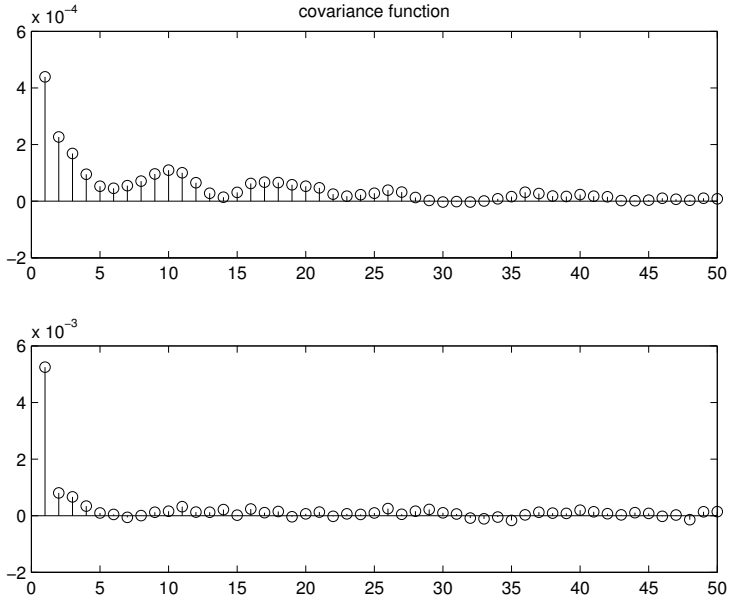

Figure 5.9 *Pulse2*: covariance function of the one-step-ahead prediction error obtained with PO-MOESP: for TCP x-position (top), and y-position (bottom)

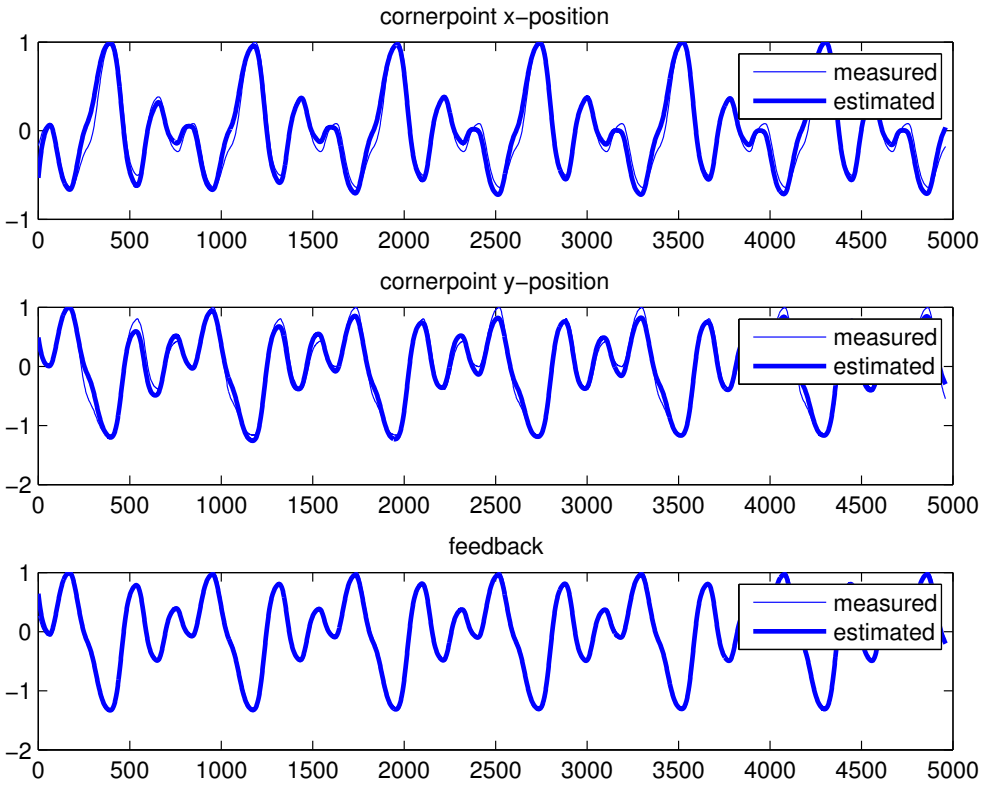

Figure 5.10 Simulation for *Arm2*

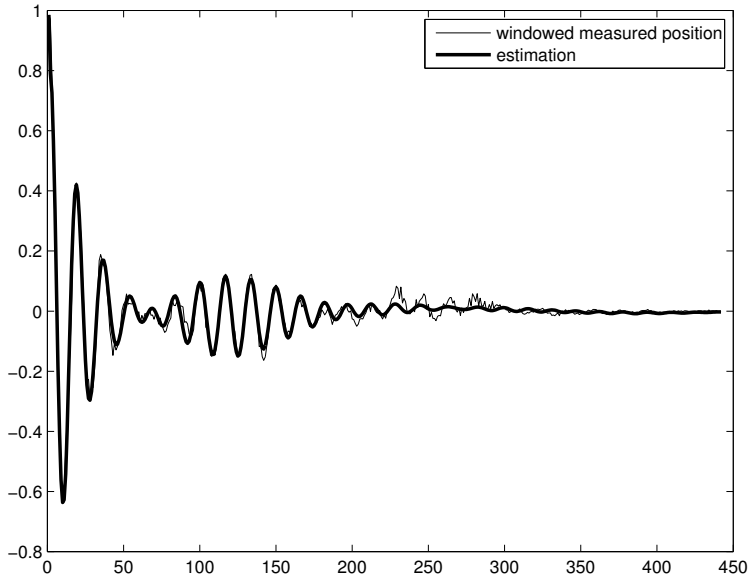

Figure 5.11 *Pulse1*: simulation of the system obtained via Ho-Kalman realization technique

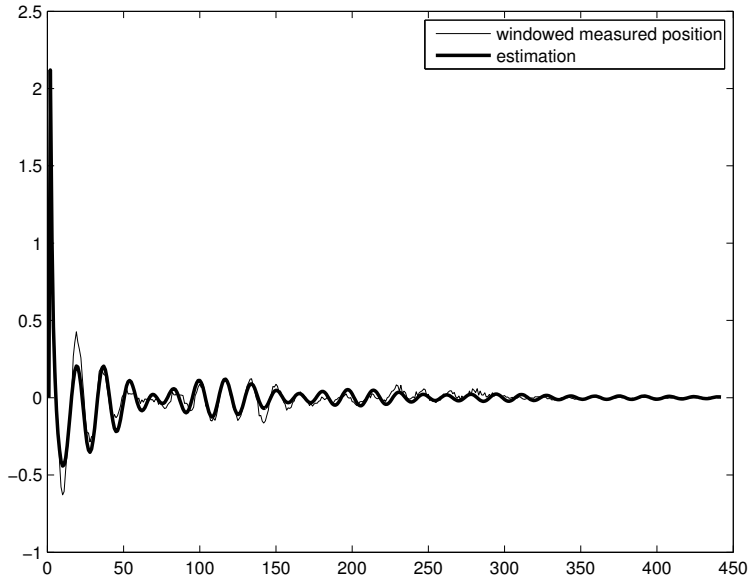

Figure 5.12 *Pulse1*: simulation of the reduced approximation of the system obtained via Ho-Kalman realization technique

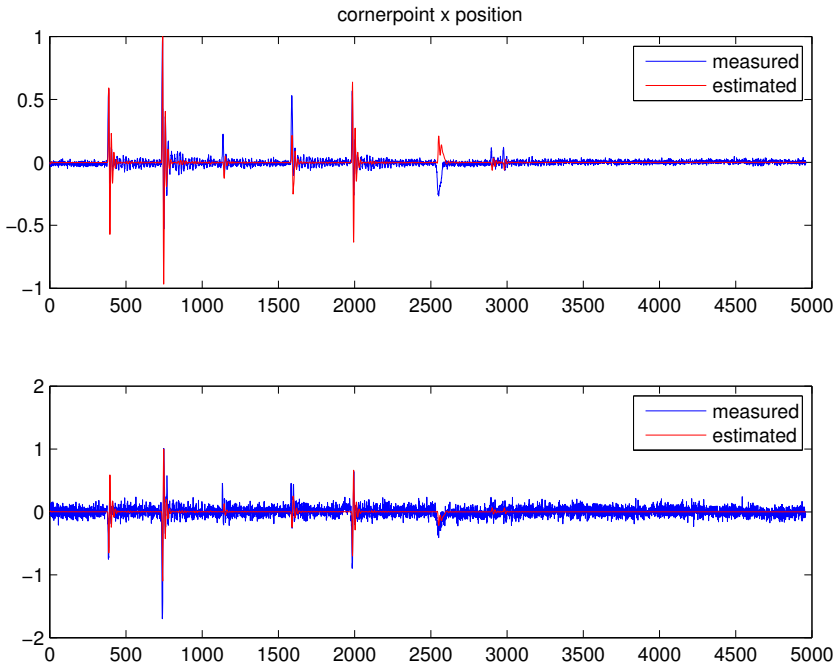

Figure 5.13 *Pulse2*: simulation of the system obtained with the orthogonal decomposition

