Direct paper-motion control using an optical mouse-sensor

M.P.M. Hendriks
s456643
DCT 2003.53

Coach:
dr.ir. S. Koekebakker

Supervisor
Prof.dr. H.Nijmeijer
Section Dynamics and Control Technology
Faculty Mechanical Engineering
University of Technology Eindhoven

Eindhoven, June 2003

"The writer of this report was given the opportunity by Océ Technologies B.V. to perform his internship research, on which this report is founded. Océ Technologies B.V. does not accept responsibility for the correctness of the data, considerations and conclusions in this report, which are fully on the account of the writer."
Summary
In Océ-machines it is very important a sheet of paper, which has to be printed on, is delivered at the right time at the right place. This so-called registrating is often performed by units, which stop paper transportation for some moments of time. Some years ago, an on-the-fly-registration module has been developed, which can simultaneously move, align rotate and shift a piece of paper. This enables positioning during transportation in order to get higher productivity and better controllability. In this registration unit, the movement of paper is indirectly encoded through rotation of motor-axes. Microsoft has developed a cheap optical PC-mouse sensor, which is able to measure displacements.
The registration unit consists of two independently driven motors. Therefore transport (x-direction) and rotation (ϕ-direction) can be actuated. However it is impossible to move the paper sideward (z-direction). Analysing the system in chapter two learns z-movement is possible: Successive rotation transport back-rotation and back transport results in z-displacement of the paper. In chapter three it is proved that the system is fully controllable and that, besides the PC-mouse, a second sensor is needed to reconstruct all three variables (x, ϕ and z). To control the non-linear system a non-linear control law is needed. However, in this case a linear control law will do: In copiers and printers a paper sheet is always imported at constant velocity. Both feed forward and feedback are used to control in x and ϕ-direction, only feedback is used in z-direction. Simulations show errors for all three variables will converge to zero.
During this traineeship an experimental set-up is built in order to check theoretical analysis. In chapter four this system is identified. The same control strategy is used as the one in the simulations and in chapter six this is checked. Due to lack of time control in z-direction is not implemented here. However, satisfying results are obtained: Experiments with three different settings are done. x-transport controlled first. After that, ϕ-rotation is controlled and finally composed actuations in x- and ϕ-direction are controlled in order to get z-displacement. In all these experiments the error will be less than 200 μm, whereas the sensor resolution is 63,5 μm. Although the sensor should be able to detect up to 0.3 m/s the system goes instable at velocities larger than 0.24 cm/s. This is caused by both paper-bulging (required distance between sensor and paper is not obtained) and electrical charging of the sensor. Because the error is quite small it can be said the PC-mouse sensor works quite well in a control-loop. During the experiments the initial conditions are not determined by the sensors but manually inserted. The developed algorithm, which should solve this, should be integrated in a follow-on traineeship as well as z-control. Furthermore a non-linear control law could be developed for such a non-linear system. Finally solutions for the paper-bulging and electrically charging should be found.
