Modelling and control of a six DOF interactive play robot

Schoenmakers, Ferry B.F.

Award date:
2013

Modelling and control of a six DOF interactive play robot

F.B.F. Schoenmakers

CST 2013.083

Master's thesis

Coach(es):	dr. ir. René van de Molengraft ir. Bart Dirkx ir. Ruud van der Aalst
Supervisor:	prof. dr. ir. Maarten Steinbuch

Eindhoven University of Technology
Department of Mechanical Engineering
Control Systems Technology

Eindhoven, August, 2013

Preface

In front of you right now is my master's graduation thesis. In it I describe my master's graduation project I did on University of Technology Eindhoven. The past year I have been cooperating with the company WittyWorx on their interactive play robot called IXI-Play. I truly hope that the target audience (children of age 1 to 6) will have as much fun with this cute robot as we did in developing it. My aim during the project was the motion control of the platform. This I believe, is one of the most satisfying things in robotics research, to see a machine move and come "alive". Although all its motions are simply a logical result of all the effort you put in it yourself, it remains to be something "magical". And this should always be kept in mind too! Whenever your robot is not behaving as it should, how lifelike it might be, it is still a result of your own actions! The same goes for the Turtles, the soccer robots of Tech United Eindhoven, on which I have been working in the past years too. After struggling through lines of code and maintaining hardware, you end up with a machine that fully autonomously plays a game of soccer! But whenever they don't succeed in winning a match, cursing the robots will most certainly not make them win the next match!

Writing this last piece of my thesis, which will actually be the first page of it, I think this is the right place to give my word of thanks. Without the following people I would not have been able to let all this magic happen and therefore you deserve to be on this first page!

First, I would like to thank the company WittyWorx for giving me the opportunity to work on such a cool project during my graduation and for all the support, feedback and fun I had during my graduation. Also I would like to thank the department of mechanical engineering, my labmates and colleagues for supporting me during graduation and sharing knowledge on the subject of robotics. This especially goes for the team of Tech United Eindhoven. Together with having a lot of fun, this team really taught me a lot! On the subject of mechatronics and programming. But also the skills to always push through, never give up and whenever you feel exhausted shift gears and just keep going! And last but certainly not least I would like to thank my family and friends. For always supporting me and enjoying the spare time I got to spend with them. Without you I would never be able to shift gears up again!

Thank you all, without your help I would not be where I am now! I hope I can continue fulfilling this passion for a long time!

Eindhoven, August, 2013

Ferry Schoenmakers

Abstract

The Dutch company WittyWorx [1] has developed a table top robot which is aimed at being a social play robot for small children with the aim of entertaining and educating them in the time they otherwise would have spent sitting passively in front of the television. Inside the robot is a hexapod like structure which enables the robot to move in six degrees of freedom. During my graduation I worked on the motion control part of this system with the aim of designing a real time inverse kinematics algorithm, identify its workspace and design and implement motion controllers to let the platform move smoothly and quietly.

The structure of the platform resembles a hexapod structure such as is found in Stewart Gough type platforms which are widely used in for example flight simulators. However, the design is different in the way the links are extended in order to keep the system compact and robust. Furthermore piezo motors are used as actuators instead of hydraulic prismatic links again for robustness concerns but also for user convenience to keep the produced noise to a minimum. Because of this exotic structure, the inverse kinematics are complex and thus an approximation to this inverse kinematics algorithm is derived with sufficient performance. The reachable workspace of the platform is analyzed and identified so that users can create motion profiles while easily identifying there feasibility. Since the system has to be kept low cost a commercial micro controller, the MBED, is used for control purposes. Data acquisition, feedback controllers and feedforward systems are developed on this platform which let the platform move fast and smooth. Finally a method has been adopted to artificially increase the encoder resolution.

Contents

Chapter 1

Introduction

1.1 Background

Parents nowadays have their hands full with work, household and numerous other tasks. Therefore several times a day their children have to be amused so the parents can complete these tasks. Unfortunately, a lot of times the children end up in front of the television and are passively entertained. A missed opportunity according to WittyWorx, the company that developed an interactive play robot called IXI-Play [1]. Children would be better off when being entertained with this interactive robot. According to the company's vision the children will then actively interact with the robot, learn several skills while doing so and stay captivated because of the versatility of the robot.

1.2 Motivation

To accomplish this a robot platform with different sensors and 6 degrees of freedom (DOF) is developed as low cost as possible. This way the platform will be affordable to the wider public. IXI-Play, depicted in Figure 1.1, is a table top device with a flexible body and a hard shell head. The head is able to move in 6 DOF letting the flexible body move with it. With the launch of IXI-Play, also an app-store will be launched where several apps and games for IXI-Play can be downloaded. The possibility to create your own apps will be included too. When creating apps for IXI-Play, prescribing motions for the robot should be easy and straightforward. Therefore it is required that the platform can calculate its inverse kinematics in real-time and control the motions requested.

1.3 Project objective

The first two prototypes of IXI-Play are built by WittyWorx and the first electronics are integrated as well. In principle IXI-Play can already move by actuating each motor independently. However decoupled control of the robot is demanded so that the end user can present trajectories for the head and don't bother about the kinematics.

- A first objective therefore is to develop an algorithm that calculates the inverse kinematics in real-time.

Furthermore the robot should move as natural as possible and should also be robust since it is interacting with young children. Grabbing the robot's head and forcing it in another direction while moving should not destroy the robot! Therefore the mechanical design and used actuators are unique. Because of the special actuators and several end-supports that prevent the robot from being destroyed, it is demanded to identify the reachable workspace of the robot.

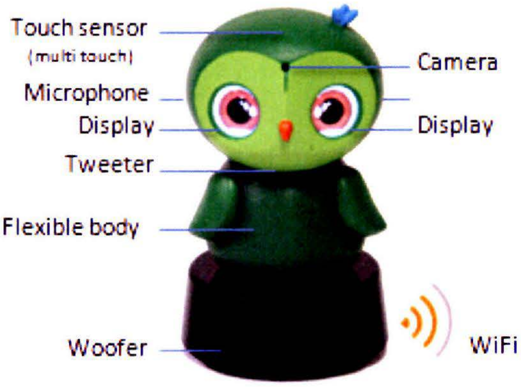


Figure 1.1: The IXI-Play platform.

- Another objective is therefore to identify this reachable workspace to verify the mechanical design and to visualize feasible movement trajectories beforehand.

Finally the movements should be controlled so that the robot will actually perform the movements requested. Other then in casual mechatronic devices however, the tracking error of this robot is not top priority. A little deviation from the requested trajectory is allowed, as long as the movements look fluent and the amount of noise produced by the system is minimized.

- A last objective therefore is to make the movement of the robot smooth and as quiet as possible.

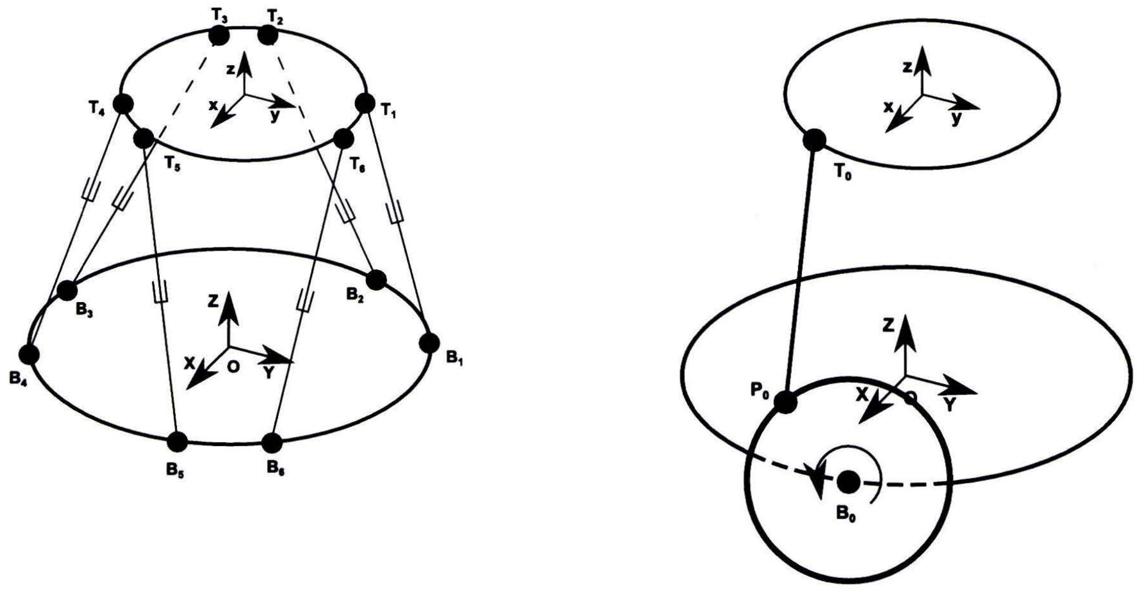
Chapter 2

Structural design

For IXI-Play to be able to move with 6 degrees of freedom a design inspired on a hexapod is used. This type of platform is often seen in flight simulators as well. Piezo motors actuating rotating wheels are used as actuators for the robot to be robust and silent.

2.1 Stewart Gough platform

A Stewart Gough platform is a parallel type of manipulator widely used in for example flight simulator platforms [2]. The platform consists of a fixed base B and a movable platform T. Between the two are six prismatic actuators. With this configuration the top platform can perform the three linear movements: lateral, longitudinal and vertical displacement and the three rotations: roll, pitch and yaw. Figure 2.1a shows a schematic drawing of such a standard Stewart Gough platform.



(a) A standard Stewart Gough platform.

(b) A new type of Stewart Gough platform.

Figure 2.1: Two parallel manipulators.

2.2 New type of Stewart Gough platform

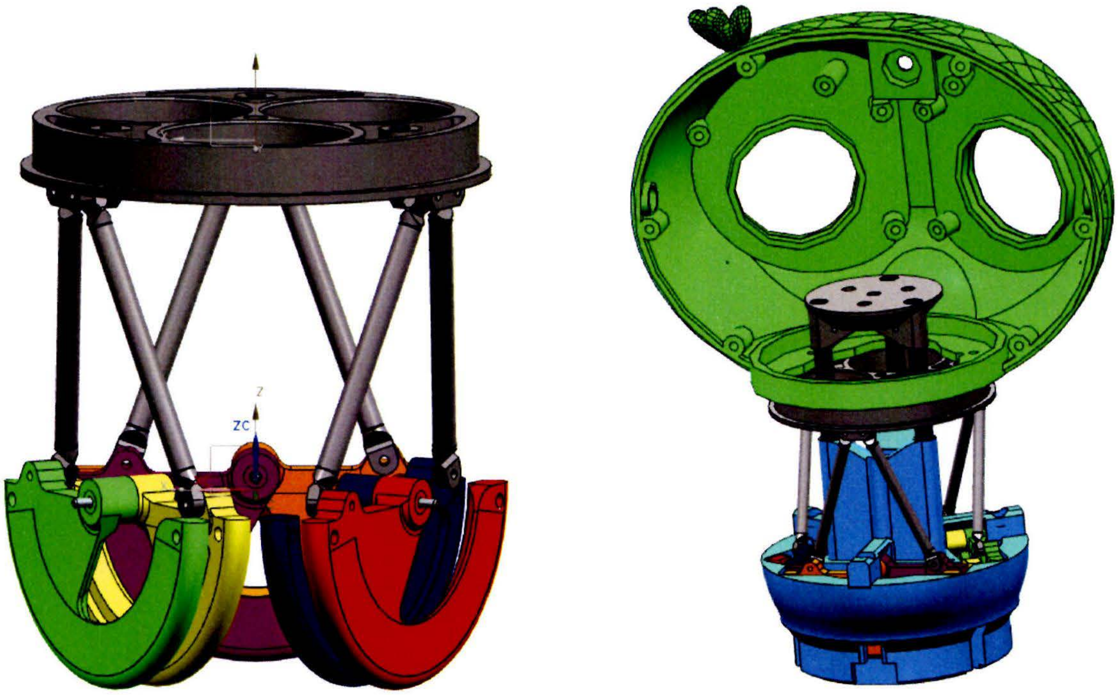
The mechanism of IXI-Play is inspired on a Stewart Gough platform, but essentially different in the way the base and movable platform are connected. No prismatic actuators are used, instead six fixed length rods are each connected to the movable platform and to separate wheels which are fixed to the base. By rotating the wheel, the distance between the wheel center and the joint on the movable platform is varied. The wheels are actuated by piezo motors which are covered in section 2.3. Figure 2.1b shows a schematic drawing of this mechanism. Only one link and wheel are shown for clearness. This new design is chosen to make the complete system more compact, robust and silent. The wheels are "pushed" around by piezo motors and these will simply slip when the robot is forced in another direction while moving. Thus, a child will not destroy the robot by grabbing it, holding it, or forcing it in another direction. Furthermore no gearboxes are present in the system and the piezo motors operate at an inaudible frequency, which make the system silent.

A complete CAD drawing of IXI-Play's mechanism is shown in Figure 2.2a. The six different wheels are visualized with six different colors. Every set of two wheels share the same axis for rotation. At the end of each wheel a push rod is attached which is again connected to the gray ring on top. This ring is the moving platform to which the head will be attached. The piezo motors are pushed against the wheels on the bottom of the wheel. At the side of the wheels a reflective strip with increments is placed to create an optical encoder for position feedback.

A more complete setup is show in Figure 2.2b. Half of the head and the flexible body are removed for visibility. The end supports are also visible in this figure. First the blue base in which the wheels are attached, has a certain height in the middle. This prevents the ring from going down too much in the z-direction. The z-direction is indicated in Figure 2.2a. On top of this blue support are three gray tubes which hold another gray end stop. This stop limits the stroke in the positive z-direction. The three tubes themselves provide a limit for the rotation around the z-axis. Furthermore the stroke and rotation in both x and y direction are also limited by these tubes. Because of all these end stops, a user can force the head in any direction without forcing the mechanism beyond its limits. An other purpose of the three tubes is the guidance of cables. Several cables have to run from the eyes, webcam, touch sensors and microphones to the bottom of IXI-Play where the rest of the electronics are located. Running these cables through these tubes keeps them clear from moving parts.

2.3 Piezo motors

The motors in IXI-Play are piezo electric motors from Elliptec AG [3] of type X15G [4]. This special type of motor operates with the use of a piezo crystal[5]. The piezo crystal is actuated at a certain frequency which makes the structure of the Elliptec motor resonate. Because of the shape and the resonance mode of the system, the tip of the Elliptec motor will make elliptical movements. With these movements the motor is pushing the wheel around with very tiny steps ($1 - 3$ μm) at a frequency of 80 or 100 kHz. Figure 2.3 shows stills from an animation of this vibration. The Elliptec motor is attached to a spring, visible on the left, which presses the motor against the wheel on the right. The elliptical movements will push the wheel around. The motor operates at frequencies of 80 or 100 kHz. At the first frequency the motor will vibrate in such a way that it will push the wheel forward, while the other frequency makes the motor push the wheel in the other direction. The frequency response is shown in Figure 2.4. The amplitude indicated in a percentage shows the speed of the motor which has a typical maximum of about 300 mm/s. This speed represents the speed of the surface moving underneath the tip of the motor. For a wheel, the motor needs to travel the complete circumference of the wheel for a complete rotation. The wheels in IXI-Play have a circumference of $20 * \pi = 62.83$ mm and thus the wheel can make a full rotation within 0.2 s. The wheel however will at most make a half rotation, which could theoretically be done in 0.1 s. If all the wheels would do this, the platform should in theory be able to move up and down five times a second and thus allows quick movements. This maximum



(a) CAD drawing of the IXI-Play mechanism.

(b) IXI-Play without its body.

Figure 2.2: IXI-Play mechanism.

speed however is not recommended, since the motors cannot deliver a lot of force anymore at these speeds. At high speeds the typical force a motor will deliver is about 200 mN, while the maximum holding force, i.e. the force it can hold while not powered, is about 800 to a maximum of 1200 mN. Converted to torques on a 20 mm wheel, this means a maximum holding torque of 8 mNm to 12 mNm. These and more specifications are shown in the data sheet in Appendix A. The frequency response of a motor as shown in Figure 2.4 is slightly different for every motor due to minimalistic differences in the motor's shape or spring characteristics. So for every motor a frequency sweep has to be performed to determine at exactly what frequency it will reach maximum speed. This is also done in IXI-Play. Each time the robot starts a frequency sweep is performed and the found resonance frequency is stored for each motor. The motors will then be actuated at this frequency. The sweep only takes up about two seconds. The reason this sweep is done every time is because of the possibility of wear. The resonance frequency might change in time when the motors or wheels wear out. By identifying the resonance frequency every time, this effect is counteracted.

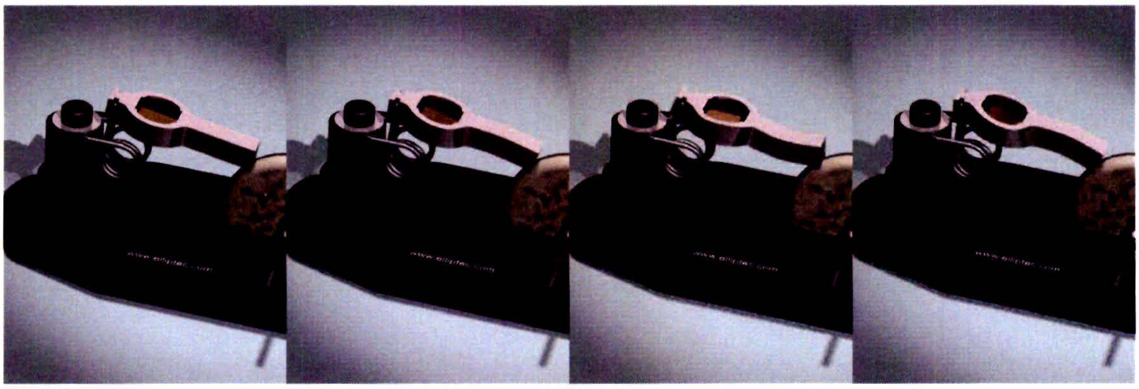


Figure 2.3: Vibration mode of Elliptec motor as shown on: www.elliptec.com.