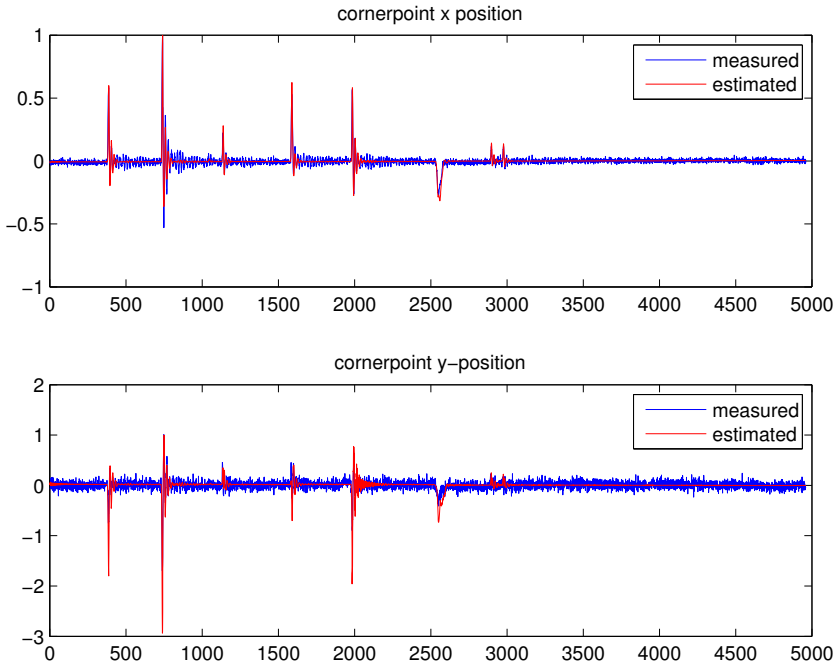

Figure 5.14 *Pulse2*: simulation of the system obtained with subspace id

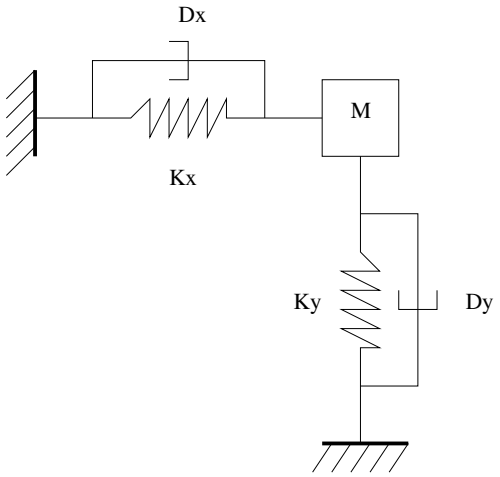

Figure 5.15 The spring-damper flexible joint model

5.3 Stiffness estimation

As already pointed out in Chapter 1 and Chapter 3 the lightweight mechanical structure of the Gantry-Tau introduces significant flexibilities and elasticities in the links. Considering the two spring-damper pairs as in Fig. 5.15, the afore-mentioned effects can be modeled. It is straightforward, then, to provide a description in a quantitative way of the spring stiffness and damping.

Herein a first attempt of stiffness estimation was done following the procedure presented in A.4. Raw measurements, i.e., not detrended data, from experiments *Pulse2* were taken into account and a linear regression problem of the type

$$\mathcal{Y} = \Phi\theta, \quad \theta = \mathcal{K}^T \tag{5.1}$$

was solved by means of least-squares identification (see e.g. [Söderström and Stoica, 1989]), where the observed variables were the forces applied to the end-effector and the regressors were the deformations along the x- and the y-axis (in the end-effector coordinate frame). The numerical value obtained was:

$$\mathcal{K} = 10^3 \begin{bmatrix} -3.7326 & -1.5671 \\ 6.9209 & 0.5560 \end{bmatrix} \tag{5.2}$$

It can be noticed that the matrix is positive definite with eigenvalues $10^3 \cdot (-1.5883 \pm 2.4996i)$ lying on the left-half plane thus it is in perfect alignement with its physical meaning.

5.4 Physical parametrization

The overall structure of the Gantry-Tau can be seen as a network with spring-damper pairs at the edges (the clusters) and masses at the nodes (the carts and the end-effector). An external force acting on a node (in the case examined acting on the end-effector) will result in deformations and elongations on springs and dampers. Thus, it is of interest detecting

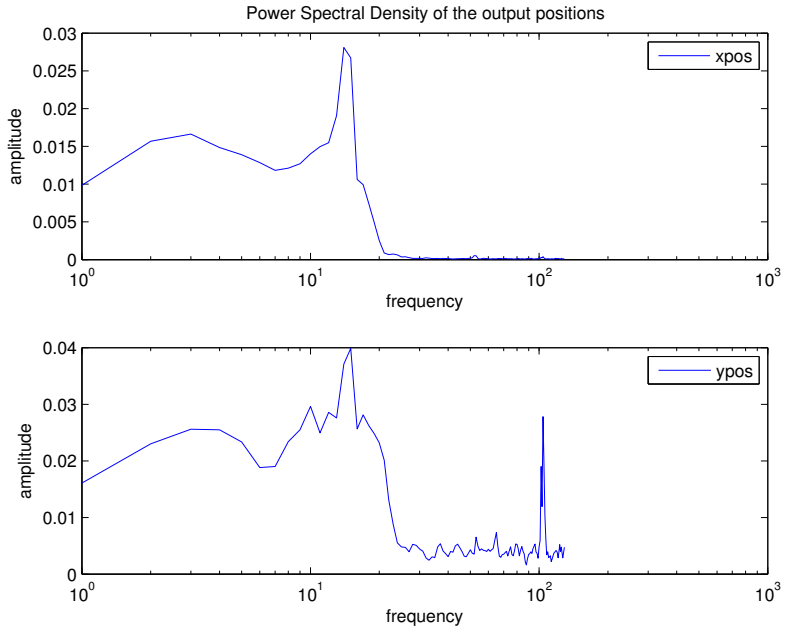

Figure 5.16 Experiment *Pulse2*: spectra of $x-$ and $y-$ TCP position measured by the camera. Two frequency intervals can be noticed: at low-frequency interval centered approximately in 15 Hz, and a high-frequency interval, centered close to 103 Hz

the resonance modes of the afore-mentioned type of network. For this reason, the fourth-order model of data *Pulse2* with parameters values given by (B.30), (B.31), (B.32) presented in section 5.1 was considered. Computing the transformation of the afore-mentioned model to linear mechanical model as suggested in A.5 and considering the stiffness matrix derived in the previous section, it was possible obtaining a numerical expression for the inertia matrix and the damping matrix, specifically

$$M = \begin{bmatrix} 0.0278 & -0.0331 \\ -0.1759 & 0.3086 \end{bmatrix}, \quad \mathcal{D} = \begin{bmatrix} 3.7432 & -4.7826 \\ -9.7408 & -3.4976 \end{bmatrix} \tag{5.3}$$

Surprisingly, whereas M is positive definite with positive trace thus perfectly reliable from a physical point of view, $\mathcal{D}$ despite its positive trace has negative determinant and it seems difficult to find a mechanical meaning. Further, the generalized eigenvalue problem

$$\det(M\lambda + \mathcal{K}) = 0 \tag{5.4}$$

determined the resonance modes $\pm\sqrt{\lambda}$ at the frequencies 14 Hz and 103 Hz matching those found with spectral analysis on the model (B.30), (B.31), (B.32) (see Fig. 5.16).