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>INTRODUCTION</td>
<td>5</td>
</tr>
<tr>
<td>2.</td>
<td>KINEMATICS</td>
<td>6</td>
</tr>
<tr>
<td>2.1</td>
<td>DRIVERS-MECHANISM</td>
<td>6</td>
</tr>
<tr>
<td>2.2</td>
<td>SENSORS</td>
<td>7</td>
</tr>
<tr>
<td>2.2.1</td>
<td>IntelliMouse</td>
<td>7</td>
</tr>
<tr>
<td>2.2.2</td>
<td>Edge-detection sensor</td>
<td>9</td>
</tr>
<tr>
<td>2.3</td>
<td>PMDC-MOTORS</td>
<td>10</td>
</tr>
<tr>
<td>3.</td>
<td>CONTROL DESIGN</td>
<td>11</td>
</tr>
<tr>
<td>3.1</td>
<td>CONTROLLABILITY</td>
<td>11</td>
</tr>
<tr>
<td>3.2</td>
<td>OBSERVABILITY</td>
<td>12</td>
</tr>
<tr>
<td>3.3</td>
<td>CONTROL</td>
<td>13</td>
</tr>
<tr>
<td>3.3.1</td>
<td>Decoupling</td>
<td>13</td>
</tr>
<tr>
<td>3.3.2</td>
<td>Control of $z_c$</td>
<td>14</td>
</tr>
<tr>
<td>3.4</td>
<td>RECONSTRUCTION OF $X_{eq}$, $\Phi$ AND $Z_c$</td>
<td>16</td>
</tr>
<tr>
<td>3.5</td>
<td>TRAJECTORY</td>
<td>16</td>
</tr>
<tr>
<td>3.6</td>
<td>SIMULATION</td>
<td>19</td>
</tr>
<tr>
<td>4.</td>
<td>SYSTEM IDENTIFICATION</td>
<td>21</td>
</tr>
<tr>
<td>4.1</td>
<td>ORIENTATION PARTS</td>
<td>21</td>
</tr>
<tr>
<td>4.2</td>
<td>DETERMINATION OF THE COULOMB FRICTION</td>
<td>21</td>
</tr>
<tr>
<td>4.3</td>
<td>TRANSFER FUNCTIONS OF PMDC-MOTORS</td>
<td>24</td>
</tr>
<tr>
<td>4.4</td>
<td>CALIBRATION EDGE-DETECTION SENSOR</td>
<td>29</td>
</tr>
<tr>
<td>5.</td>
<td>CONTROLLING THE REGISTRATION UNIT</td>
<td>30</td>
</tr>
<tr>
<td>5.1</td>
<td>CONTROLLING X-TRANSPORT</td>
<td>30</td>
</tr>
<tr>
<td>5.2</td>
<td>CONTROLLING $\Phi$-ROTATION</td>
<td>32</td>
</tr>
<tr>
<td>5.3</td>
<td>S-CURVE</td>
<td>34</td>
</tr>
<tr>
<td>6.</td>
<td>CONCLUSIONS</td>
<td>36</td>
</tr>
<tr>
<td>7.</td>
<td>RECOMMENDATIONS</td>
<td>37</td>
</tr>
<tr>
<td>REFERENCES</td>
<td></td>
<td>38</td>
</tr>
<tr>
<td>APPENDIX 2.1</td>
<td>OBSERVATIONS EDGE-DETECTION SENSOR</td>
<td>39</td>
</tr>
<tr>
<td>APPENDIX 3.1</td>
<td>CONTROLLABILITY</td>
<td>41</td>
</tr>
<tr>
<td>APPENDIX 3.2</td>
<td>OBSERVABILITY</td>
<td>42</td>
</tr>
<tr>
<td>APPENDIX 3.3</td>
<td>DETERMINING INITIAL CONDITIONS</td>
<td>43</td>
</tr>
<tr>
<td>APPENDIX 3.4</td>
<td>CONTROL OF PMDC-MOTOR</td>
<td>45</td>
</tr>
<tr>
<td>APPENDIX 3.5</td>
<td>BLOCK DIAGRAM OF CONTROLLED SYSTEM</td>
<td>48</td>
</tr>
<tr>
<td>APPENDIX 3.6</td>
<td>SIMULATION PIECE OF PAPER</td>
<td>49</td>
</tr>
<tr>
<td>APPENDIX 3.7</td>
<td>SIMULATION SYSTEM</td>
<td>53</td>
</tr>
</tbody>
</table>
APPENDIX 4.1 ENTIRE UNIT ............................................................... 54
APPENDIX 4.2 CHOICE REGISTRATION UNIT ................................. 58
APPENDIX 4.3 TUE-DACS ............................................................... 62
APPENDIX 4.4 DATASHEET PMDC-MOTOR ......................................... 63
APPENDIX 5.1 DETERMINING θ ...................................................... 64
APPENDIX 6.1 SUITABLE CONTROL LAW MOTOR .............................. 65
APPENDIX 6.2 SIMULINK MODEL TO CONTROL THE UNIT .......... 67
1. Introduction