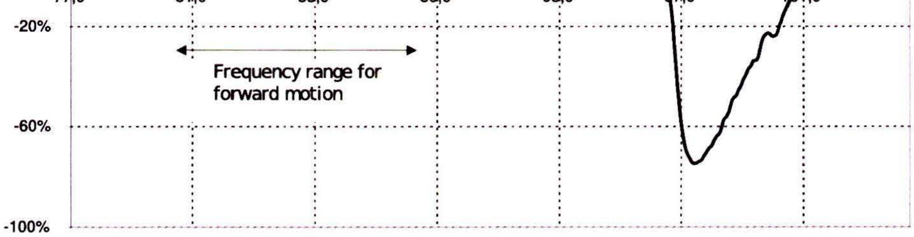
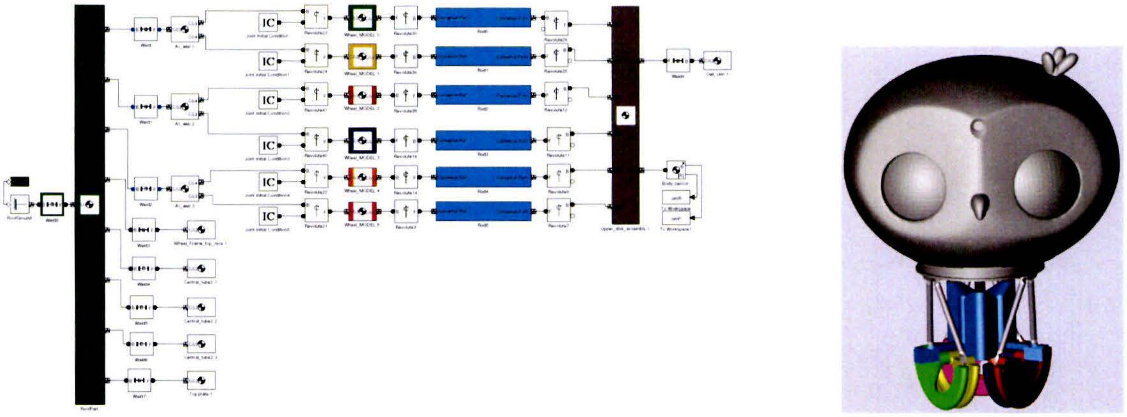


Figure 2.4: Frequency response of Elliptec motor.

Chapter 3

SimMechanics model

In a first attempt of creating a model of the dynamics of IXI-Play, a SimMechanics model was made in Simulink software from MathWorks [6]. With this multibody simulations software several characteristics of the system can be addressed. For example forward and inverse kinematics, force analysis and full motion simulation. When a decent CAD model exists with the right boundary conditions and materials assigned, this model can be directly imported and converted to a SimMechanics model in Simulink. Figure 3.1a shows the model in Simulink and Figure 3.1b shows the visualization of the model. Figure 3.1a is also shown in Appendix B for better visibility. The usage of this model will be mentioned in the following sections.



(a) SimMechanics model of IXI-Play.

(b) Visualization of SimMechanics model.

Figure 3.1: SimMechanics model of IXI-Play.

Chapter 4

Inverse kinematics

IXI-Play's mechanism is a parallel type of robot manipulator. Where for a serial manipulator the forward kinematics are straightforward and the inverse kinematics are complicated, for a parallel manipulator the opposite is true [7, 8, 9, 10, 11, 12]. However, because of the new design of the platform, the inverse kinematics are not as easy to solve either. For a standard Stewart Gough platform, the inverse kinematics are calculated as follows.

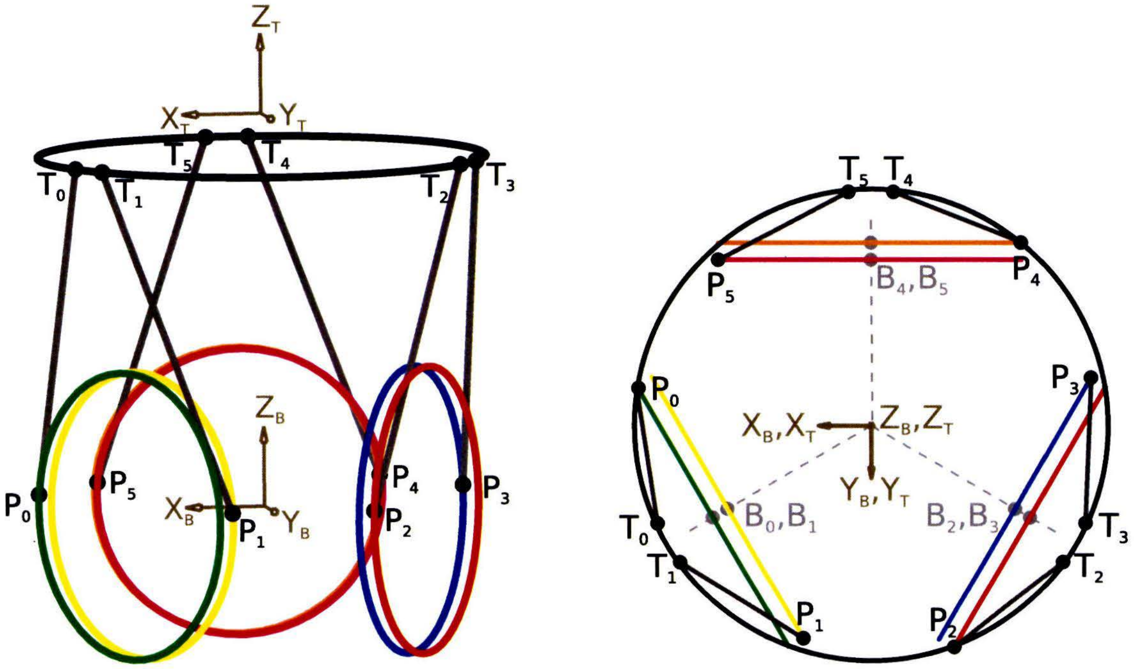
4.1 Inverse kinematics of standard Stewart Gough platform

Consider a fixed base B with attached coordinate frame X_B, Y_B, Z_B. Similarly consider a moving platform T with attached coordinate frame X_T, Y_T, Z_T. Now let the 3x3 rotation matrix R describe the rotation of X_T, Y_T, Z_T w.r.t. X_B, Y_B, Z_B and the vector D describe the translation of frame X_T, Y_T, Z_T with respect to frame X_B, Y_B, Z_B. Six links are attached between the base and the platform. The joint coordinates of those links in the base are defined w.r.t X_B, Y_B, Z_B as $B_i = [B_{x,i}, B_{y,i}, B_{z,i}]^T$; $i = 1, 2, ..., 6$ and in the moving platform with respect to X_T, Y_T, Z_T as $T_i = [T_{x,i}, T_{y,i}, T_{z,i}]^T$; $i = 1, 2, ..., 6$. Then, the link-lengths of the platform can be calculated as:

$$l_i = \sqrt{||RT_i + D - B_i||}; \quad i = 1, 2, ..., 6 \tag{4.1}$$

4.2 Analytical approach for IXI-Play platform

This inverse kinematics equation is simple since the joints are fixed to the platform and base and are connected using prismatic links. In the new design however, the actuation of the platform is not an extension of a link but a rotation of a wheel. Therefore there is a more complex relation between the joint coordinates and the actuator inputs. In Figure 4.1, a schematic of the new platform is depicted. The wheels are attached to the base at points B_i where $i = 0, ..., 5$ indicates the number of the wheel. The axes of rotation are shown in Figure 4.1b by the gray dashed lines. One can also observe the frame definition in this Figure. The negative y-axis coincides with the axis of rotation of wheel 4 and 5. The positive y-axis lies exactly in between the other two axes of rotation. This way it aligns with the direction IXI-Play is looking at. In other words, if one would imagine to be inside IXI-Play and looking through its eyes, the y-axis is straight forward and the x-axis points to the right. This definition is chosen for the sake of usability when the user wants to prescribe motions to the platform. When a wheel rotates, the corresponding point P_i moves in a circular motion around B_i. Subsequently the distance between T_i and B_i varies as a result. Since P_i and T_i are joints with two degrees of freedom, only the distance between T_i and P_i is fixed for a given position of the actuator wheel and is equal to the link length. Other degrees of freedom remain. Therefore the distance between T_i and B_i is not fixed as a result, where for a standard Stewart Gough platform, this distance is fixed for a given actuator input (link length). When all six wheels in the system are fixed however, the platform has no degrees of freedom left



(a) Schematic 3D view of IXI-Play.

(b) Schematic top view of IXI-Play.

Figure 4.1: Schematic view of IXI-Play.

and therefore a desired configuration can be assigned by defining a certain rotation for all the wheels. To find out what wheel angles need to be set in order to arrive at a desired configuration, we want to derive an inverse kinematics algorithm that returns the wheel angles depending on the platform configuration.

Consider again Figure 4.1. For a given configuration of the platform T, the 3D position of T_i w.r.t. X_B, Y_B, Z_B is known. Since the link connecting T_i and P_i is fixed in length, all the possible positions of P_i in 3D space are described by a sphere S with radius L, the link length. Note that in practice the link length is not exactly fixed because of the flexible hinges on either side of the links. However, this slight deviation in link length is neglected. A circle C circumscribing B_i with radius r (wheel radius) defines all possible positions of the point P_i as well. This circle intersects the sphere at two positions in 3D space giving two solutions for the position of point P_i. But it only does so if $||T_i - B_i|| \leq ||L + r||$. Otherwise the desired configuration is not feasible. To visualize this, take a look at Figure 4.2. Here the sphere S, the circle C and the two intersection points P_0, $P_0.'$ are shown for a single link.

In an analytical way, this is described as follows. A circle in 3D space is formulated as (4.2). Here vector c is a vector pointing to the center of the circle. Vectors a and b are two unit vectors perpendicular to each other and to c and describe the plane the circle is lying in.

$$\begin{aligned} x(\theta) &= c_1 + rcos(\theta)a_1 + rsin(\theta)b_1 \\ y(\theta) &= c_2 + rcos(\theta)a_2 + rsin(\theta)b_2 \\ z(\theta) &= c_3 + rcos(\theta)a_3 + rsin(\theta)b_3 \end{aligned} \tag{4.2}$$

Furthermore a sphere can be described by (4.3), where x_s, y_s, z_s are 3D coordinates presenting the center of the sphere and r equals the radius of the sphere. All values x, y, z that fulfill this equation are on the outer shell of the sphere.

$$r^2 = (x - x_s)^2 + (y - y_s)^2 + (z - z_s)^2 \tag{4.3}$$

Now all values x, y, z that lie both on the sphere's shell and on the circle, represent intersections of the sphere and the circle. If we substitute (4.2) into (4.3) and move r^2 to the other side of the

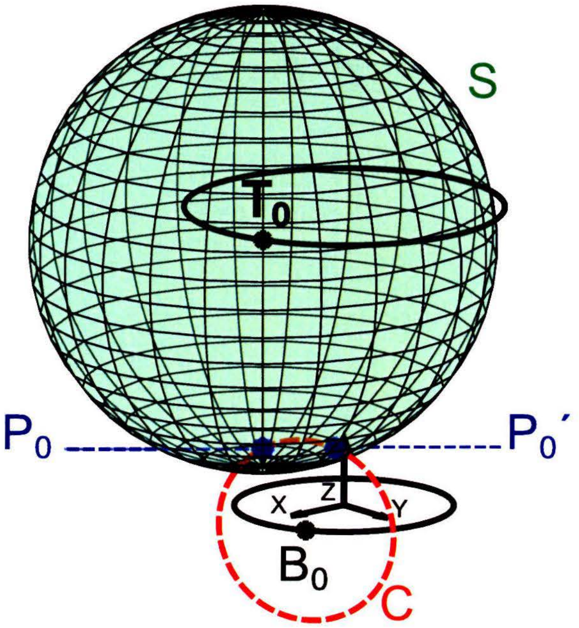


Figure 4.2: A visualization of the sphere S intersecting with circle C

equation, we get (4.4).

$$\begin{aligned} 0 =& (c_1 + rcos(\theta)a_1 + rsin(\theta)b_1 - x_s)^2 + \\ & (c_2 + rcos(\theta)a_2 + rsin(\theta)b_2 - y_s)^2 + \\ & (c_3 + rcos(\theta)a_3 + rsin(\theta)b_3 - z_s)^2 - r^2 \end{aligned} \tag{4.4}$$

For any given x_s, y_s, z_s, solving the equation will result in an angle θ that represents the angle on the circle where the intersection occurs. As can be seen from Figure 4.2 this are typically two angles. However, since a wheel in the system can only rotate for a maximum of π radians, either from $\frac{\pi}{2}$ to $1\frac{\pi}{2}$ or from $-\frac{\pi}{2}$ to $-1\frac{\pi}{2}$, only one of the solutions will be a feasible one for an actuator. Unfortunately, there is no simple closed form solution to equation (4.4). It can be solved using numerical approaches, but since we want to implement this inverse kinematics algorithm in a control loop it should be solved very fast. So now that the inverse kinematics are solved, the calculations need to be made fast.

4.3 Approximation of the inverse kinematics algorithm

To make the inverse kinematics equations fast, it is chosen to fit a polynomial model to the results and try to find a simpler and faster model. First a grid of data was created using Matlab and the inverse kinematics algorithm from (4.4). On an x, y, z grid of $15 \times 15 \times 15$ all θ values are calculated. Data for one motor is shown in Figure 4.3. Since the data is four dimensional (x, y, z, θ), visualization is difficult. Therefore the following representation is chosen. At a fixed coordinate z a grid of size 15×15 of x, y coordinates is created and the corresponding motor angles are calculated. This is done for 15 fixed values of z and the results are stacked in one plot. So every plane in Figure 4.3 represents data for a fixed z value while the actual height in the plot represents the motor angle. Since the planes don't intersect, one can easily distinguish one plane from another and thus the four dimensional results can be shown in one single figure. The bounds on the x, y, z grid are chosen, such that most of the reachable workspace of the joint is covered. For example, because of the poles shown in Figure 2.2b, the ring is not able to move more than 16 mm in x or y direction from its normal position. The normal position is defined as the position shown in Figure 2.2. I.e. the wheels are horizontal and make a zero angle with the x, y plane. Furthermore in the z direction movement is restricted to 24 mm. Thus, the grid in x, y direction has range $[x_{norm} - 8\ ,\ x_{norm} + 8]$ mm and $[y_{norm} - 8\ ,\ y_{norm} + 8]$ mm while the grid in z direction has range $[z_{norm} - 12\ ,\ z_{norm} + 12]$ mm. Because of this reason one can also see that the bottom three planes in Figure 4.3 are not complete. The missing points are unfeasible points since here the sphere and circle do not intersect anymore.

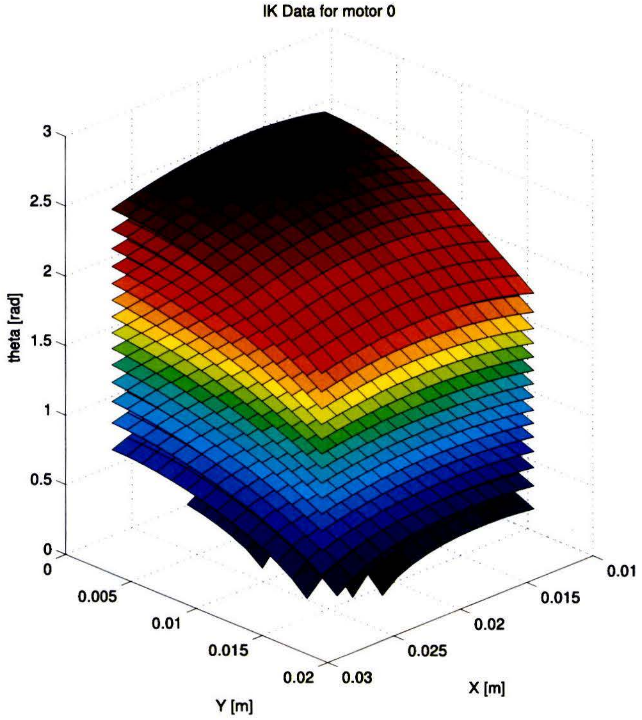


Figure 4.3: Angles for motor 0 as a function of sphere center coordinates

This data was then used to fit a model to. On each plane a 2D surface model was fitted using a second order function (4.5).

$$\theta = P_{00} + P_{10}x + P_{01}y + P_{11}xy + P_{20}x^2 + P_{02}y^2 \tag{4.5}$$

A resulting fit for a fixed z value of -1.34 mm from its normal position is shown in Figure 4.4. The shown plane represents the inverse kinematics result, while the red stars indicate the surface fit. After all surfaces are successfully fitted, the polynomial parameters $P_{00}, P_{10}, P_{01}, P_{11}, P_{20}, P_{02}$ are plotted as a function of the z-coordinate to see how they change per plane. The results are shown in Figure 4.5 indicated by the blue stars. Again a polynomial is fitted to this data, so that the parameters $P_{00}, P_{10}, P_{01}, P_{11}, P_{20}, P_{02}$ can be predicted depending on the z coordinate. For this fit a 1D fourth order polynomial is used and the results are also shown in Figure 4.5 indicated by the red lines.