6. Discussion of the results

In this section all the results achieved throughout all the thesis will be commented.

6.1 Consistency condition and ill-conditioning of the estimates

As mentioned in Section 4.5 the dynamics of the input signal is crucial for the outcome of an identification experiment. As shown so far, the experimental data treated in this work fulfill the necessary requirements of excitation. Moreover, the consistency conditions claimed in [Jansson and Wahlberg, 1997] and in [Chiuso and Picci, 2004b] are satisfied: the principal angles between the rowspaces of the state and future input are not zero. Thus the computation of the parameters (A, C) of the regression can be considered not ill-conditioned.

6.2 On the models obtained

Although all the routines applied for the identification have provided an estimation for the D matrix, the physical nature of the problem has required to force the afore-mentioned parameter to a null matrix.

From what has been pointed out in Sections 5.1 and 5.2 it seems reasonable to assume that the system exhibits resonances at 15 Hz. Since many tools can be mounted on the TCP in order to accomplish tasks of various nature it is worthwhile knowing where resonant poles can be located, when designing a controller. Moreover, the disturbance component guessed from the early model given in Fig. 4.8 turned out to be not only by chance. Looking into the controllability gramians of the estimated models it emerges that the noise does not affect the measurements only, but instead it affects the structure.

6.3 Software tool in subspace-based identification

Throughout the work three tools were used: N4SID, SMI toolbox and ID-toolbox. Whereas the first suffers from instability in case of poor experimental conditions, the second seems to be robust, providing good models. Last, the toolbox based on the orthogonal decomposition approach turned out to give interesting insights in the disturbance components. Clearly, one may conclude that the quest for an efficient subspace procedure (if any) is still open.

7. Conclusions

This thesis has dealt with the identification of a Parallel Kinematic Manipulator dynamics. The identification procedures applied have treated the robot system as a MIMO system, being an important contribution since coupling effects in a modern industrial robot cannot be ignored.

The resonances in the structure appeared at a frequency of 15 Hz and can be explained as beam oscillations during the motion as well as the component found at 100 Hz.

As far as the first attempt on the physical parametrization is concerned, the results obtained are satisfactory, even though a deeper analysis can be carried out taking into account the specification of the material and the geometric properties of the links. Moreover, new experiments performed with a load on the end-effector may clear up what was discussed so far.

8. Bibliography

ABB Automation Technologies AB, Västerås, SE (2004): *Product Specification-Controller IRC5 with FlexPendant*, revision 2 edition.

Brogårdh, T. (2002): "PKM Research-Important Issues, as seen from a Product Development Perspective at ABB Robotics." In *Proc. of the Workshop on Fundamental Issues and Future Research Directions for Parallel Mechanisms and Manipulators*.

Brogårdh, T., S. Hanssen, and G. Hovland (2005): "Application-oriented development of parallel kinematic manipulators with large workspace." In *Proc. 2nd International Colloquium of the Collaborative Research Center 562: Robotic Systems for Handling and Assembly, Braunschweig, Germany, May 2005, pp.153-170.*, pp. 153–170. Braunschweig, Germany.

Chiuso, A. and G. Picci (2004a): "Asymptotic variance of subspace methods by data orthogonalization and model decoupling: a comparative analysis." *Automatica*, **No 40**, pp. 1705–1717.

Chiuso, A. and G. Picci (2004b): "On the ill-conditioning of subspace identification with input." *Automatica*, **No 40**, pp. 575–589.

Chiuso, A. and G. Picci (2004c): "Subspace identification by data orthogonalization and model decoupling." *Automatica*, **No 40**, pp. 1689–1703.

Dressler, I., A. Robertsson, and R. Johansson (2007): "Accuracy of kinematic and dynamic models of a gantry-tau parallel kinematic robot." In *International Conference on Robotics and Automation (ICRA '07)*. Rome.

Haverkamp, B. (2000): *Subspace Method Identification, theory and practice*. PhD thesis, TU Delft, Delft, The Netherlands.

Haverkamp, B. and M. Verhaegen (1997): *SMI Toolbox: State space Model Identification software for multivariable dynamical systems*. TU Delft, Delft, The Netherlands, 1.0 edition.

Intel Corporation (2008): "Open source computer vision library." `http://www.intel.com/technology/computing/opencv/`, visited 2008-02-29.

Jansson, M. and B. Wahlberg (1997): "Counterexample to general consistency of subspace system identification methods." In *SYSID97*, pp. 1667–1682. Fukuoka, Japan.

Jansson, M. and B. Wahlberg (1998): "On consistency of subspace methods for system identification." *Automatica*, **34:12**, pp. 1507–1519.

Johannesson, L., V. Berbyuk, and T. Brogårdh (2005): "Gantry-tau-a new three degrees of freedom parallel kinematic robot." In *Proc. 4th Chemnitz Parallel Kinematics Seminar*, pp. 731–734. Germany.

Johansson, R. (1993): *System Modeling and Identification*. Prentice Hall, Englewood Cliffs, NJ.

Katayama, T. and G. Picci (1999): "Realization of stochastic systems with exogenous inputs and subspace identification methods." *Automatica*, **No 35**, pp. 1635–1652.

Lyzell, C. (2007): "Modeling and identification of the gantry-tau parallel kinematic machine.". Master's thesis, Linköpings Universitet—Tekniska Högskolan.

Merlet, J.-P. (2000): *Parallel Robots*. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Murray, M., G. Hovland, and T. Brogårdh (2006): "Collision-free workspace design of the 5-axis gantry-tau parallel kinematic machine." In *Proc. of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Beijing, China.

Olsson, T. (2007): *High-Speed Vision and Force Feedback for Motion-Controlled Industrial Manipulators*. PhD thesis, Lunds Tekniska Högskola, Lunds Universitet.

Picci, G. (2005): "Geometric methods for state space identification.".

Picci, G. (2006): "Metodi statistici per l'identificazione di modelli lineari.".

Picci, G. and T. Katayama (1996): "Stochastic realization with exogenous inputs and 'subspace-methods' identification." *Signal Processing*, **No 52**, pp. 145–160.

Söderström, T. and P. Stoica (1989): *System Identification*. Prentice Hall International (UK) Ltd.

van Overschee, P. and B. De Moor (1994): "N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems." *Automatica*, **30**, pp. 75–93.

Wernholt, E. (2007): *Multivariable Frequency-Domain Identification of Industrial Robots*. PhD thesis, Linköpings Universitet—Tekniska Högskolan.

A. Implementation Details

A.1 On the predictor used in the corner tracker application

Considering constant velocity, let the system be described by the following continous-time state-space model:

$$\begin{aligned} \dot{x}(t) &= \underbrace{\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}}_{G} x(t) \\ y(t) &= \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{H} x(t) \end{aligned} \tag{A.1}$$

with state vector $x(t) = \begin{bmatrix} x^1(t) \\ x^2(t) \end{bmatrix}$, where $x^1(t)$ is the position and $x^2(t)$ is the velocity.
The discretization with $T_s = 4$ ms gives:

$$\begin{aligned} \begin{bmatrix} x^1_{k+1} \\ x^2_{k+1} \end{bmatrix} &= \underbrace{\begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix}}_{\Phi} \begin{bmatrix} x^1_k \\ x^2_k \end{bmatrix} \\ y_k &= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x^1_k \\ x^2_k \end{bmatrix} \end{aligned} \tag{A.2}$$

A one-step-ahead predictor for the velocity can be formulated as:

$$\begin{aligned} \hat{x}_{k+1|k} &= \Phi \hat{x}_{k|k-1} + K(y_k - \hat{y}_{k|k-1}) \\ \hat{y}_{k+1|k} &= H \hat{x}_{k+1|k} \end{aligned} \tag{A.3}$$

Computing the estimation error $\tilde{x}$ one obtaines:

$$\tilde{x}_{k+1} = x_{k+1} - \hat{x}_{k+1} = (\Phi - HK)\tilde{x}_k \tag{A.4}$$

Thus, $\tilde{x}_{k+1} \to 0$ if $(\Phi - HK)$ is Hurwitz-polynomial. In order to fulfill the above requirement the unknown parameter K have been determined according to pole-placement techniques. Of course, the choice of pole assignment is not unique. In this project poles in 0.6 and 0.8 turned out leading to a sufficiently well performance of the predictor. The gain matrix K further obtained is:

$$K = \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} = \begin{bmatrix} 0.6 \\ 20 \end{bmatrix} \tag{A.5}$$

Rewriting the equation:

$$\begin{aligned} \hat{x}_{k+1|k} &= \begin{bmatrix} \hat{x}^1_{k+1|k} \\ \hat{x}^2_{k+1|k} \end{bmatrix} = \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{x}^1_{k|k-1} \\ \hat{x}^2_{k|k-1} \end{bmatrix} + \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} (y_k - \hat{y}_{k|k-1}) \\ \hat{y}_{k+1|k} &= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \hat{x}^1_{k+1|k} \\ \hat{x}^2_{k+1|k} \end{bmatrix} \end{aligned} \tag{A.6}$$

Thus, explicitly, in a more handable way for code implementation in C++:

$$\begin{aligned}\hat{x}^1_{k+1|k} &= (1-k_1)\hat{x}^1_{k|k-1} + k_1 x^1_k + T_s \hat{x}^2_{k|k-1} \\ \hat{x}^2_{k+1|k} &= \hat{x}^2_{k|k-1} + k_2 x^1_k - k_2 \hat{x}^1_{k|k-1}\end{aligned} \tag{A.7}$$

Note that $\hat{x}^1_{k+1|k}$ represents the predicted position, whereas $\hat{x}^2_{k+1|k}$ represents the predicted velocity.

A.2 On the computation of the residuals

Consider a discrete-time-invariant system $\Sigma(A,B,C,D)$ with the state-space equation

$$\begin{cases}\mathbf{x}(k+1) = A\mathbf{x}(k) + B\mathbf{u}(k) + \mathbf{w}(t) \\ \mathbf{y}(k) = C\mathbf{x}(k) + D\mathbf{u}(k) + \mathbf{v}(k),\end{cases} \tag{A.8}$$

with input $\mathbf{u}(k) \in \mathbb{R}^m$, output $\mathbf{y}(k) \in \mathbb{R}^p$, state vector $\mathbf{x}(k) \in \mathbb{R}^n$ and noise sequences $\mathbf{w}(k) \in \mathbb{R}^n$, $\mathbf{v}(k) \in \mathbb{R}^p$. This can be manipulated (see the literature for details) in order to get an innovation representation (A.9) with the same statistics but with more interesting noise properties.

$$\begin{bmatrix}\mathbf{x}(k+1) \\ \mathbf{y}(k)\end{bmatrix} = \begin{bmatrix}A & B \\ C & D\end{bmatrix}\begin{bmatrix}\mathbf{x}(k) \\ \mathbf{u}(k)\end{bmatrix} + \begin{bmatrix}K \\ I\end{bmatrix}\mathbf{e}(k) \tag{A.9}$$

Indeed, the innovation process $\{\mathbf{e}(k)\}$, which is a sequence of zero-mean independent identically distributed (i.i.d) stochastic variables, may be reconstructed by the output sequence $\{\mathbf{y}(k)\}$. Moreover, a residual sequence computed as the estimate $\{\hat{\mathbf{e}}(k)\}$ of the innovation is obtained by means of the following:

$$\begin{bmatrix}\hat{\mathbf{x}}(k+1) \\ \hat{\mathbf{e}}(k)\end{bmatrix} = \begin{bmatrix}A-KC & B-KD \\ -C & -D\end{bmatrix}\begin{bmatrix}\hat{\mathbf{x}}(k) \\ \mathbf{u}(k)\end{bmatrix} + \begin{bmatrix}K \\ I\end{bmatrix}\mathbf{y}(k) \tag{A.10}$$

A.3 On the decoupled state-space model

Let's consider the deterministic component's submodel:

$$\begin{cases}\mathbf{x}_d(k+1) = A_d\mathbf{x}_d(k) + B_d\mathbf{u}(k) \\ \mathbf{y}_d(k) = C_d\mathbf{x}_d(t)\end{cases} \tag{A.11}$$

corresponding to the input-output transfer function:

$$Y_d(z) = \underbrace{C_d(zI - A_d)^{-1}B}_{G(z)}\, U(z) \tag{A.12}$$

Similarly, let the stochastic submodel be:

$$\begin{cases}\mathbf{x}_s(k+1) = A_s\mathbf{x}_s(k) + K_s\mathbf{w}(k) \\ \mathbf{y}_s(k) = C_s\mathbf{x}_s(k) + \mathbf{w}(k)\end{cases} \tag{A.13}$$

with input-output transfer function:

$$Y_s(z) = \underbrace{[C_s(zI - A_s)^{-1}K_s + I]}_{H(z)} W(z) \tag{A.14}$$

Thus, the following state-space realization is achieved:

$$\begin{cases} \hat{\mathbf{x}}_d(k+1) = A_d\hat{\mathbf{x}}_d(k) + B_d\mathbf{u}(k) \\ \hat{\mathbf{y}}_d(k) = C_d\hat{\mathbf{x}}_d(k) \end{cases} \tag{A.15}$$

$$\begin{cases} \hat{\mathbf{x}}_s(k+1) = (A_s - K_sC_s)\hat{\mathbf{x}}_s(k) + K_s(\mathbf{y}(k) - \hat{\mathbf{y}}(k)) \\ \hat{\mathbf{w}}(k) = \mathbf{y}(k) - \hat{\mathbf{y}}_d(k) - C_s\hat{\mathbf{x}}_s(k) \end{cases} \tag{A.16}$$

For

$$\hat{\mathbf{x}}(k) = \begin{bmatrix} \hat{\mathbf{x}}_d(k) \\ \hat{\mathbf{x}}_s(k) \end{bmatrix}, \quad \hat{\mathbf{y}}(k) = \begin{bmatrix} \hat{\mathbf{y}}_d(k) \\ \mathbf{y}_s(k) \end{bmatrix}$$

$$\begin{cases} \hat{\mathbf{x}}(k+1) = \begin{bmatrix} A_d & 0 \\ 0 & A_s \end{bmatrix} \hat{\mathbf{x}}(k) + \begin{bmatrix} B \\ 0 \end{bmatrix} \mathbf{u}(k) + \begin{bmatrix} 0 \\ K \end{bmatrix} (\mathbf{y}(k) - \hat{\mathbf{y}}(k)) \\ \hat{\mathbf{y}}(k) = \begin{bmatrix} C_d & 0 \\ 0 & C_s \end{bmatrix} \hat{\mathbf{x}}(k) \\ \hat{\mathbf{w}}(k) = \mathbf{y}(k) - \hat{\mathbf{y}}(k) \end{cases} \tag{A.17}$$

A.4 On the linear regression problem for stiffness estimation

Let's recall Hooke's law:

$$\mathbf{F} = \mathcal{K}\Delta\mathbf{q}, \qquad \Delta\mathbf{q} = [\,\Delta\mathbf{x} \quad \Delta\mathbf{y}\,]' \tag{A.18}$$

Thus, given

$$\begin{aligned} \mathbf{F} &= \mathcal{K}(\mathbf{q} - \mathbf{q}_0) + \mathbf{F}_0 \\ &= \mathcal{K}\mathbf{q} + (\mathbf{F}_0 - \mathcal{K}\mathbf{q}_0) \end{aligned} \tag{A.19}$$

multiplying from the right by $\mathbf{q}^T$ one obtaines:

$$\mathbf{F}\mathbf{q}^T = \mathcal{K}\mathbf{q}\mathbf{q}^T + (\mathbf{F}_0 - \mathcal{K}\mathbf{q}_0)\mathbf{q}^T \tag{A.20}$$