After all fits are successful, a polynomial model is found that describes the inverse kinematics algorithm. The model is the same for each motor, only the parameter values are different. The complete model is given in (4.6). For each motor 30 parameters are stored. The input to the model are the coordinates x_s, y_s, z_s that describe the center of the sphere i.e. the link joint position on

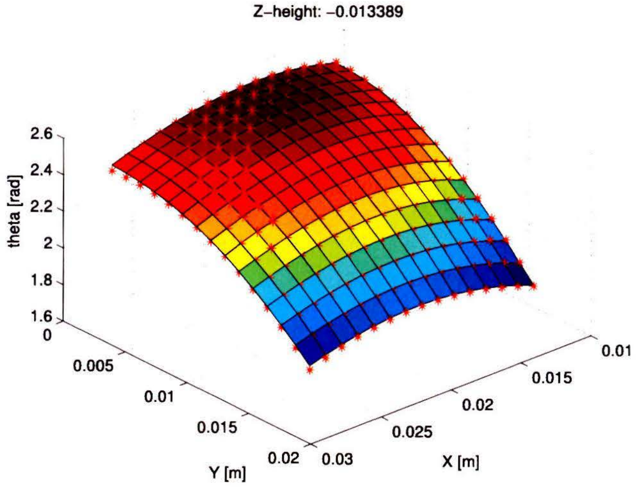


Figure 4.4: 2D fit using a second order polynomial

the moving platform. The output is the corresponding motor angle.

$$\begin{aligned}
P_{00} &= p_1 z_s^4 + p_2 z_s^3 + p_3 z_s^2 + p_4 z_s + p_5 \\
P_{10} &= p_6 z_s^4 + p_7 z_s^3 + p_8 z_s^2 + p_9 z_s + p_{10} \\
P_{01} &= p_{11} z_s^4 + p_{12} z_s^3 + p_{13} z_s^2 + p_{14} z_s + p_{15} \\
P_{20} &= p_{16} z_s^4 + p_{17} z_s^3 + p_{18} z_s^2 + p_{19} z_s + p_{20} \\
P_{11} &= p_{21} z_s^4 + p_{22} z_s^3 + p_{23} z_s^2 + p_{24} z_s + p_{25} \\
P_{02} &= p_{26} z_s^4 + p_{27} z_s^3 + p_{28} z_s^2 + p_{29} z_s + p_{30}
\end{aligned} \tag{4.6}$$

$$\theta = P_{00} + P_{10} x_s + P_{01} y_s + P_{11} x_s y_s + P_{20} x_s^2 + P_{02} y_s^2$$

Altogether, suppose we now want to describe a configuration for the platform with position x_e, y_e, z_e and orientation $\alpha_e, \beta_e, \gamma_e$. The position indicates the 3D position of the point to be controlled (PTC) which is indicated in Figure 4.6 and corresponds with the center of mass of the top ring. Its coordinates are expressed in frame X_{TF}, Y_{TF}, Z_{TF}. This frame is fixed in 3D space, while frame X_T, Y_T, Z_T is moving with the top ring of the platform. When the platform is in its normal position as shown in Figure 2.2, the two frames X_T, Y_T, Z_T and X_{TF}, Y_{TF}, Z_{TF} coincide and thus the position of the PTC with respect to frame X_T, Y_T, Z_T equals $[0; 0; 0]$. In Figure 4.6 the platform is set to a position of $[3; 3; 3]$ mm and orientation $[\frac{\pi}{36}; \frac{\pi}{36}; \frac{\pi}{36}]$ $radians$ defined as a roll, pitch yaw rotation. First T_i^{TF}; $i = 0...5$, the 3D positions of T_i with respect to X_{TF}, Y_{TF}, Z_{TF} need to be determined. T_i^T, the positions of T_i with respect to X_T, Y_T, Z_T are fixed and known and thus T_i^{TF} can be found using a coordinate transformation (4.7) with transformation matrix (4.8). Note that in (4.8) $cos(\theta)$ and $sin(\theta)$ are replaced by c_θ and s_θ respectively for ease of notation.

$$\begin{bmatrix} T_i^{TF}(1) \\ T_i^{TF}(2) \\ T_i^{TF}(3) \\ 1 \end{bmatrix} = H_{trans} \times \begin{bmatrix} T_i^{T}(1) \\ T_i^{T}(2) \\ T_i^{T}(3) \\ 1 \end{bmatrix} \tag{4.7}$$

$$H_{trans} = \begin{bmatrix} c_{\beta_e} c_{\gamma_e} & -c_{\alpha_e} s_{\gamma_e} + s_{\alpha_e} s_{\beta_e} c_{\gamma_e} & s_{\alpha_e} s_{\gamma_e} + c_{\alpha_e} s_{\beta_e} c_{\gamma_e} & x_e \\ c_{\beta_e} s_{\gamma_e} & c_{\alpha_e} c_{\gamma_e} + s_{\alpha_e} s_{\beta_e} s_{\gamma_e} & -s_{\alpha_e} c_{\gamma_e} + c_{\alpha_e} s_{\beta_e} s_{\gamma_e} & y_e \\ -s_{\beta_e} & s_{\alpha_e} c_{\beta_e} & c_{\alpha_e} c_{\beta_e} & z_e \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.8}$$

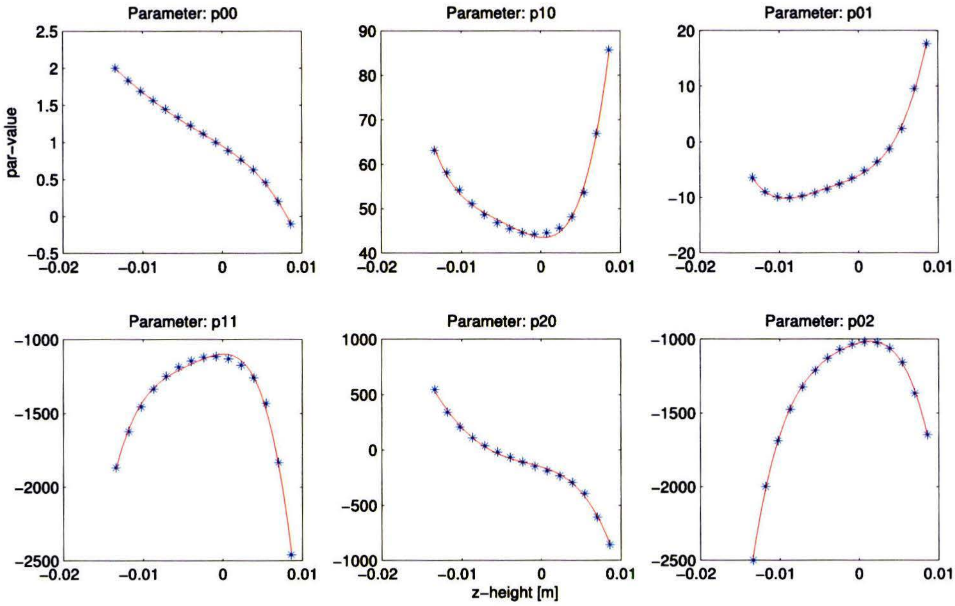


Figure 4.5: polynomial parameters as a function of z with fourth order polynomial fit

Now that all T_i^{TF} are known, one can substitute the coordinates of each T_i^{TF} in (4.6) with the corresponding motor parameters and find the corresponding motor angles. For this given configuration the corresponding motor angles read:

$$[\theta_0, ..., \theta_5] = [0.31,\ 0.17,\ 0.33,\ 0.43,\ 0.19,\ -0.038]\ radians \tag{4.9}$$

4.4 Validation of the inverse kinematics algorithm

Now that an approximation to the inverse kinematics is created, it would be useful to address its performance. So whenever the motor angles are calculated using (4.6), how much does the final configuration of the platform using these motor angles, deviate from the original desired configuration. An attempt is made to evaluate this performance. The IXI-Play platform itself is lacking sensors to accurately measure the configuration of the platform. Only sensors for motor angles are present. Therefore the SimMechanics model from chapter 3 is used for validation instead. As stated by [7, 8, 9, 10, 11, 12], the forward kinematics of a parallel type manipulator are difficult. Especially with this type of manipulator where a certain actuator input does not directly define the distance $||T_i - B_i||$, the forward kinematics are hard. Since the forward kinematics are not required in the motion control software of IXI- Play, there is no necessity for a real time forward kinematics model. The SimMechanics model is able to perform forward kinematics off line, so this model is used instead for analysis of the inverse kinematics performance. One can define the motor angles in the model and with the use of a body sensor block, the resulting configuration of the platform can be retrieved. Applying this approach to a wide range of platform configurations gives insight in the inverse kinematics performance. In chapter 5 the reachable workspace for the IXI-Play platform is addressed. For this a grid in both x, y, z space and α, β, γ is created on which the feasibility of all configurations is tested. From this massive grid almost 185,000 feasible configurations are discovered. From all these feasible configurations, 1000 configurations are randomly chosen to test the inverse kinematics algorithm on. At each configuration the error in $x, y, z, \alpha, \beta, \gamma$ is calculated as the difference between the desired configuration and the result of

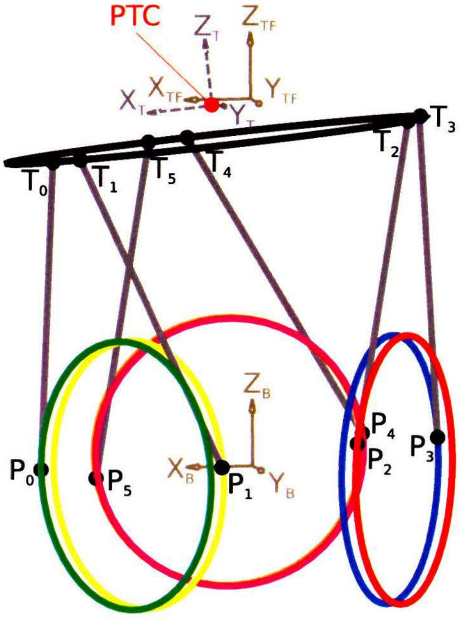


Figure 4.6: The IXI-Play platform in a desired configuration.

the forward kinematics calculation of the SimMechanics model. The resulting errors are shown in histograms in Figure 4.7. As can be seen, 95 % of all position errors is smaller than 1 *mm* and 95 % of all rotation errors is smaller than 0.025 *rad* or 1.4 *deg*. So it can clearly be stated that the configuration errors due to the inverse kinematics approximation are small. For the spectator these small errors are almost unnoticeable. Therefore the inverse kinematics approximation is accepted to implement in the software.

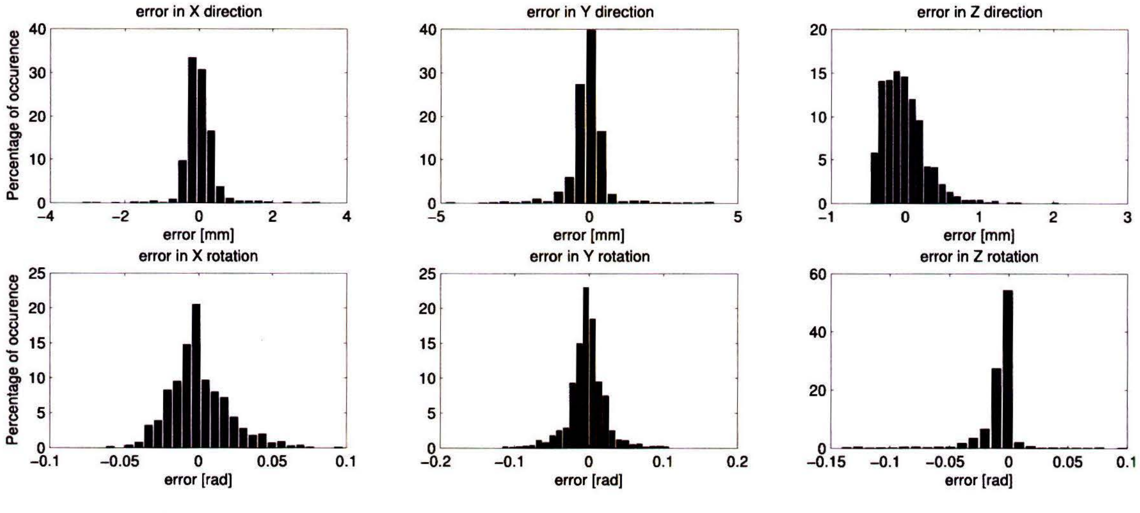


Figure 4.7: Error distributions for 1000 inverse kinematics calculations.

Chapter 5

Workspace

Identifying the reachable workspace of the platform is useful and demanded for several reasons[13, 14, 15]. When identified, one can evaluate how the platform can move and what movements it cannot do. Furthermore, for robustness reasons several end supports are included in the design. If designed correctly, all limitations on the reachable workspace should come from these supports and not from limitations of the mechanism. If the mechanism would put limits on the workspace, it means that the user could directly put loads on the mechanism and possibly damage it. So identifying the reachable workspace validates the mechanical design at the same time.

5.1 Workspace identification

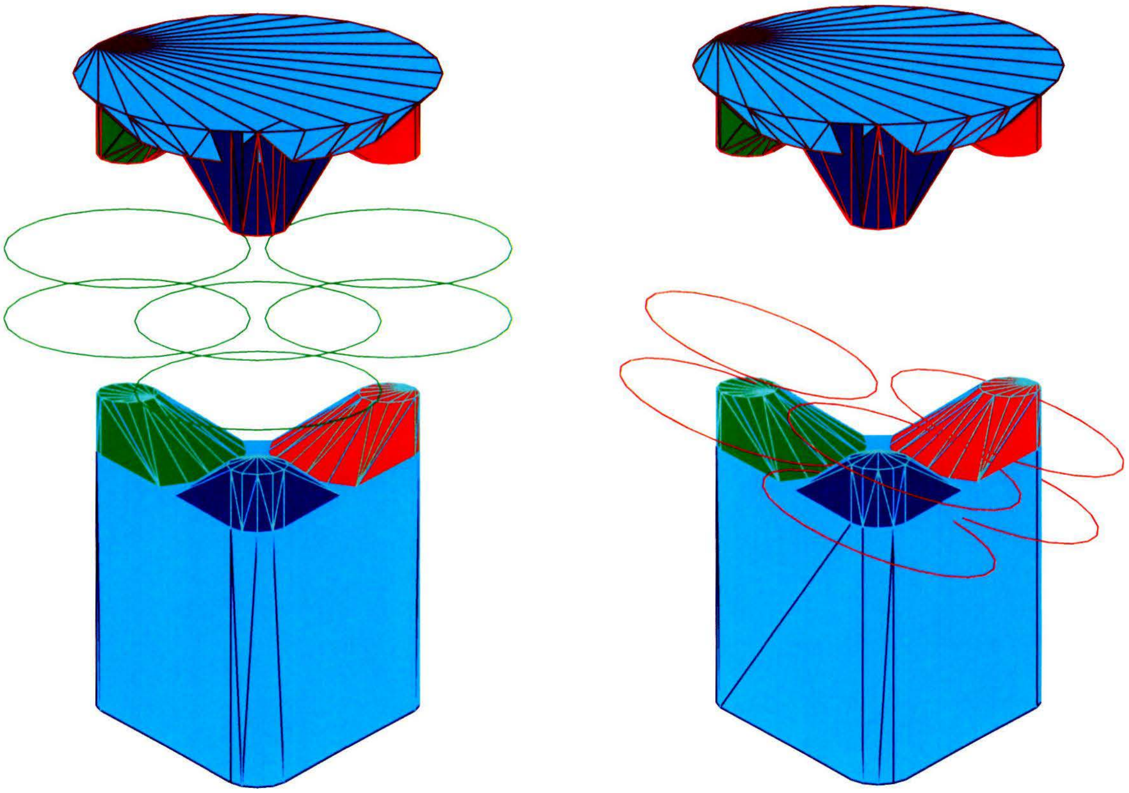
For identifying the reachable workspace of the platform, a series of feasibility tests is evaluated on a large set of configurations in 6D space $(x, y, z, \alpha, \beta, \gamma)$. The following tests are evaluated:

- Check if the top ring if free from intersection with the top or bottom support
- Check if the support tubes are inside the corresponding holes in the top ring
- Check if the links are free of intersections with the axes covering
- Check if the angles the links make with the top ring are within a feasible range
- Check if the angles of the wheels are within a feasible range

Whenever all tests pass, the configuration is considered feasible and stored as such. If one of the tests fail, it is stored as a failed configuration and what test made it fail. This way afterwards it can be analyzed why configurations are rejected. The individual tests work as follows.

Top or bottom support intersection

The top ring cannot be inside the top or bottom support of course. This test verifies if this is the case. This is done using the inhull() function in Matlab [16]. This function tests if a set of points is within a convex hull. The convex hull of a set X of points in Euclidean space is the smallest convex set that contains X. When we define the top and bottom support by such convex hulls and define the top ring by a set of points, we can test whether these points are within the convex hulls. If so, the configuration is considered unfeasible. Figure 5.1 shows two test situations. One where the configuration is feasible according to this test and one where it is unfeasible. As can be seen the top and bottom support consist of eight convex hulls in total which are shown in different colors for clarity. The moving ring is represented by six rings. These rings represent the top and bottom edges of the three holes in the moving ring. These edges are the first part of the moving ring to intersect with the supports. The outer edges of the moving ring could also intersect with the supports in extreme configurations, but in this case the ring will already have to intersect with one or more of the three support tubes and thus this configuration is already classified unfeasible.



(a) Feasible configuration.

(b) Unfeasible configuration since the rings intersect with the bottom support.

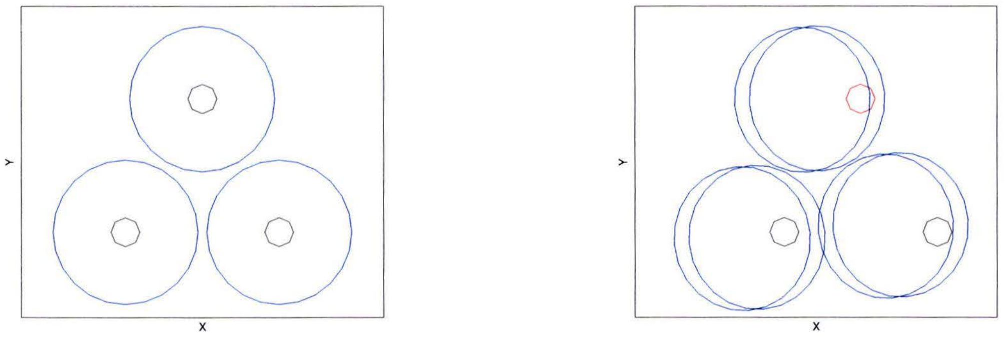
Figure 5.1: Check for intersections with top and bottom supports using convex hulls.