Now, considering a sequence of N samples:

$$\begin{bmatrix} \mathbf{f}_{x_1} \cdots \mathbf{f}_{x_N} \\ \mathbf{f}_{y_1} \cdots \mathbf{f}_{y_N} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 & \mathbf{y}_1 \\ \vdots & \vdots \\ \mathbf{x}_N & \mathbf{y}_N \end{bmatrix} = \mathcal{K} \begin{bmatrix} \mathbf{x}_1 \cdots \mathbf{x}_N \\ \mathbf{y}_1 \cdots \mathbf{y}_N \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 & \mathbf{y}_1 \\ \vdots & \vdots \\ \mathbf{x}_N & \mathbf{y}_N \end{bmatrix} \tag{A.21}$$

which clearly is the formulation of a linear regression problem of the type

$$\mathcal{Y} = \Phi\theta, \quad \theta = \mathcal{K}^T \tag{A.22}$$

$$\hat{\theta} = (\Phi^T\Phi)^{-1}\Phi^T\mathcal{Y} \tag{A.23}$$

$$\hat{\theta}^T = \mathcal{Y}^T\Phi(\Phi^T\Phi)^{-T} \tag{A.24}$$

one can estimate the matrix stiffness as follows:

$$\hat{\mathcal{K}} = \mathbf{F}\mathbf{q}^T(\mathbf{q}\mathbf{q}^T)^{-1} \tag{A.25}$$

A.5 On the transformation to linear mechanical model

Let's assume to prefer the estimated state-space model given in the following equation

$$\begin{cases} x(t+1) = Ax(t) + Bu(t) \\ y(t) = Cx(t), \quad t \in \mathbb{R} \end{cases} \tag{A.26}$$

on the format:

$$\begin{cases} x(t+1) = \begin{bmatrix} 0 & I \\ -M^{-1}\mathcal{K} & -M^{-1}\mathcal{D} \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ M^{-1}\mathcal{B}_0 \end{bmatrix} u(t) \\ y(t) = Cx(t) \end{cases} \tag{A.27}$$

where M, $\mathcal{K}$, $\mathcal{D}$ are matrices representing the inertia of the system, the stiffness and the damping respectively, the state $x \in \mathbb{R}^{2n}$ and the input $u \in \mathbb{R}^n$.
The corresponding transfer function relationship between $U(s)$ and $X(s)$ is:

$$X(s) = (Ms^2 + \mathcal{D}s + \mathcal{K})^{-1}\mathcal{B}_0 U(s) \tag{A.28}$$

which provides important information on vibration and resonances of the mechanical system as shown in [Johansson, 1993]). In the light of the above consideration, it's necessary to find a similarity transformation matrix

$$T = \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} \quad \text{for} \quad A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad B = \begin{bmatrix} B_1 & B_2 \end{bmatrix} \tag{A.29}$$

such that

$$TAT^{-1} = \begin{bmatrix} 0 & I \\ -M^{-1}\mathcal{K} & -M^{-1}\mathcal{D} \end{bmatrix}, \quad TB = \begin{bmatrix} 0 \\ M^{-1}\mathcal{B}_0 \end{bmatrix} \tag{A.30}$$

Collecting the equation as follows

$$\begin{bmatrix} T_{11} & T_{12} & T_{21} & T_{22} \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} & B_1 \\ A_{21} & A_{22} & B_2 \\ -I & 0 & 0 \\ 0 & -I & 0 \end{bmatrix} = 0 \tag{A.31}$$

and computing the singular value decomposition

$$\begin{bmatrix} U & U_\perp \end{bmatrix} \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} V^T = \begin{bmatrix} A_{11} & A_{12} & B_1 \\ A_{21} & A_{22} & B_2 \\ -I & 0 & 0 \\ 0 & -I & 0 \end{bmatrix} \in \mathbb{R}^{4n \times 3n} \tag{A.32}$$

it is possible to choose

$$\begin{bmatrix} T_{11} & T_{12} & T_{21} & T_{22} \end{bmatrix} = \tau U_\perp^T \in \mathbb{R}^{n \times 4n} \tag{A.33}$$

with τ any arbitrary invertible matrix.

B. Models

B.1 On the arm-side

♦ experiment *Arm1*:
State-space system on the form:

$$\begin{cases} \mathbf{x}(k+1) = A\mathbf{x}(k) + B\mathbf{u}(k) + \mathbf{w}(t) \\ \mathbf{y}(k) = C\mathbf{x}(k) + D\mathbf{u}(k) + \mathbf{v}(k), \end{cases} \tag{B.1}$$

with $\mathbf{u}(k) \in \mathbb{R}$, $\mathbf{y}(k) \in \mathbb{R}^2$, state $\mathbf{x}(k) \in \mathbb{R}^4$ and noise sequences $\mathbf{w}(k) \in \mathbb{R}^4$, $\mathbf{v}(k) \in \mathbb{R}$,

$$A = \begin{bmatrix} 0.9867 & -0.1178 & 0.0087 & 0.0092 \\ -0.0030 & 0.9718 & 0.0178 & -0.0346 \\ 0.0004 & 0.0031 & 0.9747 & -0.1042 \\ -0.0002 & -0.0020 & 0.0097 & 0.9860 \end{bmatrix} \tag{B.2}$$

$$B = \begin{bmatrix} 0.1796 \\ 0.0445 \\ -0.0130 \\ 0.0046 \end{bmatrix} \tag{B.3}$$

$$C = \begin{bmatrix} 0.1645 & 0.2616 & 0.1254 & 0.2054 \\ 0.1498 & 0.2258 & -0.2088 & -0.1990 \end{bmatrix} \tag{B.4}$$

$$D = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{B.5}$$

and Kalman gain:

$$K = \begin{bmatrix} 1.2409 & 8.5429 \\ 0.3824 & -4.8429 \\ 1.2609 & -1.4762 \\ 0.1044 & 0.0969 \end{bmatrix} \tag{B.6}$$

State-space system representation as in (4.3), with $\mathbf{u}(k) \in \mathbb{R}$, $\mathbf{y}(k) \in \mathbb{R}^2$, state $\mathbf{x}_d(k) \in \mathbb{R}^2$, $\mathbf{x}_s(k) \in \mathbb{R}^7$ and noise sequences $\mathbf{e}(k) \in \mathbb{R}^7$,

$$A_d = \begin{bmatrix} 0.9557 & 0.0124 \\ -0.0077 & 0.8819 \end{bmatrix} \tag{B.7}$$

$$B_d = \begin{bmatrix} -0.3741 \\ -3.4790 \end{bmatrix} \tag{B.8}$$

$$C_d = \begin{bmatrix} -0.0738 & 0.0075 \\ -0.0671 & 0.0038 \end{bmatrix} \tag{B.9}$$

$$D_d = \begin{bmatrix} 0.0026 \\ 0.0052 \end{bmatrix} \tag{B.10}$$

$$A_s = \begin{bmatrix} 0.9995 & -0.0027 & -0.0028 & -0.0084 & 0.0069 & 0.0016 & -0.0058 \\ 0.0028 & 0.9801 & -0.0537 & -0.0002 & -0.0297 & 0.0068 & 0.0122 \\ -0.0021 & 0.0520 & -0.6459 & 0.5496 & 0.2880 & 0.0369 & -0.1889 \\ 0.0103 & -0.0321 & -0.4850 & -0.1910 & -0.5018 & -0.3383 & 0.4228 \\ -0.0053 & 0.0027 & -0.3040 & -0.3997 & -0.4807 & 0.3669 & -0.4941 \\ -0.0002 & -0.0071 & 0.2543 & 0.5787 & -0.5571 & 0.1729 & -0.1749 \\ 0.0091 & -0.0092 & -0.0898 & -0.0496 & 0.0866 & 0.8172 & 0.5738 \end{bmatrix} \tag{B.11}$$

$$K_s = \begin{bmatrix} -1.9005 & 0.1211 \\ 256.8094 & -16.3693 \\ -455.4175 & 29.0289 \\ -67.4569 & 4.2998 \\ -94.8816 & 6.0479 \\ 258.5190 & -16.4783 \\ -88.8556 & 5.6638 \end{bmatrix} \tag{B.12}$$

$$C_s = \begin{bmatrix} 0.0076 & 0.0020 & 0.0003 & -0.0001 & -0.0001 & -0.0000 & 0.0001 \\ 0.0071 & -0.0001 & -0.0000 & -0.0000 & 0.0000 & 0.0000 & -0.0000 \end{bmatrix} \tag{B.13}$$

B.2 On the TCP

♦ experiment *Arm2*:

State-space system obtained with PO-MOESP on the form given in B.1, with $\mathbf{u}(k) \in \mathbb{R}$, $\mathbf{y}(k) \in \mathbb{R}^3$, state $\mathbf{x}(k) \in \mathbb{R}^5$ and noise sequences $\mathbf{w}(k) \in \mathbb{R}^5$, $\mathbf{v}(k) \in \mathbb{R}^3$,

$$A = \begin{bmatrix} 0.9862 & -0.0313 & -0.1235 & 0.0746 & -0.0860 \\ -0.0134 & 0.9796 & 0.0599 & -0.0698 & -0.1379 \\ 0.0021 & -0.0014 & 0.9931 & 0.0172 & 0.0450 \\ -0.0014 & -0.0005 & -0.0065 & 0.9937 & 0.0056 \\ -0.0006 & -0.0005 & -0.0036 & -0.0069 & 0.9934 \end{bmatrix} \tag{B.14}$$

$$B = \begin{bmatrix} 0.1480 \\ 0.1527 \\ -0.0085 \\ 0.0053 \\ 0.0041 \end{bmatrix} \tag{B.15}$$

$$C = \begin{bmatrix} 0.0211 & -0.2067 & 0.2440 & -0.0966 & -0.2716 \\ 0.2348 & 0.0598 & 0.1883 & -0.2149 & 0.1800 \\ 0.0972 & 0.1719 & 0.1172 & 0.1639 & 0.1452 \end{bmatrix} \tag{B.16}$$

$$D = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{B.17}$$

and Kalman gain:

$$K = \begin{bmatrix} 2.9959 & 4.0019 & -0.5202 \\ -4.3098 & -1.1080 & 1.8186 \\ -0.0392 & -1.6194 & 1.0140 \\ 1.9259 & 0.0059 & 0.3605 \\ 0.5520 & -0.2397 & -0.4375 \end{bmatrix} \tag{B.18}$$

♦ experiment *Pulse1*:

State-space system achieved with the Ho-Kalman realization technique on the form given in B.1, with $\mathbf{u}(k) \in \mathbb{R}$, $\mathbf{y}(k) \in \mathbb{R}$, state $\mathbf{x}(k) \in \mathbb{R}^6$ and noise sequences $\mathbf{w}(k) \in \mathbb{R}^6$, $\mathbf{v}(k) \in \mathbb{R}$,

$$A = \begin{bmatrix} 0.9396 & -0.1076 & -0.1926 & 0.2289 & 0.1112 & 0.0143 \\ 0.0843 & 0.9275 & -0.3058 & -0.1376 & -0.0802 & -0.0012 \\ 0.1851 & 0.2925 & 0.9122 & 0.1300 & 0.0510 & -0.0126 \\ -0.2200 & 0.1624 & -0.0791 & 0.8958 & -0.0134 & 0.0404 \\ -0.1099 & 0.0774 & -0.0342 & -0.0570 & 0.9740 & 0.0615 \\ -0.2038 & -0.2027 & -0.1052 & 0.4774 & -0.2319 & 0.5055 \end{bmatrix} \tag{B.19}$$

$$B = \begin{bmatrix} 0.0795 \\ 0.2898 \\ 0.5659 \\ -0.7335 \\ -0.1450 \\ 2.8114 \end{bmatrix} \tag{B.20}$$

$$C = \begin{bmatrix} 0.0982 & 0.0359 & 0.5388 & -0.6955 & -0.3720 & 0.4386 \end{bmatrix} \tag{B.21}$$

$$D = 0 \tag{B.22}$$

♦ experiment *Pulse2*:

State-space system representation as in (4.3), with $\mathbf{u}(k) \in \mathbb{R}^2$, $\mathbf{y}(k) \in \mathbb{R}^6$, state $\mathbf{x}_d(k) \in \mathbb{R}^2$, $\mathbf{x}_s(k) \in \mathbb{R}^4$ and noise sequences $\mathbf{e}(k) \in \mathbb{R}^4$,

$$A_d = \begin{bmatrix} 0.8824 & -0.3342 \\ 0.3314 & 0.8567 \end{bmatrix} \tag{B.23}$$

$$B_d = \begin{bmatrix} -0.0079 & -0.0104 \\ -0.0051 & -0.0133 \end{bmatrix} \tag{B.24}$$

$$C_d = \begin{bmatrix} -0.0694 & -0.0493 \\ 0.1128 & 0.0873 \end{bmatrix} \tag{B.25}$$

$$D_d = \begin{bmatrix} 0.0001 & 0.0046 \\ -0.0047 & -0.0041 \end{bmatrix} \tag{B.26}$$

$$A_s = \begin{bmatrix} 0.8913 & -0.3321 & 0.0214 & -0.0100 \\ 0.3304 & 0.9344 & 0.0745 & 0.0196 \\ -0.0034 & 0.0594 & -0.8177 & -0.5704 \\ -0.0108 & -0.0174 & 0.5658 & -0.8004 \end{bmatrix} \tag{B.27}$$

$$K_s = \begin{bmatrix} 18.9614 & -1.0025 \\ -3.6725 & -0.2541 \\ 4.5104 & 0.4918 \\ -0.6334 & 1.8764 \end{bmatrix} \tag{B.28}$$

$$C_s = \begin{bmatrix} 0.0133 & -0.0014 & -0.0016 & -0.0019 \\ -0.0134 & -0.0007 & 0.0094 & -0.0200 \end{bmatrix} \tag{B.29}$$

Using subspace id the state-space realization is given as (4.3), with $\mathbf{u}(k) \in \mathbb{R}^2$, $\mathbf{y}(k) \in \mathbb{R}^2$, state $\mathbf{x}(k) \in \mathbb{R}^4$ and noise $\mathbf{e}(k) \in \mathbb{R}^2$:

$$A = \begin{bmatrix} 0.8901 & -0.3311 & 0.0492 & -0.0103 \\ 0.3194 & 0.8757 & 0.1085 & -0.0304 \\ 0.0098 & 0.0224 & -0.8014 & -0.6038 \\ -0.0186 & -0.0348 & 0.5335 & -0.7991 \end{bmatrix} \tag{B.30}$$

$$B = \begin{bmatrix} 0.0030 & -0.0033 \\ 0.0081 & -0.0020 \\ -0.0479 & -0.0480 \\ 0.2838 & 0.2258 \end{bmatrix} \tag{B.31}$$

$$C = \begin{bmatrix} -0.0748 & -0.0537 & -0.0038 & 0.0109 \\ 0.1193 & 0.0776 & -0.0614 & 0.0408 \end{bmatrix} \tag{B.32}$$

$$D = \begin{bmatrix} -0.0015 & 0.0032 \\ -0.0184 & -0.0149 \end{bmatrix} \tag{B.33}$$

$$K = \begin{bmatrix} -2.0626 & 0.2273 \\ -1.5904 & 0.1394 \\ -2.3304 & 0.2305 \\ -0.3247 & -0.6778 \end{bmatrix} \tag{B.34}$$

Using PO-MOESP the state-space realization is given as (B.1) with $\mathbf{u}(k) \in \mathbb{R}^2$, $\mathbf{y}(k) \in \mathbb{R}^2$, state $\mathbf{x}(k) \in \mathbb{R}^4$ and noises $\mathbf{w}(k) \in \mathbb{R}^2$, $\mathbf{v}(k) \in \mathbb{R}^2$:

$$A = \begin{bmatrix} 0.8893 & -0.3881 & 0.0508 & -0.0416 \\ 0.2684 & 0.8784 & 0.1271 & -0.1199 \\ -0.0005 & 0.0018 & -0.8032 & -0.5896 \\ -0.0180 & -0.0434 & 0.5510 & -0.7925 \end{bmatrix} \tag{B.35}$$

$$B = \begin{bmatrix} -0.0033 & -0.0052 \\ -0.0046 & -0.0070 \\ 0.0174 & 0.1724 \\ -0.0282 & 0.0635 \end{bmatrix} \tag{B.36}$$

$$C = \begin{bmatrix} -0.1965 & -0.1682 & 0.0036 & 0.0498 \\ 0.3129 & 0.2485 & -0.2151 & 0.2007 \end{bmatrix} \tag{B.37}$$

$$D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{B.38}$$

and Kalman gain

$$K = \begin{bmatrix} -0.7170 & 0.0872 \\ -0.5728 & 0.0426 \\ -0.5323 & 0.0432 \\ -0.0812 & -0.1566 \end{bmatrix} \tag{B.39}$$