Support tubes intersection

To test whether the moving top ring intersects with one or more of the three support tubes, a similar test using the inhull() function is evaluated. Although this test is evaluated solely on the x, y plane. Without any intersections, all three tubes should have their x, y coordinates lying within the corresponding circles. If one of the tubes is partially outside of its corresponding circle, the configuration is classified unfeasible. For this test, the six circles representing the edges of the three holes in the moving top ring are each a 2D convex hull. The points representing the three support tubes should all be in their corresponding hull for a feasible configuration. Figure 5.2 shows two situations again, where the blue circles represent the six edges and the black circles represent the support tubes. The first situation shows all tubes are within their corresponding hull, the second situation shows one support tube partially outside one of its hulls. This tube is shown in red and the configuration is classified unfeasible. Note that in Figure 5.2a only three circles are visible since there was no rotation α or β applied and thus the bottom three edges coincide with the top three edges.

Link axes covering intersection

On top of the wheel axes sits a piece of plastic functioning as a support as well. It limits the rotation of a wheel. Since a wheel is not fully circular but only half, the piezo motor could push

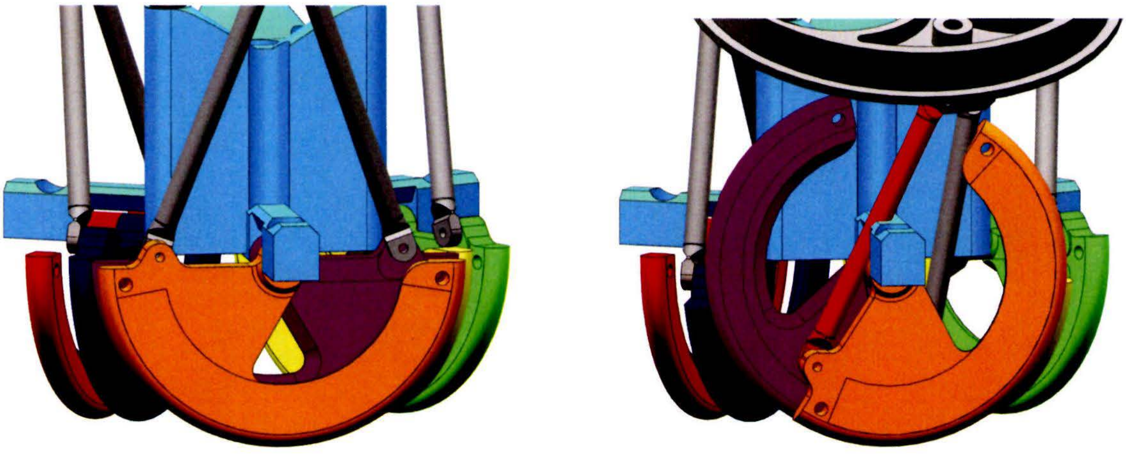


(a) Feasible configuration.

(b) Unfeasible configuration since one of the tubes intersects with a top ring edge.

Figure 5.2: Check for intersections with support tubes.

the wheel too far until it looses contact with the wheel. To avoid this, the axes covering limits the wheel rotation. However, in some extreme configurations one of the links could possibly intersect with this covering. Such a situation is shown in Figure 5.3b where the intersecting link is shown in red, while Figure 5.3a shows a normal feasible situation. For a known configuration the 3D position of the two joints of a link are known. A 3D line can thus be defined between these points. Another 3D line is defined as the axis of rotation for the wheel. Whenever the shortest 3D distance between these two lines becomes too small, i.e. smaller than the radius of a link plus the radius of the axes covering, the configuration is considered unfeasible.



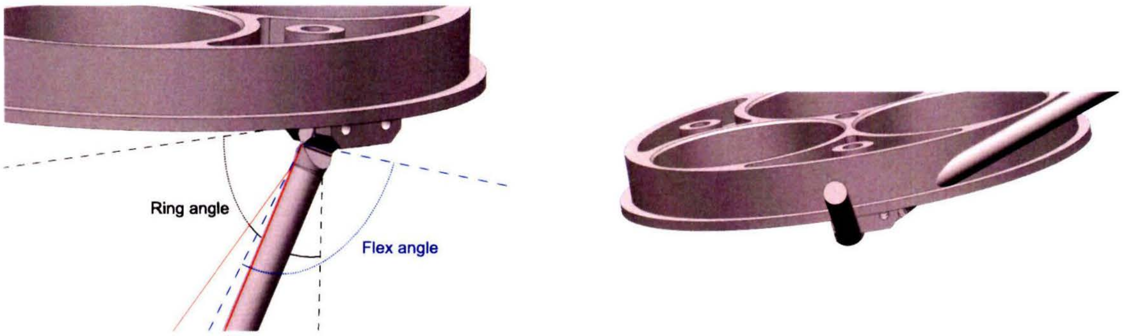
(a) Feasible configuration.

(b) Unfeasible configuration because one link intersects with an axis covering.

Figure 5.3: Check for intersections of links with axes covering.

Link angles with top ring

The links are connected to the top moving ring by a flexible and a rotational joint and thus obtain two rotational degrees of freedom. However, these angles cannot be beyond certain limits. On one hand because they would intersect with the top ring. On the other hand because the flexible joint provides a limited rotation. Thus, for each configuration tested, these angles need to be within the feasible range. The flex-angle, as defined in Figure 5.4a has a range of $[-\frac{\pi}{4}, \frac{\pi}{4}]$. The ring-angle, which is defined in Figure 5.4a as well has range $[-\frac{\pi}{2}, \frac{\pi}{2}]$. Figure 5.4b shows two unfeasible positions for the links because of these angles being beyond the defined limits.



(a) Defintion of the ring-angle and flex-angle.

(b) Two unfeasible positions for the links because of wrong angles in the joints.

Figure 5.4: Angles between links and top moving ring.

Wheel angles

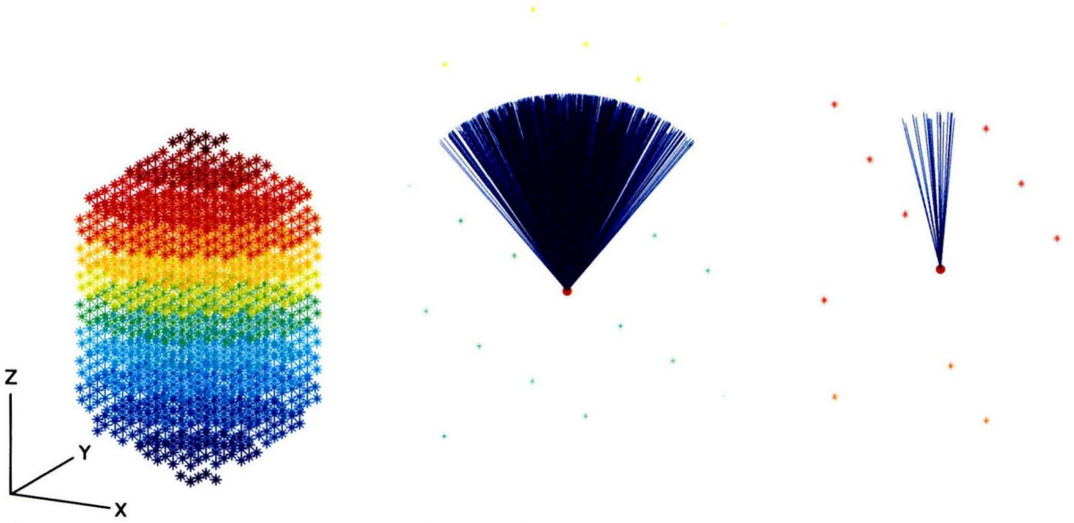
The final test simply classifies a configuration unfeasible when the angles of the motor wheels are too big. When these angles are outside of the range $[-\frac{\pi}{2}, \frac{\pi}{2}]$ the piezo motor would lose contact with the motor wheel and thus this situation is not desired and classified as unfeasible too.

5.2 Results

Using these tests, a configuration can be classified either feasible or unfeasible. All feasible configurations together are defined as the reachable workspace of the platform. To define a finite number of configurations to test, a grid of test configurations is created. The 6D grid is split in two. First a 3D position grid is created. It is known that when no rotations are applied, the platform has a maximum displacement of 16 mm in both x and y direction and a maximum displacement of 24 mm in z direction. With rotations applied, these maximum displacements become smaller, so the grid is set to a maximum size of $[-8, 8]$ mm in x, y direction starting from the normal configuration and $[-12, 12]$ mm in z direction. In each direction the grid is divided in 15 equally spaced points, resulting in a grid with $15 \times 15 \times 15 = 3375$ possible positions in 3D space.

Furthermore, on each position another grid is created defining the rotations to test. Rotations, described as roll, pitch, yaw rotations cannot exceed $\frac{\pi}{4}$ rad, around either the x, y or z axis because of the support structure. Therefore on each point in the position grid, a rotation grid is created with angles of range $[-\frac{\pi}{4}, \frac{\pi}{4}]$ rad and split in 15 equally spaced values about all three axes. Resulting in 3375 rotations to test on each position and thus 11,390,625 possible configurations in total to test.

After running all tests a big set of possible configurations is the result. Since the results are six dimensional, visualizing them is difficult. Therefore the configuration grid is split in a position and rotation grid for visualization. The positional reachable workspace is shown in Figure 5.5a. This is a three dimensional point cloud showing all feasible positions for the platform with zero rotation. For visualization purposes the color of the points is linked to their z value. The plot is created in Matlab and made interactive. Whenever a user selects a point from the cloud, all feasible rotations are indicated by orientation vectors, representing the normal vector to the top face of the moving ring. Figure 5.5b shows such a rotation workspace. Figure 5.5c shows another rotation workspace from a point close to the position workspace boundary. It is clearly visible



(a) Position workspace as point cloud.

(b) A rotation workspace at one point from the position point cloud.

(c) Another rotation workspace from a point close to the position workspace boundary.

Figure 5.5: Reachable workspace split in reachable position and reachable orientation workspace.

that the number of feasible rotations is significantly smaller here.

Now that the reachable workspace is known, one can get a good feeling of the limitations of the platform with the visualization tool as shown in Figure 5.5. Analyzing the results of all the tests also revealed that the support structure is responsible for almost all the unfeasible configurations. There were still some configurations classified unfeasible by other tests, but either support collision tests classified them unfeasible as well, or the top moving ring was so close to the support structure that the support collision tests did not classify the configuration unfeasible due to numerical issues. This is a good results since it verifies that the support structure will limit all movements of the platform and thus the user is not able to directly put loads on the motor mechanisms with the possibility of damaging these. Furthermore another visualization tool is developed in Matlab were the trajectory of a user developed motion is visualized and checked for feasibility. The user can thus tests its trajectories off line. See how they perform and make sure that the path is feasible. One could think of a feasibility check running on line that constantly checks the current path for collisions and re-plan or stop the movement if necessary. However because of the high computational load for this check and the ease of checking the path off line, it is chosen not to implement this on line check.

Chapter 6

Data acquisition and system identification

6.1 Data acquisition

The real time inverse kinematics approach as described in 4.3 is implemented in C on an MBED micro-controller [17]. This micro-controller is connected to a custom developed amplifier board which in its turn powers the motors of the platform. The used micro-controller, the mbed NXP LPC1768, is based on a 32-bit ARM® Cortex^{TM}-M3 processor running at 96 Mhz. The board has a few possibilities for communication to a computer. For testing, analyzing and for system identification it is demanded that one can acquire data from the mirco-controller and more specifically, controller signals in real time. For system identification for example using the indirect three point measurement, three signals have to be acquired for each motor. The control loop runs at 500 Hz and the signals are of an integer data type (4 $bytes$ large). Which means that according to (6.1) every second at least $36,000$ bytes have to be transmitted.

$$3\ values \times 4\ bytes \times 6\ motors \times 500\ Hz = 36000\ bytes/sec \tag{6.1}$$

In every data frame also a header is inserted which requires a total of $38,000$ bytes to be send every second. This requires the transfer speed to be $304,000$ bps. An easy way to communicate to a host computer is by a serial connection. Therefore the highest serial baudrate on the MBED should be selected which is 921600 *baud*. Unfortunately it turned out that the MBED was too slow for this purpose. Solely transferring data at this speed is feasible, but the MBED requires way more processing time for the rest of the program. I.e. the control loop and communication to the amplifier's FPGA. Several setups have been tested, including using the USB port of the MBED and configuring it as a USB-HID device. But they failed due to problems with the rest of the software. In the end the following setup turned out to work. A second MBED is connected using an SPI connection. Setting the SPI registers doesn't take as much processing time as sending serial signals and is therefore much faster. All data is transferred to the second MBED via SPI and the second MBED on its turn sends the data via a high speed serial connection (921600 *baud*) to the host computer. Data collection at the host computer is done by a Matlab program. Matlab can easily communicate with serial devices. The program is constantly filling a big buffer with serial data and if wanted it displays this data in graphs every set timespan as can be seen in Figure 6.1. This figure shows the signals during an identification experiment of motor 0. The columns in this figure correspond to the six motors while the rows indicate the different signals that are being measured. The first row with the red signals shows the error signals for each motor. The second row with the blue signals shows the noise signals that are injected. The last row with the green signals shows the plant input signals. The plant input is defined as the controller output signal plus the injected noise signal.

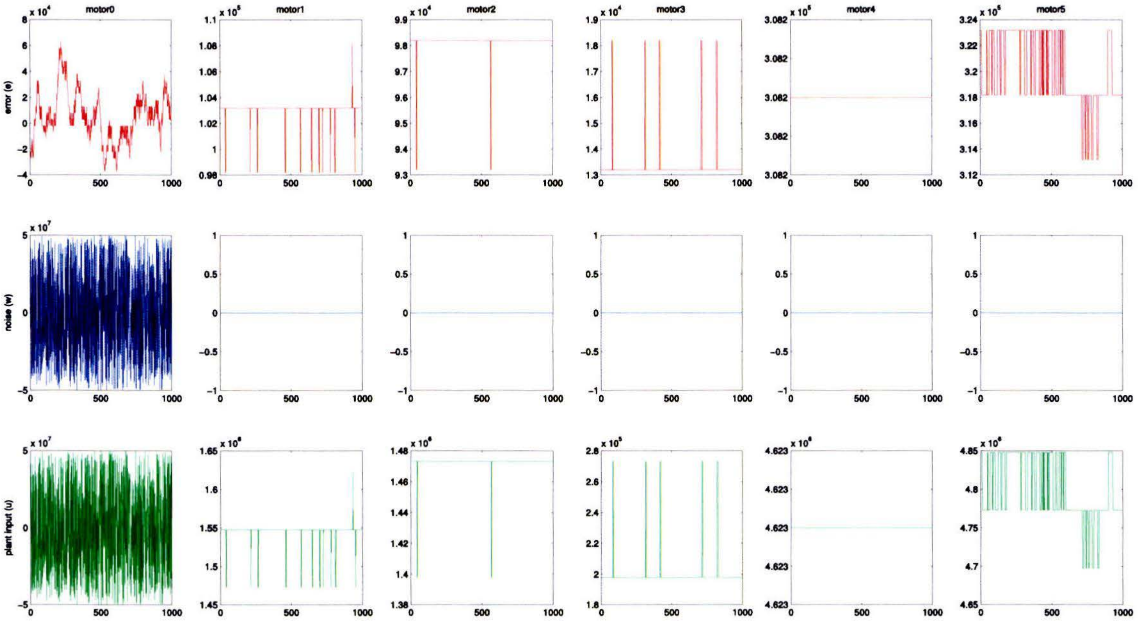


Figure 6.1: Measurement signals during identification experiment of motor 0.

6.2 System identification

The plant to be identified could be considered a 6 × 6 MIMO system. The 6 inputs are then the motor angles and the 6 outputs the position and orientation of frame X_T, Y_T, Z_T. However, it is chosen to handle the system as six individual SISO systems, with the motor input to the wheel motors as input signal and the wheel rotation as output. There are no sensors in the platform to directly measure the position and orientation of frame X_T, Y_T, Z_T. The configuration can be measured indirectly by measuring the wheel angles and applying forward kinematics. However, as stated in section 4.4, such an algorithm is chosen not to derive for this platform. There are several reasons for this. First, as described in section 4.2 the inverse kinematics of a parallel type manipulator are relatively easy to solve. For this compact platform however they are harder and for real time performance an approximation of the inverse kinematics problem is derived. The forward kinematics will be even harder to solve and again need an approximation for real time performance, since these have to be calculated every controller time step. So one could argue how precise the actual configuration can be determined by an approximation of the forward kinematics. Furthermore, the movements the platform needs to make do not need a very high precision. IXI-Play's moves need to look good and smooth, but a little deviation from its trajectory will not or hardly be visible to the user. More important is the amount of noise produced by the motors. For the end user noise is very annoying and should therefore be limited as much as possible. It is therefore of great interest to know what the wheels and motors are doing. Finally the system shows very little interaction. If one wheel is actuated, all other wheels hardly move because the piezo motors on the other wheels have a breaking effect when not actuated. It is therefore more easy and chosen to control six individual SISO systems instead.

One such SISO system consists of a wheel, actuated by a piezo motor. The input to the system is the input to the piezo motor. The piezo motor is actuated by a pwm signal at either ±80 kHz or ±100 kHz. The pwm signal has a maximum duty cycle of 50%. With a maximum duty cycle the elliptical movement of the tip of the piezo motor is maximal. Lowering the duty cycle makes the piezo crystal in the motor "push" the motor less and thus results in smaller elliptical movements. Effectively this results in a lower speed of the wheel. See for example Figure 6.2 for two of these pwm signals. According to Elliptec [3] the motors can be actuated in several ways.