C. Plot of input-output signals used for identification

On the Arm side

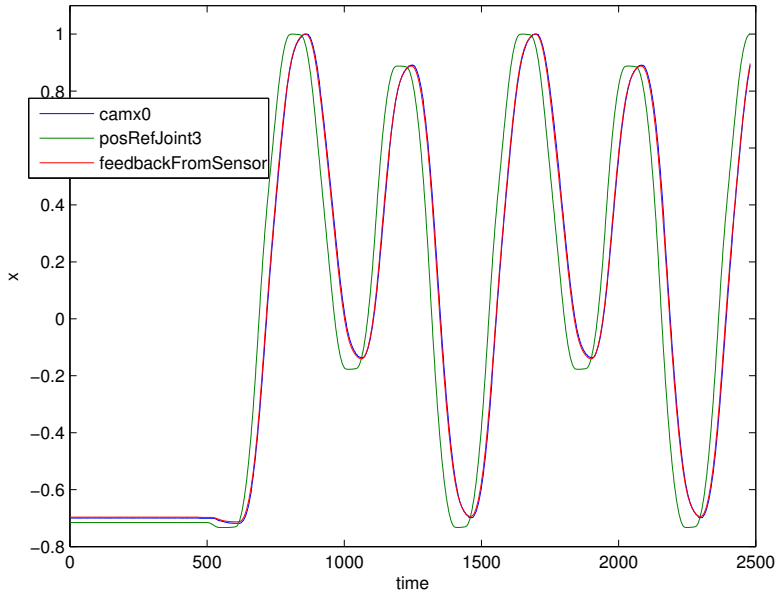

Figure C.1 Experiment *Arm1*. Input signals: cart position reference to controller. Output signals: cart position measured by sensor and by the camera

C.1 On the TCP

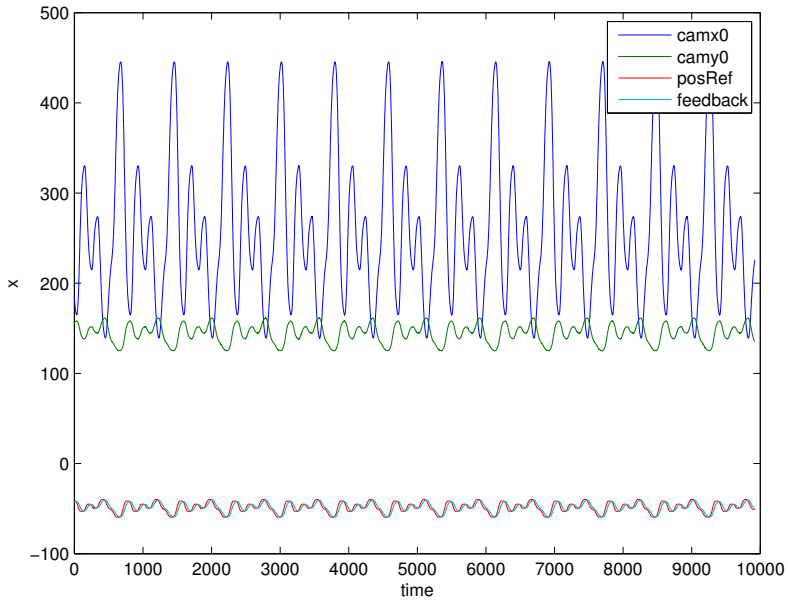

Figure C.2 Experiment *Arm2*. Input signals:cart position reference to controller. Output signals: cart position measured by the resolover, TCP x- and y-coordinate in TCP coordinate frame

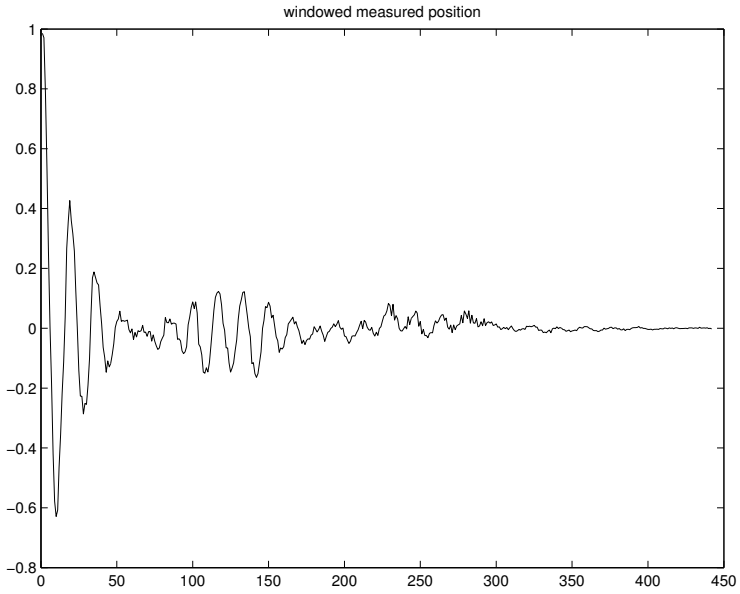

Figure C.3 Experiment *Pulse1*. Output signal: TCP position from camera

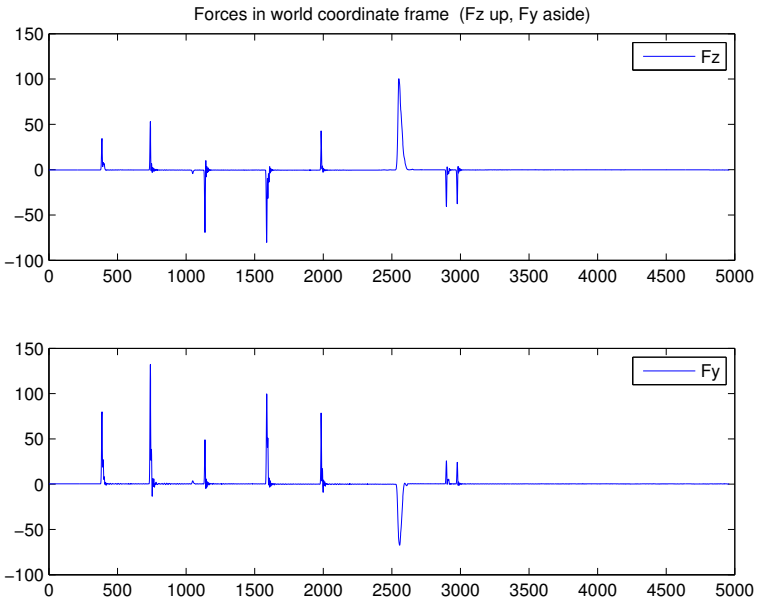

Figure C.4 Experiment *Pulse2*. Input signals: forces applied to the end-effector in world coordinate frame

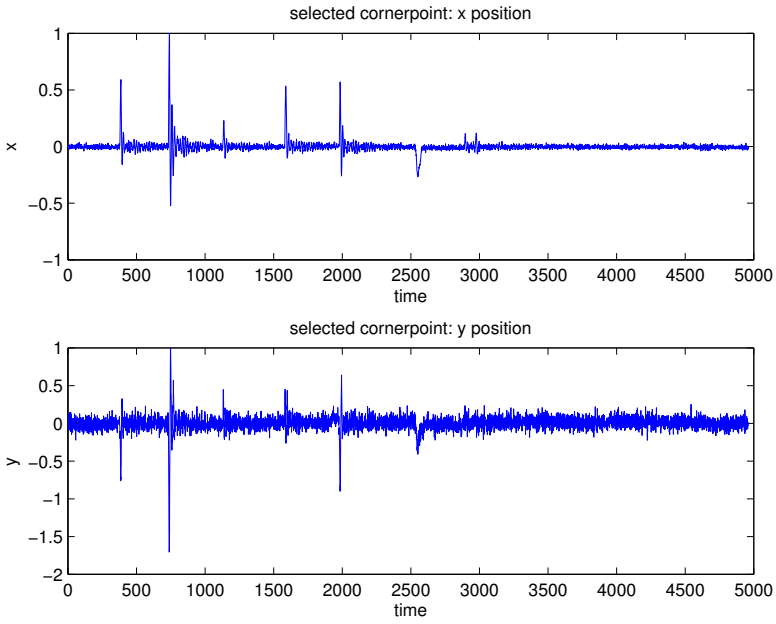

Figure C.5 Experiment *Pulse2*. Output signals: TCP x- and y-position from camera in world coordinate frame