One technique is to always send a maximum duty cycle signal to the motor but switch the signal on and off. Actually this resembles extrapolating a second pwm signal on a larger time scale. This method however is chosen no to use since the larger time scale pwm signal has frequencies that are typically within the audible frequency range and thus produce noise. Another technique is again to always steer the motors with a maximum duty cycle signal, but change the frequency of the pwm signal. As can be seen in Figure 2.4, the speed of the motor will change with different frequencies. However, the relation between pwm frequency and motor speed is not linear[18]. Furthermore steering the motors at different inefficient frequencies introduces unnecessary heat in the motors. It is thus chosen to always keep the optimal pwm frequency but change the duty cycle of this signal accordingly. Although this method is not described by Elliptec, it is believed that this is the optimal control strategy for this system. On the wheels a reflective strip with slots is placed to create an optical encoder and provide position feedback. The optical encoder has a resolution of 0.005 *rad* or 0.29 *deg*. So the input to a SISO system is the duty cycle of the steer signal and the output is the angle of the motor. With this setup six identification measurements are performed. A simple controller is plugged into the system which solely consists of a proportional gain with gain factor 15. Every experiment, all wheels get a reference signal of zero. After the controller a noise signal is injected but only to the motor of interest. Finally the noise signal n, the error signals e and the plant input signals u are measured. The complete configuration for one motor is schematically shown in Figure 6.3. The experiment is repeated for each motor only changing the motor at which the noise signal is injected.

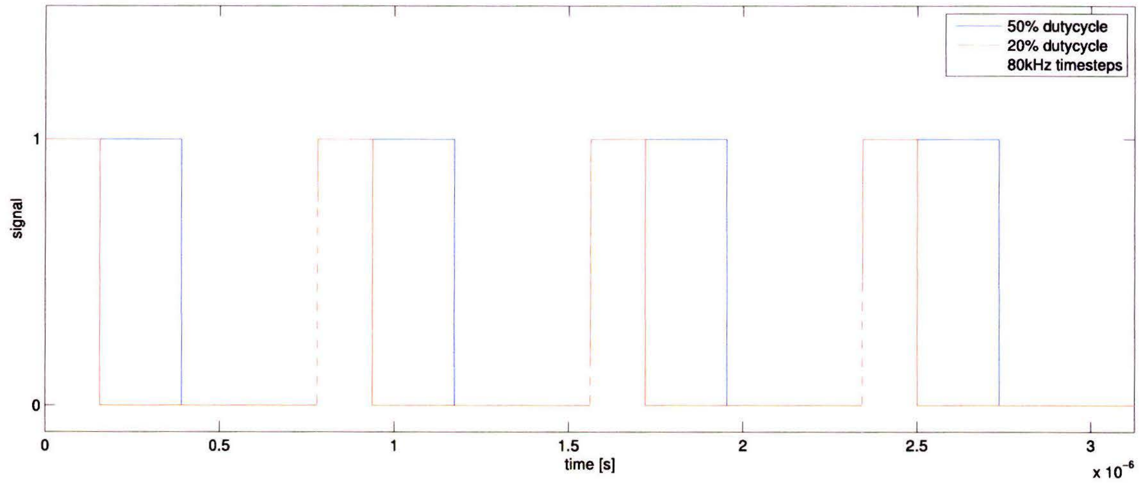


Figure 6.2: PWM signals for the piezo motor.

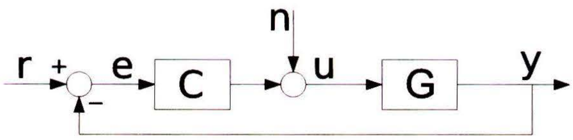


Figure 6.3: Setup for system identification.

Using all signals from all experiments a full MIMO transfer matrix can be identified. This way interactions can be studied. The full MIMO transfer matrix G is given by (6.2), with $S = \frac{u}{d}$ the sensitivity and $PS = -\frac{e}{d}$ the process sensitivity. The transfer matrix G of size 6×6 describes the transfer of all inputs u_i, $i = 1...6$ to each output y_i, $i = 1...6$. For each element of G the transfer functions S and PS have to be determined using Matlab's tfestimate() function. A Hanning window is applied to the data with a frame size of 1024 points and 512 points overlap. With the sample frequency of 500 Hz this results in a frequency resolution of $\frac{500}{1024} \approx 0.5\ Hz$.

$$Y = GU, \quad with \quad G = S^{-1}PS \tag{6.2}$$

The resulting transfer matrices are included in Appendix C for clarity. As can be seen from the transfer matrix in Appendix C.3, the interaction is very limited. This can be visualized even clearer by means of the Relative Gain Array (RGA) matrix. The RGA is defined as (6.3), with $\circ$ being the Hadamard product, or element wise multiplication.

$$RGA(G) = \Lambda(G) \triangleq G \circ (G^{-1})^T \tag{6.3}$$

The RGA can be evaluated at each frequency and the result is shown in Appendix C.4. Whenever the diagonal terms of the RGA are equal to one, the MIMO system is properly decoupled. As can be seen this is mostly the case. Apparently at $15 - 20\ Hz$ there is a little interference between the motors and furthermore there is some interaction between motors 4 and 5 (The last two entries in the matrix, since we start counting at zero). This most probably has to do with the current design of IXI-Play's head. In the front of it's head a lot of electronics are mounted like the camera and oled-displays. This makes the head heavy in the front and this results in a pulling force on motors 4 and 5 in the back of the platform. Therefore they show more interaction, but still the diagonal terms are dominant here. With this results it can be concluded that the system has indeed very little interaction and thus it should be feasible to consider it as six SISO systems to control. Figure 6.4 shows the bode plots of all six motors together in one figure. One can see that there is some variation in the frequency responses of the individual motors. So during control design one should keep this in mind and take care of the robustness margins. The frequency response makes one think of a mass-spring-damper system with near critical damping. At least up to 10 Hz the frequency responses resemble such a system.

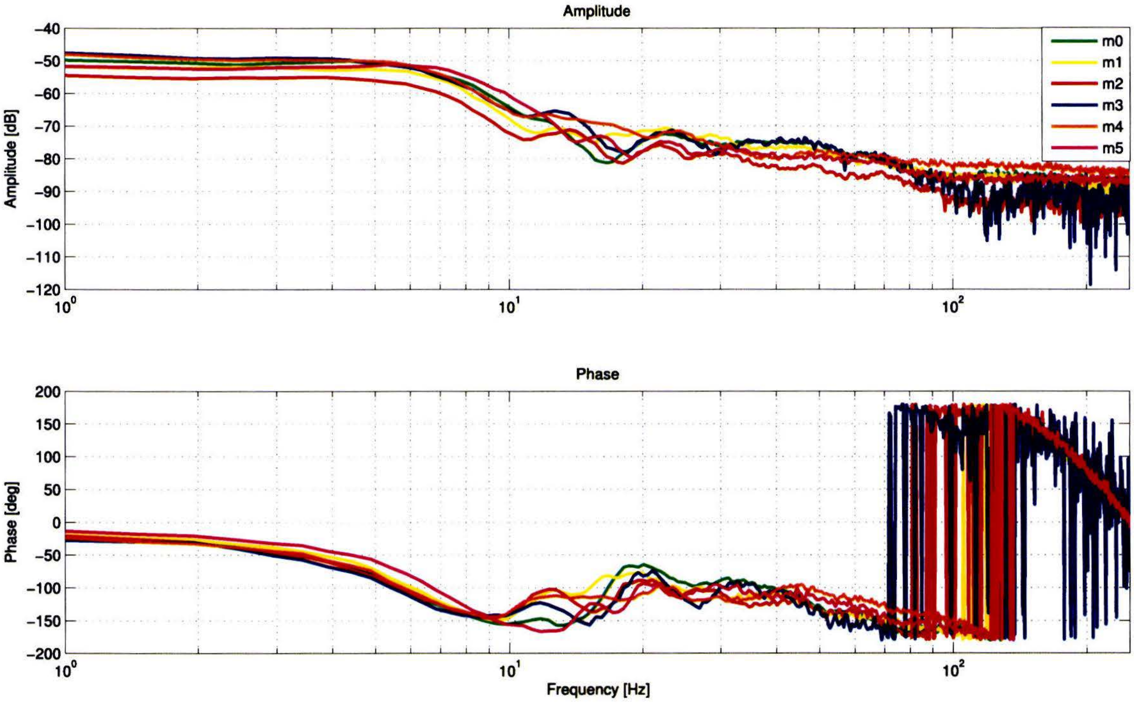


Figure 6.4: The six frequency responses of the six motors in one bode plot.

Chapter 7

Feedback controller and feedforward design

7.1 Feedback controller design

After system identification a controller will be designed. A single controller with sufficient robustness margins will be designed to use for each motor. A modulus margin of $MM < 6\ dB$, phase margin $PM > 30\ deg$ and gain margin $GM > 10\ dB$. The controller design will be performed on a nominal system. The nominal system is considered to be the mean of the six identified systems shown in Figure 6.4. Figure 7.1 shows these frequency responses again with the nominal plant plotted over it. The nominal plant is used during loopshaping to design a controller. First to filter

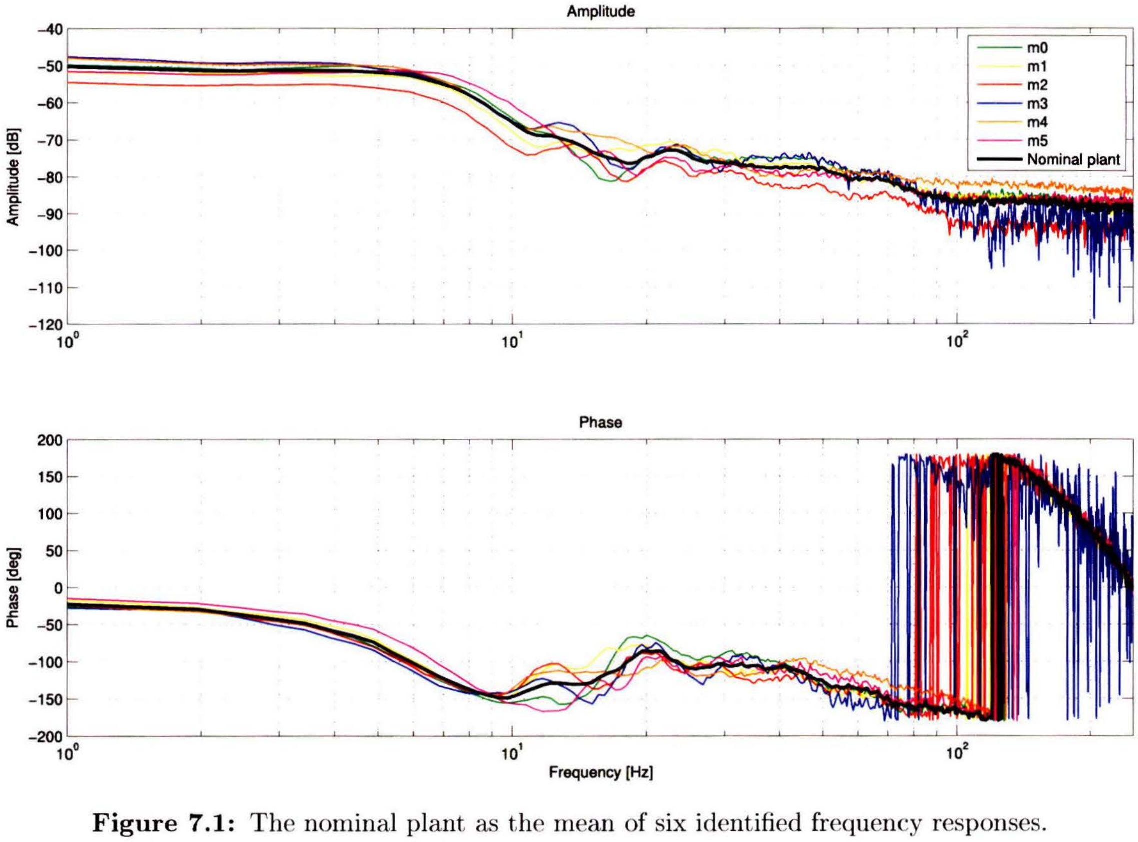


Figure 7.1: The nominal plant as the mean of six identified frequency responses.

out the high frequency noise, a first order low pass filter is introduced. The pole of the low pass filter is placed at 30 Hz. The gain is increased to a factor of 650 and correspondingly a lead filter is added to gain phase around the cross over frequency of 10 Hz with a zero at $\frac{10}{3}$ Hz and its pole at $10 * 3$ Hz. The transfer function of the controller is therefore equal to:

$$C(s) = \frac{31.04s + 650}{2.814e5s^2 + 0.01061s + 1} \tag{7.1}$$

Discretization of the controller using Tustin's method with a sample frequency of 500 Hz leads to:

$$C_d(z) = \frac{797z^2 + 32.7z - 764.3}{z^2 - 1.366z + 0.4662} \tag{7.2}$$

The frequency response of the discretized controller together with the frequency response of the continuous time controller are shown in Figure 7.2. As can be seen the discrete time controller is a good approximation up to a frequency of 100 Hz. For higher frequencies the discrete time controller shows significantly more roll-off. The phase loss at higher frequencies is limited however, so the discretization will only cause more suppression of high frequency noise. With the controller applied, the continuous time open loop frequency responses are shown in Figure 7.3. As can be observed, the achieved bandwidth is around 10 Hz. Figure 7.4 shows the Nyquist plots of the continuous time open loop systems. As can be observed all systems are stable with sufficient margins.

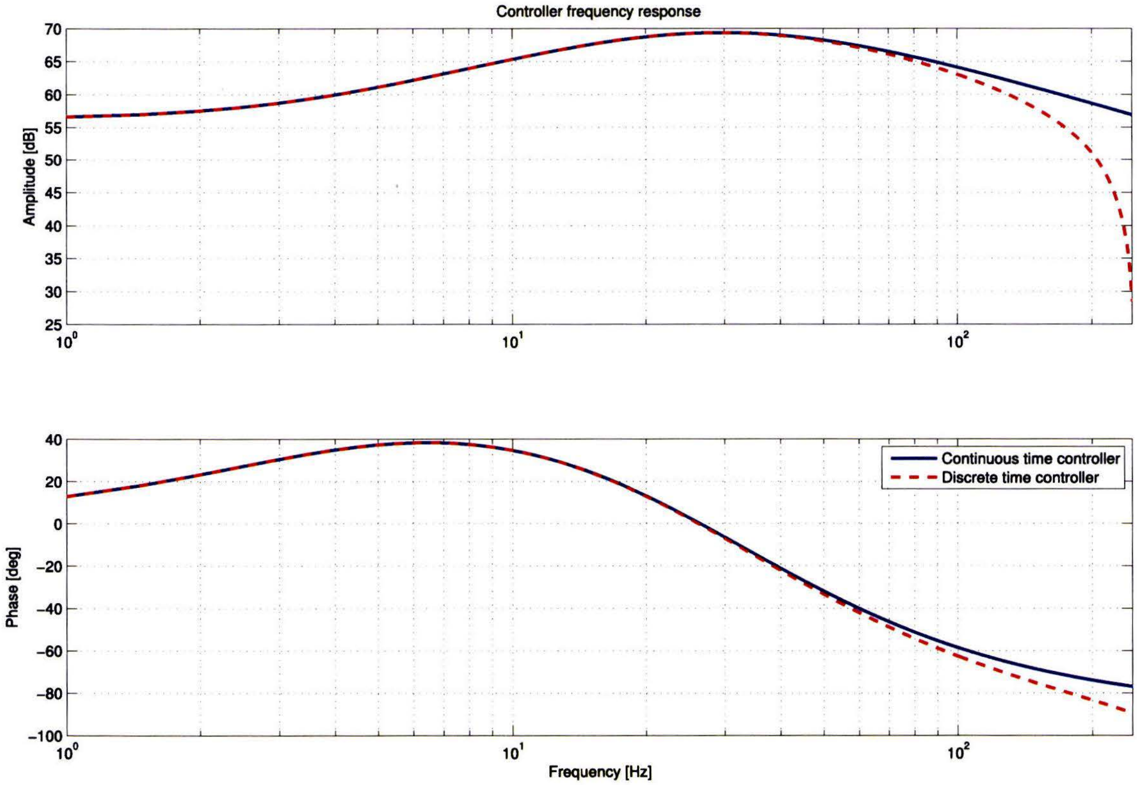


Figure 7.2: Controller frequency response in continuous time and discrete time.

7.2 Controller implementation

The designed controller as introduced in section 7.1 is implemented in the software on the MBED micro controller in C code. Since the piezo motors have a maximum duty cycle of 50 %, the

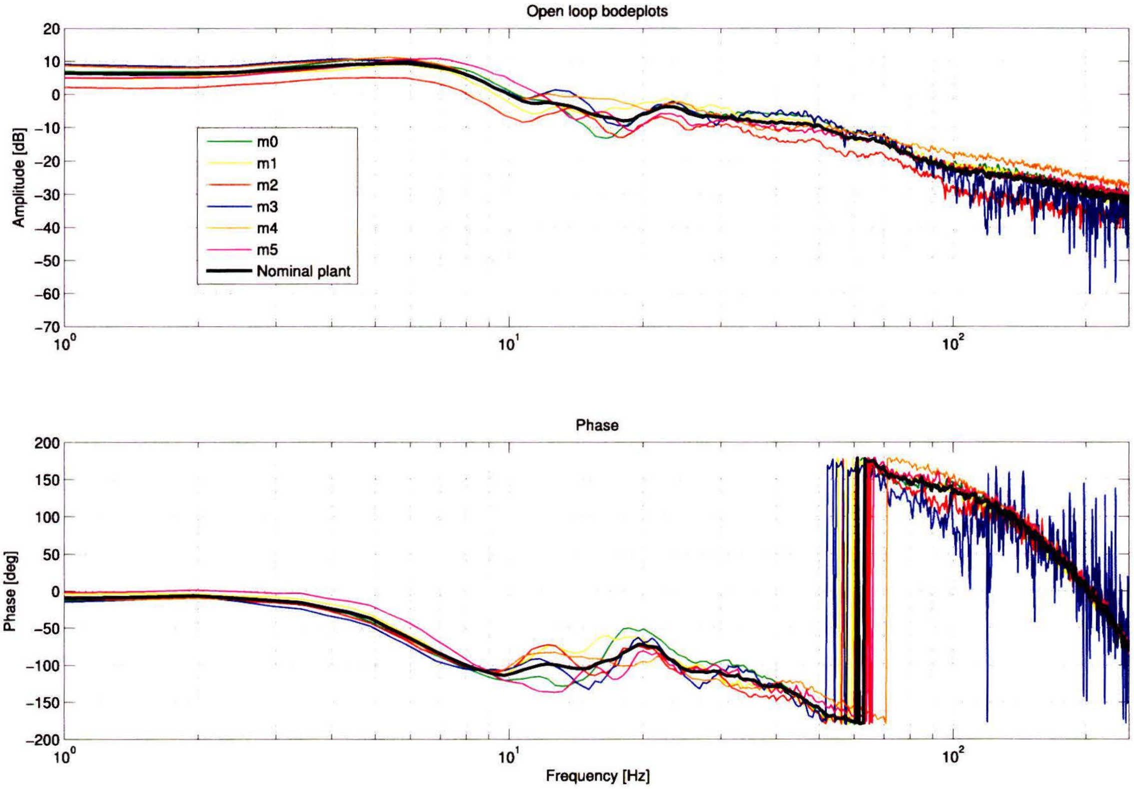


Figure 7.3: Open loop bodeplots of all controlled systems.

controller output is saturated in software at this maximum. With the controllers implemented in software some test were run. Figure 7.5 shows the step response of the system. The system is positioned out of its normal configuration and suddenly the normal configuration is set as a reference for the controllers. This corresponds to a reference of 0 *deg* for all motors. As can be seen from the figure, the system reacts quick without overshoot. Although a static error remains. The maximum error here is 2.5 *deg*. Which corresponds to 9 encoder counts with the current encoder resolution. It could be considered to add an integrator to the controller to improve low frequent behavior and minimize the steady state error. However it is not chosen to do so. From visual feedback it is noticed that the 2.5 *deg* error is not noticeable at all to the human eye looking at the movements of the robot. Furthermore an extra integrator makes some movements look weird to the spectator. Some movements could build up a slight drift from the reference trajectory and the integrator makes the movement suddenly jump to the reference trajectory because of the integrator's windup. For the spectator this looks weird and thus a smooth movement which slightly deviates from the reference trajectory is more pleasant to the eye. Finally the sudden jumps that occur due to the integrator control action can introduce extra noise which is again unpleasant to the spectator. Therefore the current feedback control scheme is considered satisfactory.

However, definitely a remark should be noted here. Especially with more complex or fast movements it appeared that the controllers saturated rather quickly. I.e. the maximum duty cycle of 50% was applied often. This results in worse tracking behavior and more noise since the piezo motor should actually supply more power but is slipping instead. One should therefore be critical in designing complex motion profiles. To improve this behavior one could consider to chose more powerful actuators or change the wheel diameter. The latter results in more torque but a reduction of speed. Another consideration should be to lower the mass of the head as much as possible by choosing lighter electronics.

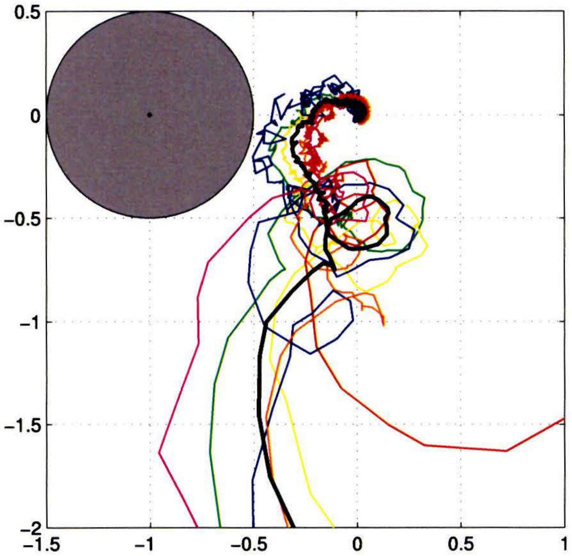


Figure 7.4: Nyquist plots of all controlled systems.

7.3 Feedforward design

To increase controller performance, a feedforward signal can be added to the control scheme. Especially for movements with high accelerations, a feedforward signal is a pleasant addition, since it already steers the system in the right direction before a feedback error could build up. Furthermore, a higher controller gain makes the system follow the reference trajectory more closely, but the controller output is a result of the error signal. Because of the limited encoder resolution, the error will have an evident staircase like profile. A controller with high feedback gain will therefore introduce a control signal with an amplified staircase profile and thus introduce noise again to the system. With a smooth feedforward signal however, this extra noise is not introduced, while the feedback error can be reduced when tuned properly. In general a feedforward signal is implemented as shown in Figure 7.6. This is the ideal case, where the feedforward signal is constructed by passing the reference signal r through the inverse of the plant G^{-1}. The exact transfer G^{-1} however is difficult to derive and most probably unstable. It is therefore chosen to derive a stable estimation of G^{-1}. Starting with the nominal plant as shown in Figure 7.1, which is simply a system of type FRD in Matlab. Which stands for Frequency Response Data and is a vector of complex numbers, representing the frequency responses evaluated at a defined vector of frequencies. Taking the inverse of these complex numbers, one arrives at an FRD which describes the frequency response of the inverse system. As noticed before in 6.2, the identified systems resemble a mass-spring-damper system with near critical damping. Such a system is a second order system. Therefore it is first tried to fit a second order stable system to this inverse frequency response. The result is shown in Figure 7.7 as the red dashed line. The result is not really satisfactory and therefore the same fit is tried again using a third order approximation. This result is shown in Figure 7.7 also as the green dashed line and is way more pleasing. Therefore the third order fit is chosen to be used instead. Up to a frequency of 10 Hz, which is also the bandwidth of the feedback system, the fit resembles the inverse system well. The order of the fit is chosen to be as low as possible to keep the processing time on the micro controller to a minimum. The transfer function of the inverse fit is equal to:

$$G^{-1}(s) \approx \frac{2830s^3 + 8.091e04s^2 + 3.739e06s + 1.369e07}{s^3 + 31.03s^2 + 7941s + 6.291e04} \tag{7.3}$$

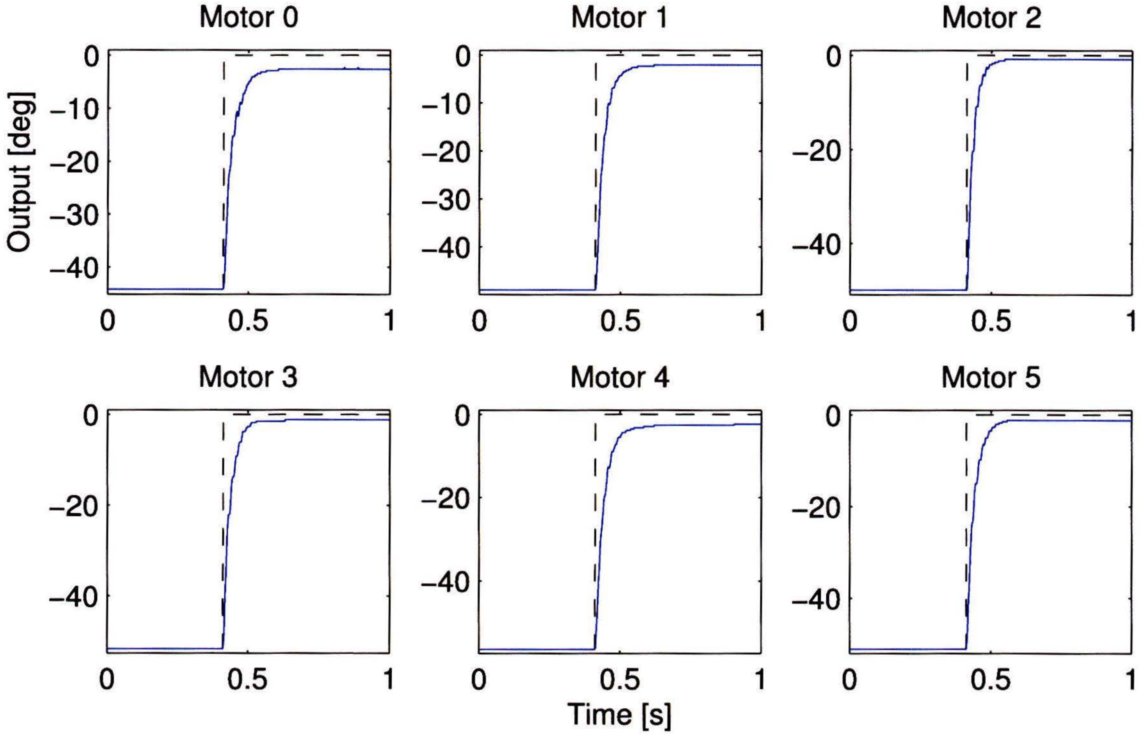


Figure 7.5: Step response of the IXI-Play platform.

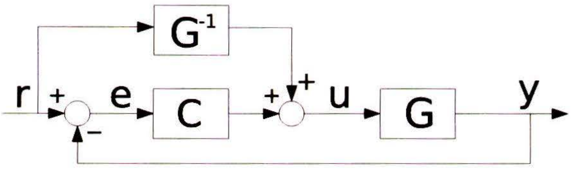


Figure 7.6: General feedforward control scheme.

Discretized using Tustin's method gives:

$$G_d^{-1}(z) \approx \frac{2805z^3 - 8245z^2 + 8089z - 2649}{z^3 - 2.909z^2 + 2.85z - 0.9401} \tag{7.4}$$

The obtained feedforward model is implemented in the micro controller software as well. To test its performance a trajectory is created. The head should follow a circle shaped path on the xy-plane at a frequency of 1 Hz. The trajectories for the motors therefore result in sinusoid signals. Figure 7.8a shows this trajectory and the plant output without the feedforward signal applied. Figure 7.8b shows the same trajectory but with the feedforward signal enabled. It can be seen that the tracking performance is increased, especially at points where the direction of movement changes. Another possible improvement would be to model the effect of the flexible body and the inertia of the head. The effect of these two factors on the system are almost unknown but are expected to be of great importance. Modeling these goes beyond the scope of this thesis but would definitely be a complement to the current control scheme.

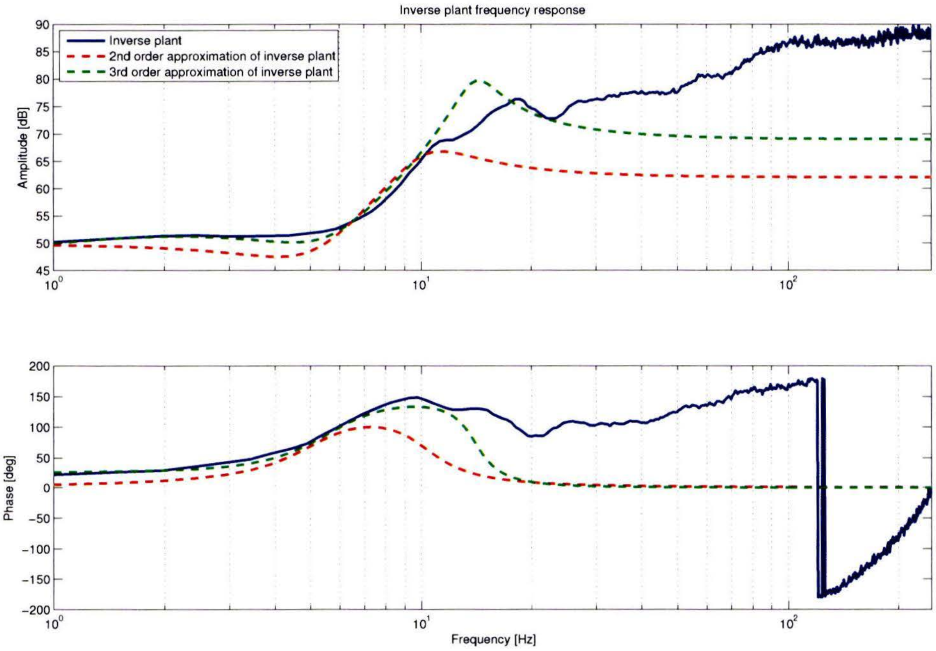
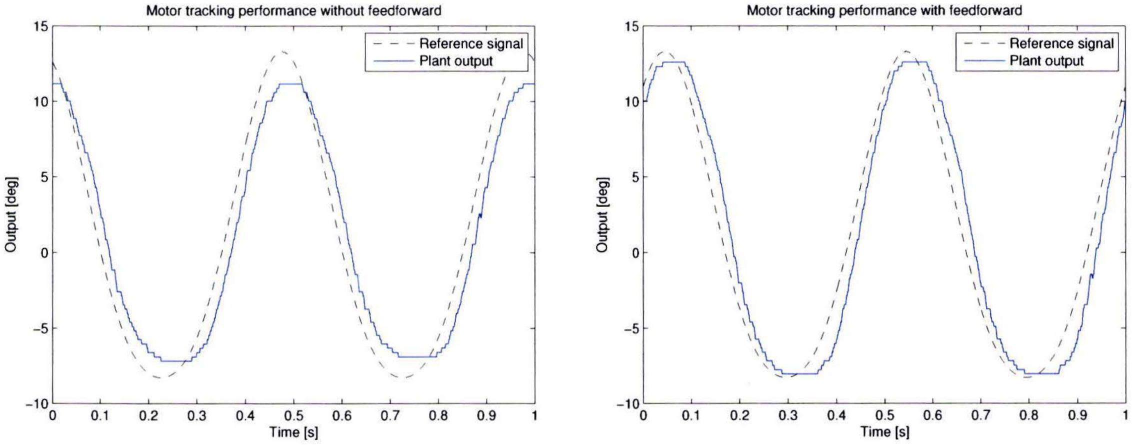


Figure 7.7: Second and third order fits of the inverse system.



(a) Tracking performance without feeforward.

(b) Tracking performance with feeforward applied.

Figure 7.8: Tracking performance with and without feedforward.

Chapter 8

Increasing the optical encoder resolution

As noticed in previous sections, the resolution of the optical encoders on the wheels is limited to 0.005 *rad* or 0.29 *deg*. The encoder quantization is actually significantly present in measurements as can be seen in Figures 7.8 and 7.5. Higher resolution encoders are more expensive which is not desired for this platform, since it should sell as a consumer product in the end. For control purposes however, a higher encoder resolution is desired. Especially for the reduction of noise this higher resolution is desired as described in section 7.3. A possible solution therefore is to artificially increase the encoder resolution by means of software. Increasing the encoder resolution by means of time stamping is such a method [19, 20]. The basic idea is to store a series of so called encoder-events and fit a low order polynomial function through these points. With the fitted polynomial function the encoder value at a current time step can be predicted by extrapolating the found polynomial function.

8.1 Algorithm

The time stamping method first stores a series of encoder events. An encoder event is the pair (t_k, x_k) with t_k being the time of an encoder pulse transition and x_k the corresponding position value. The index k indicates the encoder event number. Figure 8.1 shows the time stamping concept graphically. One transition is shown and so is the stored encoder event. The number of

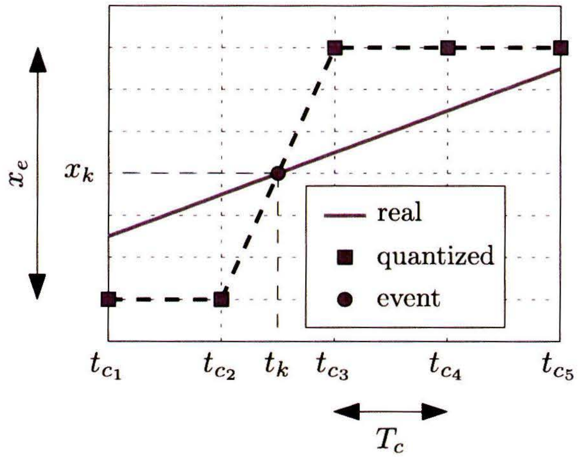


Figure 8.1: Timestamping concept.

stored encoder events n is tunable and depends on the setup. After capturing n encoder events, a low order polynomial function is fitted through the encoder events by means of the least squares method. For m, the order of the polynomial fit, at least $n \geq m$ encoder events have to be occurred. Let $p_0, ..., p_m$ be the polynomial coefficients, $t_1, ..., t_n$ the encoder event time stamps and $x_1, ..., x_n$ the encoder event position values. Next define matrix A and vectors P, B as:

$$A = \begin{bmatrix} t_1^m & t_1^{m-1} & \cdots & 1 \\ t_2^m & t_2^{m-1} & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ t_n^m & t_n^{m-1} & \cdots & 1 \end{bmatrix}, \quad P = \begin{bmatrix} p_m \\ p_{m-1} \\ \vdots \\ p_0 \end{bmatrix}, \quad B = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \tag{8.1}$$

For $n = m$ an exact fit is made through the encoder events, for $n > m$ the over-determined system of linear equations to be solved equals:

$$AP = B \tag{8.2}$$

The least squares method is then defined as:

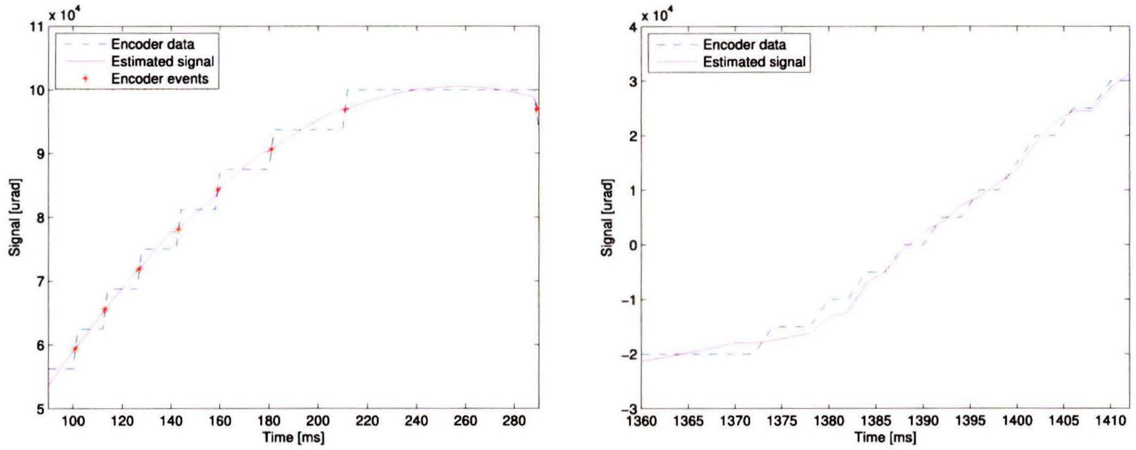
$$A^T AP = A^T B \tag{8.3}$$

As suggested by [19], LU-factorization without pivoting is used to solve this system in real time. When solved, the found polynomial function is used to predict a position value by means of extrapolation. The estimated position $\tilde{x_e}$ at time t_e is then calculated by:

$$\tilde{x_e} = p_m t_e^m + p_{m-1} t_e^{m-1} + ... + p_0 \tag{8.4}$$

8.2 Results

The algorithm is first tested off line. A 2 Hz sinusoid signal is quantized and tried to estimate using the time stamping method. It is chosen to fit a 2^{nd} order polynomial through the last 5 encoder events. The result is shown in Figure 8.2a. The algorithm is implemented in the software as well. The result of the experiment on the setup is shown in Figure 8.2b. Clearly the signal is still not smooth all the way, however it is much more fluent compared to the quantized encoder signal. The maximum error of this time stamping method is equal to the maximum quantization error. Furthermore, since the estimation is extrapolated, there is no lag introduced by the method. This makes it a very satisfactory method to use for artificially increasing the encoder resolution. In terms of noise, especially with slow moving trajectories where the staircase profile of the quantized signals is more obvious, it is desirable to include the time stamping method since it significantly reduces noise. A problem arises however since the calculation time on the micro controller sometimes exceeds the 2 ms with all software described so far. This means that the control loop will not have a fixed sampling time of 500 Hz and the timing of the system is uncertain. The MBED, the currently used micro controller is especially chosen for its ease to program. The final design of IXI-Play will certainly have another faster micro controller embedded. But while implementing, one should be careful and make sure that the micro controller is fast enough and the processing time never exceeds 2 ms.