C.2 C++ code for one-step ahead predictor

```
/*-------------------------------------------------------------
 *
 * discrete-time Kalman filter for the prediction
 * of the next cornerpoint's position in the form:
 *
 * X_hat(k+1) = [1 Ts;0 1]*X_hat(k)+[K11;K22]*(Y(k)-Y_hat(k))
 * Y_hat(k+1) = [1 0]*Y_hat(k+1)
 *
 *
 * The motion has been decoupled along the x-axis and the y-axis,so:
 *
 * X_hat(k) = [x_hat;vx_hat]  state vector for motion along x
 * Y_hat(k) = [y_hat;vy_hat]  state vector for motion along y
 *
 *-------------------------------------------------------------*/

  CvPoint2D32f pc[nr_feat];
  CvPoint2D32f cornerpoint;
  int nr_trackpoints=0;

  float K11 = 0.6;
  float K22 = 20;
  float Ts = 0.004;//sampling time

  float x_hat_old[nr_feat];// x_hat(k)
  float vx_hat_old[nr_feat];// vx_hat(k)
  float x_hat[nr_feat];// x_hat(k+1)
  float vx_hat[nr_feat];// vx_hat(k+1)

  float y_hat_old[nr_feat];// y_hat(k)
  float vy_hat_old[nr_feat];// vy_hat(k)
  float y_hat[nr_feat];// y_hat(k+1)
  float vy_hat[nr_feat];// vy_hat(k+1)

  camcomm_camx cx;
  camcomm_camy cy;

  bool done=false;
  int ii;

  for (ii=0;ii<nr_feat;ii++) {

   vx_hat_old[ii] = 0.0;
   x_hat_old[ii] = 0.0;

   vy_hat_old[ii] = 0.0;
   y_hat_old[ii] = 0.0;

   pc[ii].x = 0.0;
   pc[ii].y = 0.0;

  }

/*-----------------------------------
```

```
 * find next cornerpoint
 *-----------------------------------*/

  for (ii=0;ii<nr_trackpoints;ii++) {

   x_hat[ii] = (1-K11)*x_hat_old[ii] + K11*(float)pc[ii].x
                  + Ts*vx_hat_old[ii];
   vx_hat[ii] = vx_hat_old[ii] + K22*(float)pc[ii].x
                  - K22*x_hat_old[ii];

   y_hat[ii] = (1-K11)*y_hat_old[ii] + K11*(float)pc[ii].y
                  + Ts*vy_hat_old[ii];
   vy_hat[ii] = vy_hat_old[ii] + K22*(float)pc[ii].y
                  - K22*y_hat_old[ii];

   cornerpoint.x = x_hat[ii];
   cornerpoint.y = y_hat[ii];

   cvFindCornerSubPix(cvimage,&cornerpoint,1,cvSize(7,7),cvSize(-1,-1),
              cvTermCriteria(CV_TERMCRIT_EPS+CV_TERMCRIT_ITER,10,1.0));

/*--------------labcomm--------------*/
  cx.a[ii]=(float)pc[ii].x;
  cy.a[ii]=(float)pc[ii].y;
/*-----------------------------------*/

  data_buffer[2*(nr_feat*log_ptr+ii)]=(double)cornerpoint.x;
  data_buffer[2*(nr_feat*log_ptr+ii)+1]=(double)cornerpoint.y;

  cvCircleAA(cvimage,cvPoint((int)(32.0*cornerpoint.x),...
            (int)(32.0*cornerpoint.y)),32*5,255,5);

  pc[ii].x = cornerpoint.x;// update of the current x  position
  pc[ii].y = cornerpoint.y;// update of the current y position

  x_hat_old[ii] = x_hat[ii];
  vx_hat_old[ii] = vx_hat[ii];

  y_hat_old[ii] = y_hat[ii];
  vy_hat_old[ii] = vy_hat[ii];

  }

  for (ii=nr_trackpoints;ii<nr_feat;ii++) {

   data_buffer[2*(nr_feat*log_ptr+ii)]=0.0;
   data_buffer[2*(nr_feat*log_ptr+ii)+1]=0.0;

  }

  if ((it_counter(int)(FRAME_RATE/display_freq))==0) {
      display_frames(buff_ptr);
  }
  XFlush(display);
  it_counter++;
```

```
  buff_ptr++;
  if (buff_ptr>=n_frame_buffers) buff_ptr=0;

  log_ptr++;
  if (log_ptr>=data_buffer_size) log_ptr=0;

  while(XPending(display)>0){
  XNextEvent(display,&xev);
  switch(xev.type){

   case ConfigureNotify:
   width=xev.xconfigure.width;
   height=xev.xconfigure.height;
   break;
   case ButtonPress:
   if (nr_trackpoints<nr_feat) {

   pc[nr_trackpoints].x = (float)xev.xbutton.x;
   pc[nr_trackpoints].y = (float)xev.xbutton.y;

   x_hat_old[nr_trackpoints] = pc[nr_trackpoints].x;// initialization
   y_hat_old[nr_trackpoints] = pc[nr_trackpoints].y;

   nr_trackpoints++;

   }
   else
   fprintf(stderr,"all features already defined - ignored\n");
   break;
   case KeyPress:
   done=true;
   break;

  }
```

C.3 Matlab code

```
function [A,B,C,D] = HoKalman(h);% where h is the impulse response

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% HoKalman realization algorithm      %
% modified version by Juang and Pappa %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

N = floor(max(size(h))/2)-1;
H0 = zeros(N,N);% Hankel matrix
H1 = zeros(N,N);% Hankel matrix
for ii = 1:N,
  for jj = 1:N,
    H0(ii,jj) = h(ii+jj);%forming the matrix from Markov coefficients
    H1(ii,jj) = h(ii+jj+1);
  end;
end
```

```
r = input('insert system order:')
H0;
H1;
size(H0)
size(H1)

D = h(1);
Ey = [1 zeros(1,N-1)]';
Eu = Ey;
[U,S,V] = svd(H0);%singular value decomposition of H0
figure;
semilogy(diag(S),'o');

% matrix definition

S1 = sqrt(S(1:r,1:r));
Un = U(:,1:r);
Vn = V(:,1:r);
A = inv(S1)*Un'*H1*Vn*inv(S1);
B = S1*Vn'*Eu;
C = Ey'*Un*S1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% estimation with spectral analysis %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

r1 = data(:,1)-mean(data(:,1));% tracked x-pos cornerpoint
rn1 = r1/max(r1);
x_pos = detrend(rn1(1573:2010));

Ts = 0.004; %sampling time
F = 1/Ts; %sampling frequence

% with windowed data:

N = max(size(x_pos));
w = hamming(2*N);%hamming window
w1 = w((N+1):(2*N))';
figure;
plot(w1);
pause;
x_pos_w = x_pos'.*w1;
figure;
plot(x_pos_w);
title('windowed measured position');
%pause;

G = fft(x_pos_w)./0.45;

W = [0:length(G)-1]*(F)*2*pi/(length(G)-1);

figure(1000);
subplot(2,1,1);
semilogx(W, 20*log10(abs(G)));
ylabel('|G(jw)| [dB]','FontSize',12);
```

```
grid on;
subplot(2,1,2);
semilogx(W, phase(conj(G')));
ylabel('phase(G(jw)) [grad]','FontSize',12);
grid on;
xlabel('Frequency [rad/s]','FontSize',12);


% experiment to get transformation matrix

ForceZY_world = [-1 0;0 +1;+1 0;0 -1];
FxFy_jr3 = [-46 31;-26 -35;35 -22;+21 35];

%normalize

FxFy_jr3_n(1,:) = FxFy_jr3(1,:)/norm(FxFy_jr3(1,:));
FxFy_jr3_n(2,:) = FxFy_jr3(2,:)/norm(FxFy_jr3(2,:));
FxFy_jr3_n(3,:) = FxFy_jr3(3,:)/norm(FxFy_jr3(3,:));
FxFy_jr3_n(4,:) = FxFy_jr3(4,:)/norm(FxFy_jr3(4,:));

% Find T such that   ForceZY_world= T*FxFy_jr3_n

T = ForceZY_world' / FxFy_jr3_n';
T = [0.8293 -0.5548 -0.5462 -0.8389];

forceZY_w = T*[Fx_jr3 Fy_jr3]';
```