(a) Simulation with a quantized sinusoid and estimated signal.

(b) Experiment data with quantized signal and estimated signal.

Figure 8.2: Results of increasing encoder resolution with time stamping method.

Chapter 9

Conclusions and future work

This thesis is aimed at the motion control of the IXI-Play platform. Conclusions and recommendations are given in this chapter.

9.1 Conclusions

For this project it was asked to develop:

- A real-time inverse kinematics algorithm
- Identify the reachable workspace of the platform
- Design control algorithms that let the robot move smoothly and as quiet as possible

Chapter 4 explains how the inverse kinematics of the IXI-Play platform are derived and how an approximation to this inverse kinematics is developed for real-time performance. After validation the approximation proved to be useful since it is fast and only has small errors. The positional errors of the platform are smaller then 1 mm in 95% of the cases while the rotational error is smaller than 1.4 deg in 95% of the cases. These little deviations are hardly visible to the eye of the spectator and acceptable for the IXI-Play platform. The inverse kinematics approximation is fast enough to run real-time in the 500 Hz control loop.

The reachable workspace identification is treated in chapter 5 together with the results. It is verified that the supporting structure responsible for the unfeasible configurations which means that the platform is robust, since the user cannot directly put loads on the actuator systems. Furthermore a visualization is made to show the reachable workspace and a simulator is created to simulate and verify the feasibility of motions off line.

In chapter 7 the design of the feedback controller and feedforward is shown. First the system was identified and with this information a controller and feedforward was designed. The tracking behavior of the system proved to be sufficient and noise was kept to a minimum. In chapter 8 a method to increase the optical encoder resolution by means of software is adopted to even further reduce noise.

9.2 Future work

As stated in section 7.2, the actuators saturate often when complex or fast motion profiles are given to the system. This should definitely be a point of attention for future work. Several options can be investigated. For example changing the used actuators or the wheels to deliver more power with the eventual loss of actuator speed. Furthermore keeping the mass of the head and especially the electronics in the head to a minimum is of great importance.

Another interesting research project is the development of models that describe the influence of inertia of the moving head and the force delivered by the flexible body to the system. The latter could both help or counteract movements of the platform but could possibly be of great importance in the control strategy of the platform.

A last remark is about the timing on the micro controller. The MBED, the micro controller currently used because of its ease to program has too little resources for all software developed. Since the MBED will not be the micro controller of choice in the final IXI-Play platform, attention should be payed to the computational power of the micro controller.

Chapter 10

Bibliography

[1] WittyWorx, "Wittyworx the art of technology." [Online]. Available: http://www.wittyworx.com/

[2] S. Koekebakker, "Model based control of a flight simulator motion system," Ph.D. dissertation, TU Delft, December 2001.

[3] Elliptec, "Elliptec resonant actuator ag." [Online]. Available: http://www.elliptec.com/

[4] Elliptec Resonant Actuator AG, "X15g datasheet," 2012.

[5] H. Adriaens, W. L. De Koning, and R. Banning, "Modeling piezoelectric actuators," Mechatronics, IEEE/ASME Transactions on, vol. 5, no. 4, pp. 331–341, 2000.

[6] Mathworks, "Multibody simulation - simmechanics." [Online]. Available: http://www.mathworks.com/products/simmechanics/

[7] M. Tarokh, "Real time forward kinematics solutions for general stewart platforms," IEEE International Conference on Robotics and Automation, pp. 901–906, 2007.

[8] Y. Cao, W. Ji, Z. Li, H. Zhou, and M. Liu, "Orientation-singularity and nonsingular orientation-workspace analyses of the stewart-gough platform using unit quaternion representation," Chinese Control and Decision Conference, pp. 2282–2287, 2010.

[9] Z. Bingul and O. Karahan, Dynamic Modeling and Simulation of Stewart Platform, 2012.

[10] C. C. Nguyen, S. S. Antrazi, J. Park, and Z.-L. Zhou, "Trajectory planning and control of a stewart platform-based end-effector with passive compliance for part assembly," Journal of Intelligent and Robotic Systems, vol. 6, no. 2-3, pp. 263–281, 1992.

[11] J.-P. Merlet, "Solving the forward kinematics of a gough-type parallel manipulator with interval analysis," The International Journal of robotics research, vol. 23, no. 3, pp. 221–235, 2004.

[12] K. Liu, J. M. Fitzgerald, and F. L. Lewis, "Kinematic analysis of a stewart platform manipulator," Industrial Electronics, IEEE Transactions on, vol. 40, no. 2, pp. 282–293, 1993.

[13] J.-P. Merlet, "A formal-numerical approach for robust in-workspace singularity detection," Robotics, IEEE Transactions on, vol. 23, no. 3, pp. 393–402, 2007.

[14] M. Majid, Z. Huang, and Y. Yao, "Workspace analysis of a six-degrees of freedom, three-prismatic-prismatic-spheric-revolute parallel manipulator," The International Journal of Advanced Manufacturing Technology, vol. 16, no. 6, pp. 441–449, 2000.

[15] I. A. Bonev and J. Ryu, "A new approach to orientation workspace analysis of 6-dof parallel manipulators," Mechanism and Machine Theory, vol. 36, no. 1, pp. 15–28, 2001.

[16] J. D'Errico, "Mathworks file exchange: Inhull." [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/10226-inhull

[17] A. Holdings, "Mbed, rapid prototyping for microcontrollers." [Online]. Available: https://mbed.org/handbook/mbed-NXP-LPC1768

[18] F. Gharibnezhad and A. Shahri, "Elliptec piezo electric motor: Modeling and control using fuzzy approaches," in Mechatronics and Its Applications, 2008. ISMA 2008. 5th International Symposium on. IEEE, 2008, pp. 1–9.

[19] R. Merry, M. van de Molengraft, and M. Steinbuch, "Optimal higher-order encoder time-stamping," Mechatronics, 2013.

[20] R. Merry, R. van de Molengraft, and M. Steinbuch, "Error modeling and improved position estimation for optical incremental encoders by means of time stamping," in American Control Conference, 2007. ACC'07. IEEE, 2007, pp. 3570–3575.

Appendix A

Elliptec datasheet

Elliptec
RESONANT ACTUATOR

X15G Datasheet

General Description

This small, inexpensive piezoelectric motor has been designed for precision and responsiveness. It has a broad, variable speed range and can operate almost noise-free.

Features

- High resolution: < 10 µm
- Ultra light: 1.2 g
- Highly responsive: 0 to v_0 in 100 µs
- Rugged environment compatible
- Vacuum compatible

Motor Characteristics

(Under common enviromental conditions, TA = 25 C unless otherwise noted; for more details see motor manual.)

Symbol	Parameter	Ratings			Units	Notes
		min.	typ.	max.		
v_0	No-Load Speed	300	350	550	mm/s	Measured with standard driver electronics
a	Acceleration to Max. Speed with Mass of 1.5g		5		ms	Depends on accelerated mass, here: 1.5g (see fig. 1)
F_H	Unpowered Holding Force	0.5	0.8	1.2	N	Slider or rotor material IXEF 1032 / PF 7595
F_B	Maximum Blocking Force	200	300	500	mN	Slider or rotor material IXEF 1032 / PF 7595
F_{100}	Max. Operating Force @ 100mm/s	100	200	350	mN	Slider or rotor material IXEF 1032 / PF 7595
F_{200}	Max. Operating Force @ 200mm/s	25	100	200	mN	Slider or rotor material IXEF 1032 / PF 7595
M_B	Blocking Torque @ 20mm Wheel	2	3	5	mNm	Depends on wheel diameter, here: 20mm
M_{100}	Operating Torque @ 20mm Wheel @ 95.5rpm	1	2	3.5	mNm	Depends on wheel diameter, here: 20mm
M_{200}	Operating Torque @ 20mm Wheel @ 191rpm	0.25	1	2	mNm	Depends on wheel diameter, here: 20mm
$f_{Forward}$	Forward Frequency @ v_{max}	77	81	84	kHz	Temperature dependent, see f_D
$f_{Backward}$	Backward Frequency @ v_{max}	93	98	108	kHz	Temperature dependent, see f_D
f_r	Frequency Resolution		0.2	0.6	kHz	Higher frequency resolution allows better performance (min. req.: 8-Bit timer@16MHz)
f_D	Temperature Dependent Frequency Drift	35	50	70	Hz/ C	Lower temperatures result in higher frequencies and vice versa
rs	Resolution	var.	5-100		µm	Minimum resolution depends on speed and position sensor resolution
P_{pd}	Power of Driver Electronics (at full speed)		1.8		W	Measured with standard electronics
I_{Driver}	Current to Driver Electronics	300	450	600	mA	Measured at 5V and v_{max} (max. current decreases linearly with decreasing driving speed)
U_{motor}	Peak Voltage at Motor	5	7	10	V	Measured as peak-to-peak voltage of sinus wave shaped driving signal
I_{motor}	Peak Current			1.5	A	Measured with standard driver electronics (see fig. 2)
L_{IXEF}	Lifetime with slider or rotor material IXEF 1032	5			km	Motor wear limits total stroke, not total operation time
L_{PF7595}	Lifetime with slider or rotor material PF 7595	40			km	Motor wear limits total stroke, not total operation time

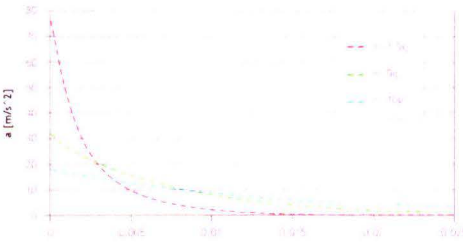


Figure 1: Acceleration vs. time

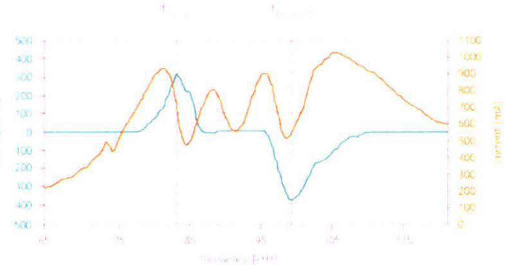
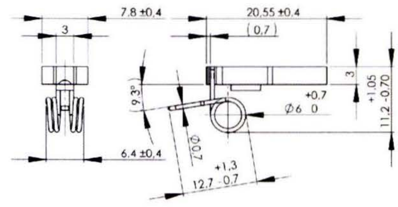


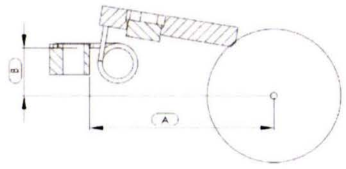
Figure 2: Typical speed vs. frequency v(f)

Physical Dimensions/Mounting Instructions (mm)

To avoid interfering with motor vibrations, no secondary parts may touch the motor.

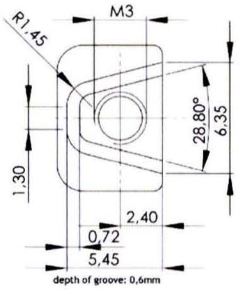


Side View
(regular spring)

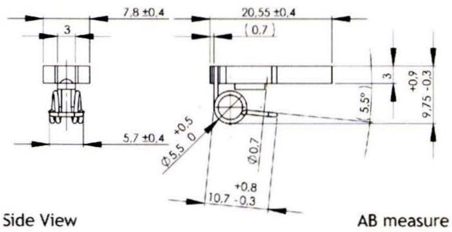


AB measure
(regular spring)

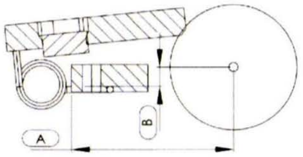
Wheel diameter	A	B
(mm)	(mm)	(mm)
10	24,49	3,37
14	25,77	4,90
20	27,70	7,20



Mounting Block
(regular spring)

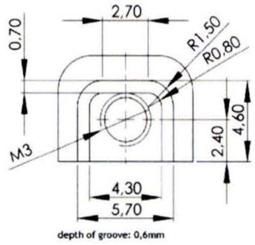


Side View
(reverse spring)

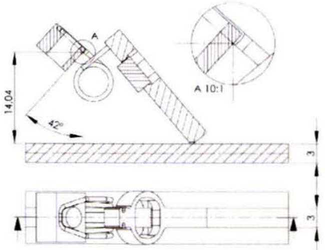


AB measure
(reverse spring)

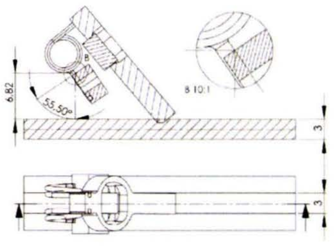
Wheel diameter	A	B
(mm)	(mm)	(mm)
10	16,85	3,23
14	18,50	2,10
20	20,97	0,40



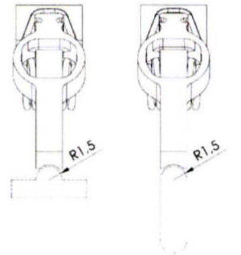
Mounting Block
(reverse spring)



Linear Setup
(regular spring)



Linear Setup
(reverse spring)



Recommended cross-sections (surface contours) of driven elements

Order information

Article Number	Spring	Cable length	Connector Plug
X15G	regular	50 mm	none
X15G-F2	reverse	50 mm	none
X-15G-K99-ST99	regular	120 mm	JST 02ZR-8M

Recommended maximum cable length 120 mm, further lengths, other springs and connector plugs on request.

For further information and design ideas please refer to the motor manual or visit our website at www.elliptec.com

Subject to change without notice.

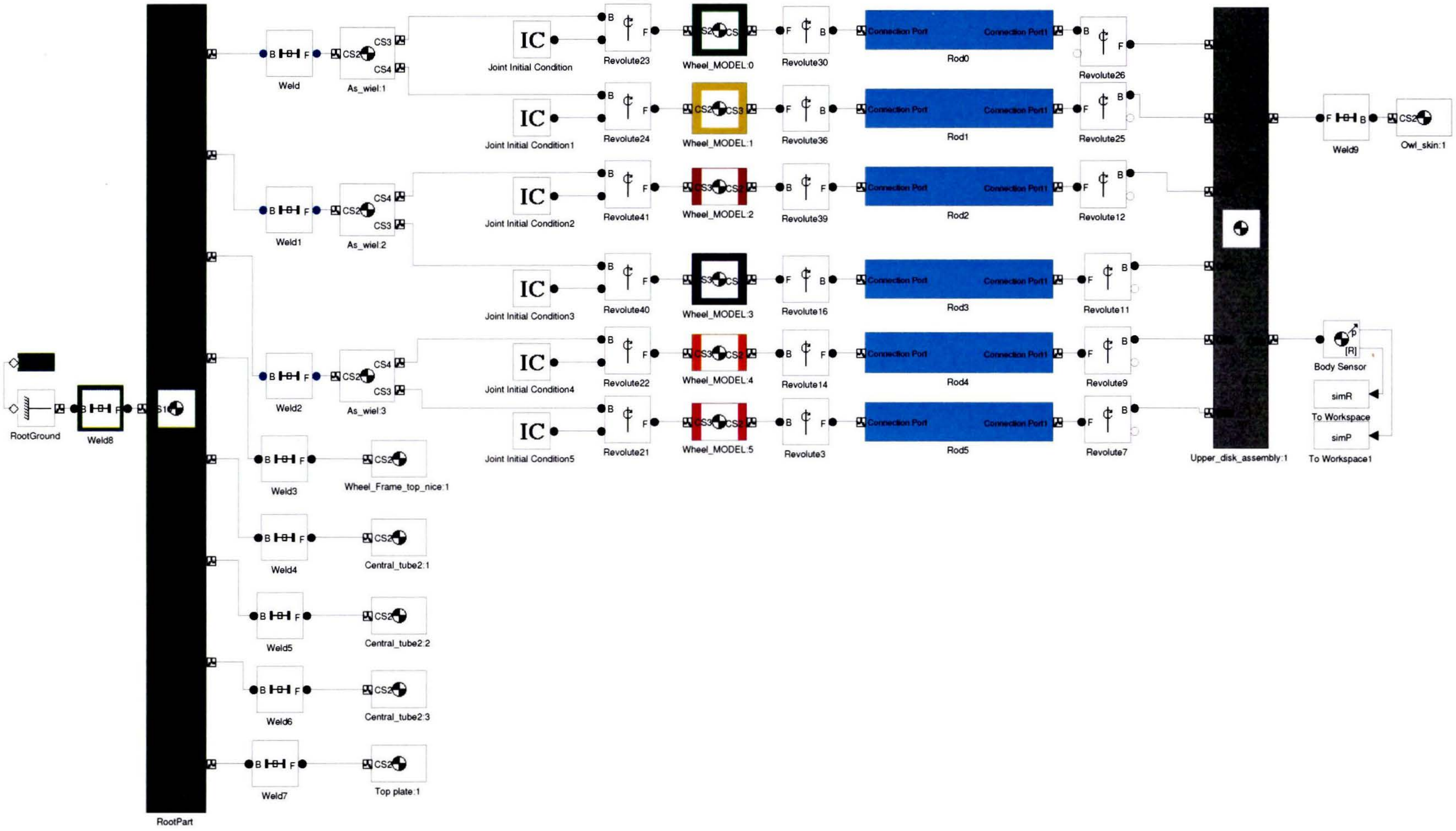
Elliptec Resonant Actuator AG
Meinhardstrasse 3
44379 Dortmund
Germany

Tel +49 (0) 2 31 29 27 02 0
Fax +49 (0) 2 31 29 27 02 50

info@elliptec.com

Appendix B

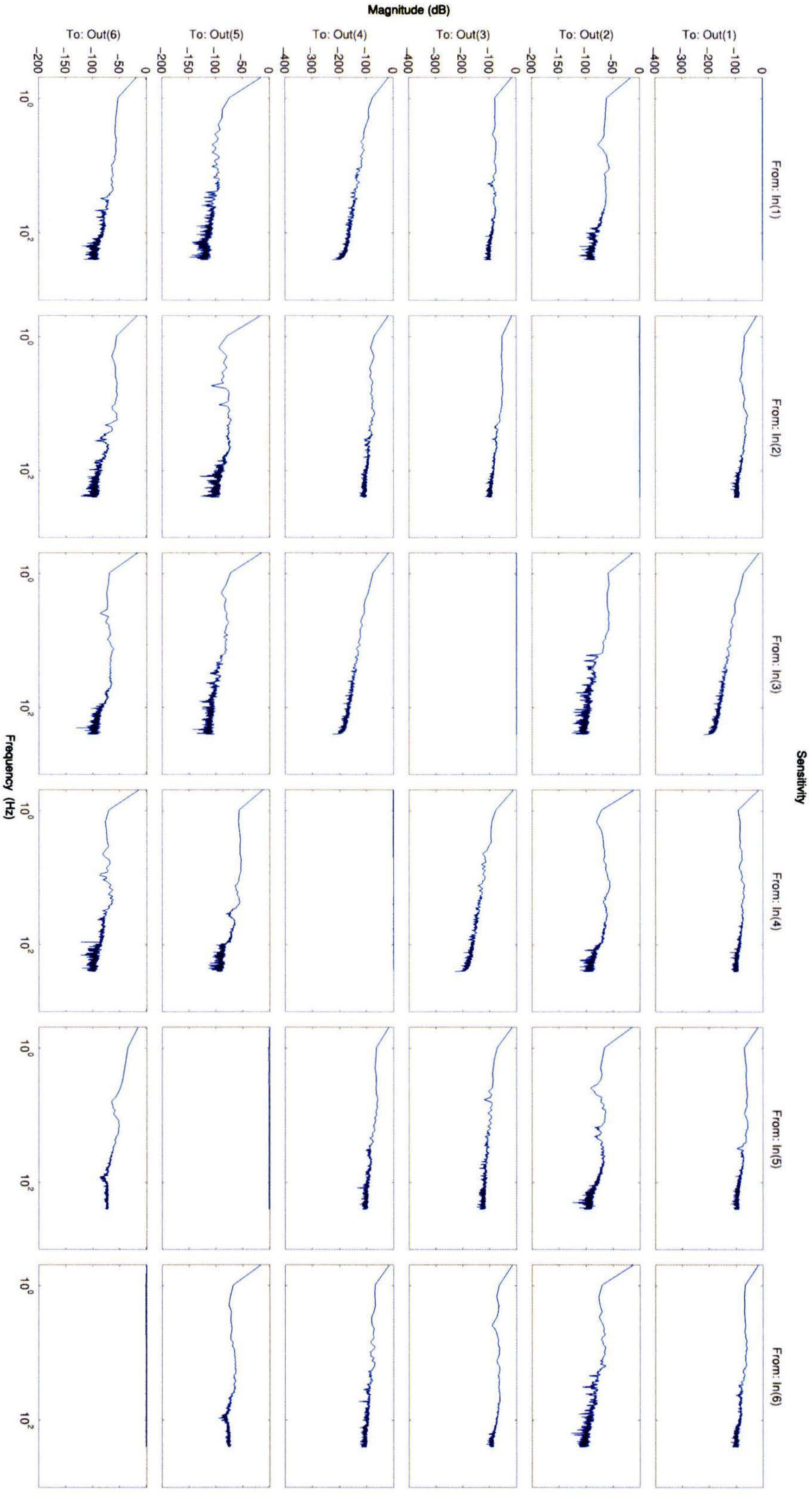
SimMechanics model


RootGround
Weld8
RootPart
Weld
As_wiel:1
Weld1
As_wiel:2
Weld2
As_wiel:3
Weld3
Wheel_Frame_top_nice:1
Weld4
Central_tube2:1
Weld5
Central_tube2:2
Weld6
Central_tube2:3
Weld7
Top plate:1
Joint Initial Condition
Joint Initial Condition1
Joint Initial Condition2
Joint Initial Condition3
Joint Initial Condition4
Joint Initial Condition5
Revolute23
Revolute24
Revolute41
Revolute40
Revolute22
Revolute21
Wheel_MODEL:0
Wheel_MODEL:1
Wheel_MODEL:2
Wheel_MODEL:3
Wheel_MODEL:4
Wheel_MODEL:5
Revolute30
Revolute36
Revolute39
Revolute16
Revolute14
Revolute3
Connection Port
Connection Port1
Rod0
Rod1
Rod2
Rod3
Rod4
Rod5
Revolute26
Revolute25
Revolute12
Revolute11
Revolute9
Revolute7
Upper_disk_assembly:1
Weld9
Owl_skin:1
Body Sensor
simR
To Workspace
simP
To Workspace1

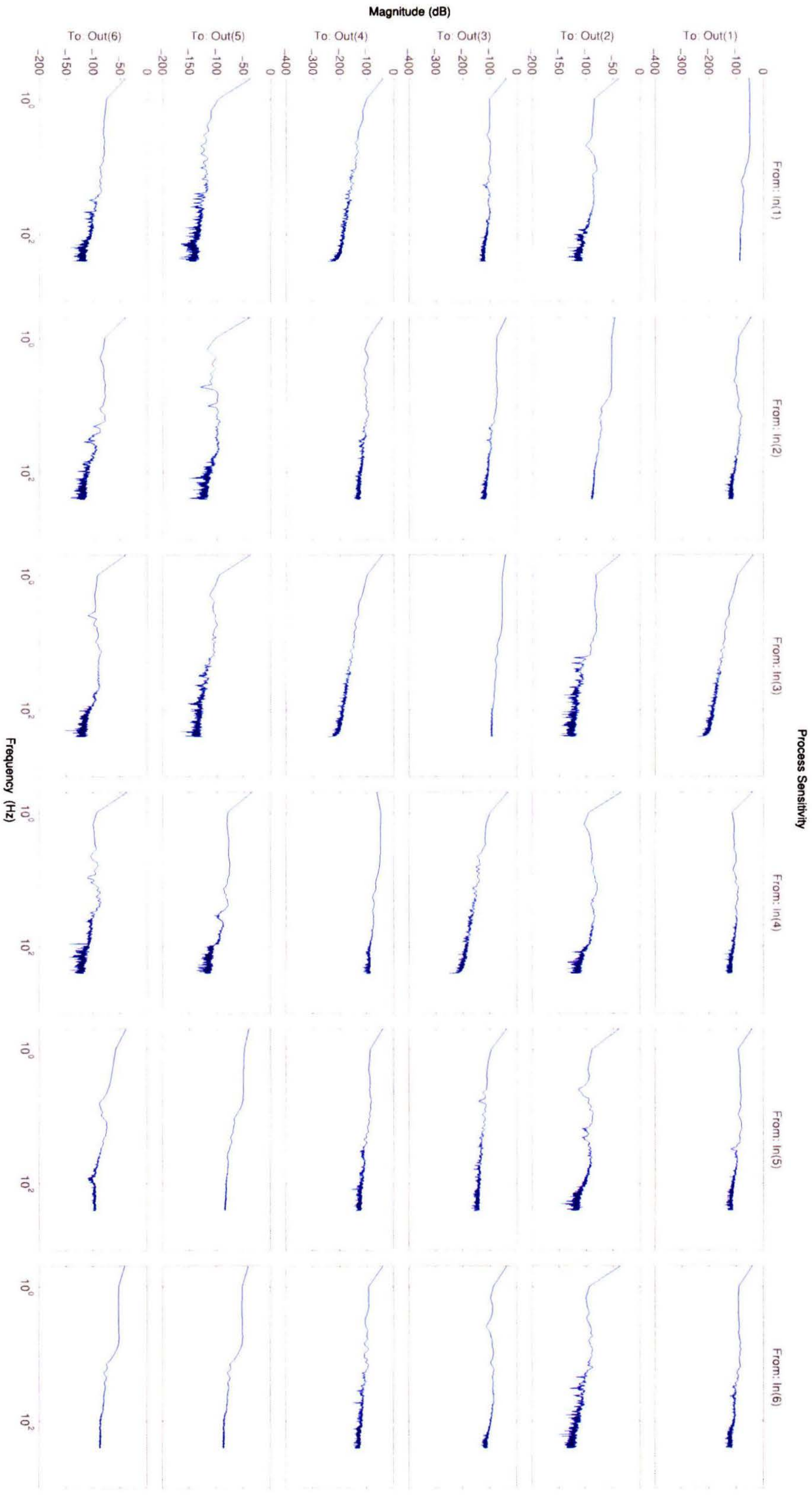
Appendix C

System identification

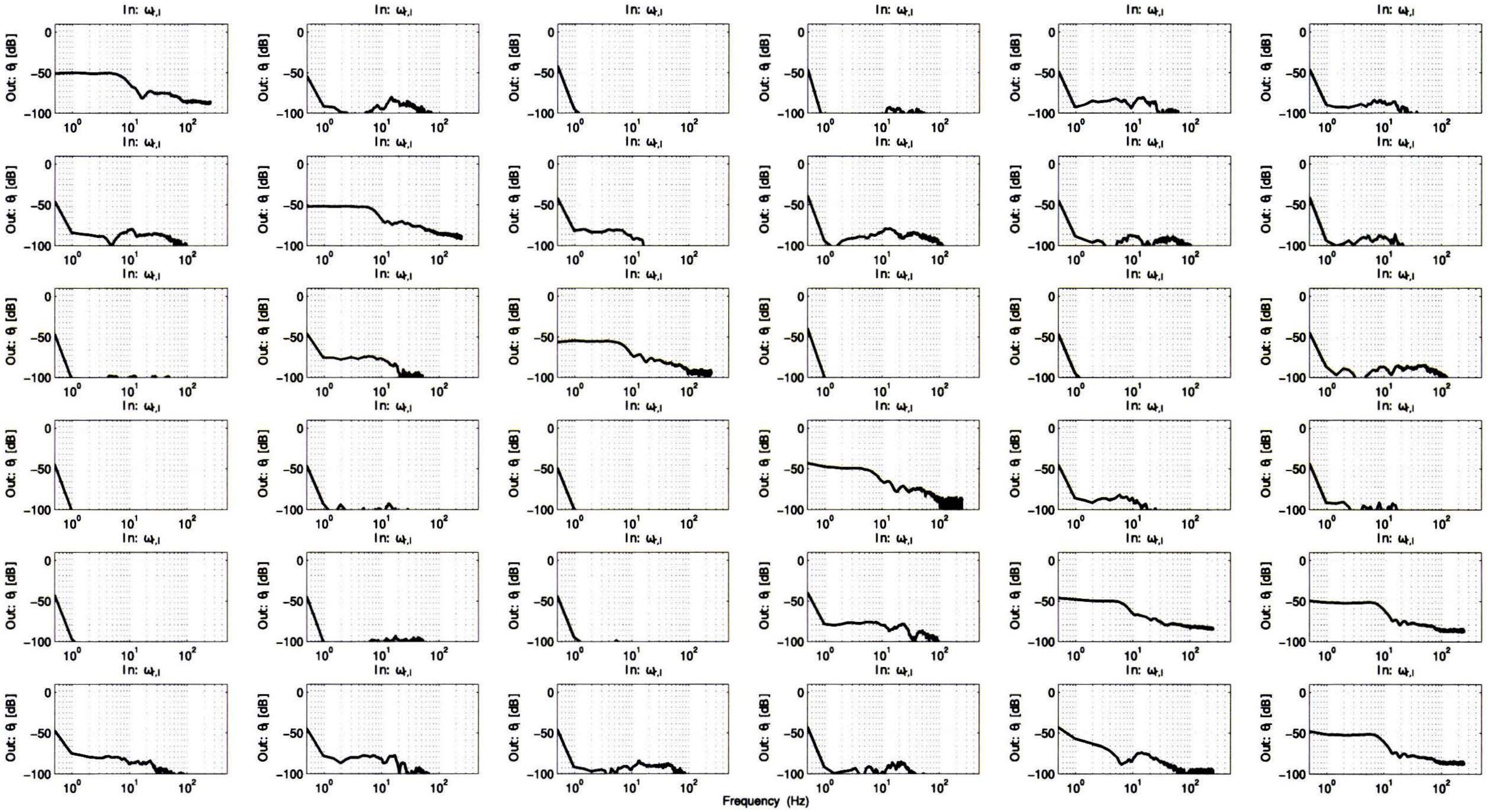
C.1 Sensitivity matrix



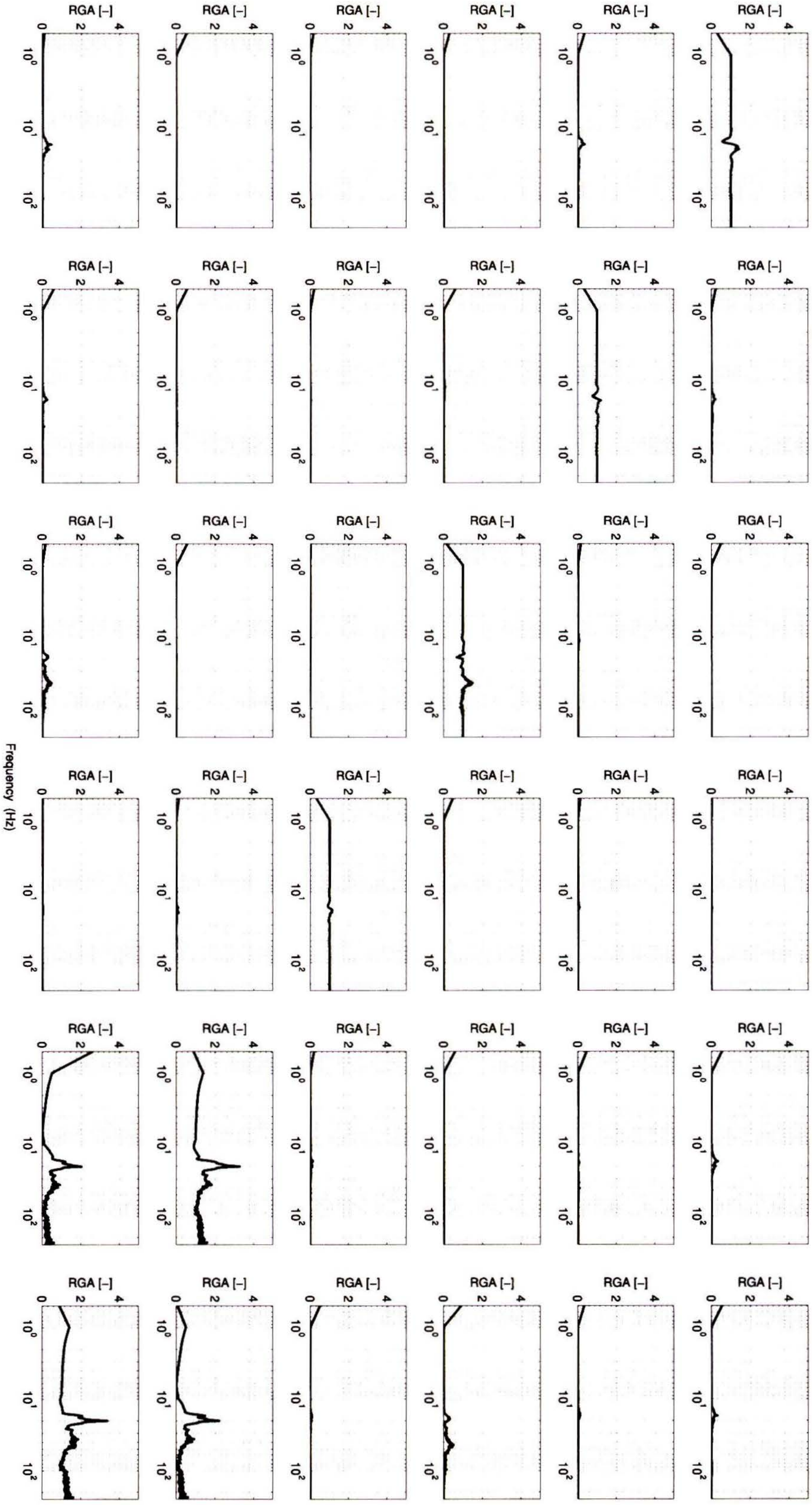
C.2 Process sensitivity matrix



C.3 Transfer matrix



C.4 Relative gain array


RGA [–]
10^0
10^1
10^2
Frequency (Hz)