In Océ-machines it is very important a sheet of paper, which has to be printed on, is delivered at the right time at the right place. This so-called registrating is often performed by units, which stop paper transportation for some moments of time. Some years ago, an on-the-fly-registration module has been developed, which can simultaneously move, align rotate and shift a piece of paper. This enables positioning during transportation in order to get higher productivity and better controllability. In this registration unit, the movement of paper is indirectly encoded through rotation of motor-axes. Microsoft has developed a cheap optical PC-mouse sensor, which is able to measure displacements.

During my second traineeship of the Technical University Eindhoven I have been working at GR1 (Group Research) of Océ-Technologies B.V. for sixteen weeks. The object of this traineeship is to investigate whether the sensors of the IntelliMouse can be used for the positioning of a piece of paper.

Several questions have to be answered to pursue the objective: Where should the IntelliMouse be placed with respect to the registration unit? Is the system fully controllable? If so, which control law has to be integrated to achieve this? Will this sensor guarantee a fully observable system or should other sensors be included? Which performance can be obtained with such a system?

In Chapter 2 of this report the registration unit is mathematically analysed. By determining the controllability and observability of the system both control law and reconstruction method are designed in Chapter 3. The controlled mathematical model is implemented and simulated in the program Simulink, which is shown in Chapter 3. To check whether the theoretical analysis represents the reality a registration unit is built. The specific parts of the unit are mentioned in Chapter 4. Some experiments have to be done to identify the developed unit, which is shown in Chapter 5. In Chapter 6 results of controlling the unit are presented and discussed. Finally, in Chapters 7 and 8, respectively conclusions are drawn and recommendations are made.
2. Kinematics

To check the properties and abilities of a dynamical system it is useful to model it first. In this chapter the parts of the registration unit are all analysed mathematically: A paper sheet is moved by two pinches, which are driven by PMDC-motors. In paragraph 2.1 the possible movements of a paper sheet dependent on the paper velocities beneath the pinches are described. Sensors can detect these movements. In paragraph 2.2, the observations of the sensors used, dependent on the paper movements, are derived. Finally, in paragraph 2.3 the differential equations of the motors used are given.

2.1 Drivers-mechanism

In the on-the-fly registration unit the paper is transported and rotated by pinches $P_1$ and $P_2$, which can be driven independently by PMDC-motors $M_1$ and $M_2$ respectively (Figure 2.1).

Equal paper velocities at the points beneath the pinches will cause paper transport in $x$-direction, different velocities will lead to rotation $\phi$. The axes of $M_1$ and $M_2$ have to be aligned so the centre of rotation will be on the line through the motor axes. Besides, the velocities, at the points beneath the pinches, have to be perpendicular to this line. When this is not achieved the paper will bulge or stretch out while moving. For convenience this fictitious line is chosen parallel to the $z$-axis. The mechanism can be reduced to the simple model of Figure 2.2, in which $\dot{k}_1$ and $\dot{k}_2$ represent the paper-velocities in $x$-direction beneath pinch $P_1$ and $P_2$ respectively.

Figure 2.1 Drivers-mechanism

Figure 2.2 Schematic model of the drivers-mechanism
An arbitrary fixed point $P$ on the paper can move in three degrees of freedom with respect to the global co-ordinate system. Dependent on $\dot{k}_1$ and $\dot{k}_2$ every paper point will rotate around point $S$ with angular velocity $\dot{\phi}$.

$$\dot{\phi}(t) = \frac{\dot{k}_2(t) - \dot{k}_1(t)}{L} \tag{2.1}$$

A difference between $\dot{k}_1$ and $\dot{k}_2$ will result in $\dot{\phi}$ according to (2.1). If $\dot{k}_1$ is equal to $\dot{k}_2$ point $S$ will be at infinite distance. So, equations have to be derived independent on $S$.

The following equation can now be formulated for the tangential velocity $v_p$ of point $P$:

$$v_p = \dot{\phi}r \tag{2.2}$$

Now the components $\dot{x}_p$ and $\dot{z}_p$ can easily be determined:

$$\dot{x}_p = v_p \cos(\beta) = b \dot{\phi} \tag{2.3}$$
$$\dot{z}_p = -v_p \sin(\beta) = -h \dot{\phi} = -x_p \dot{\phi} \tag{2.4}$$
$$\dot{\phi} = \dot{k}_1 + z_p \dot{\phi} \tag{2.5}$$

(2.3) is rewritten into (2.5) to avoid distance $b$ and make (2.5) independent on $S$: The origin $O$ of the orthogonal co-ordinate system is chosen at $P_i$. In this way both $\dot{x}_p$ and $\dot{z}_p$ are independent on the co-ordinates of $S$. All this can be put into one matrix equation:

$$\begin{bmatrix} \dot{x}_p \\ \dot{z}_p \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} 1 & -z_p & z_p \\ \frac{L}{x_p} & \frac{L}{x_p} & x_p \\ -\frac{L}{L} & L & 1 \end{bmatrix} \begin{bmatrix} \dot{k}_1 \\ \dot{k}_2 \end{bmatrix} \tag{2.6}$$

### 2.2 Sensors

The mathematical equations, which describe the relations between movements of a sheet and sensor-observations, are derived in this paragraph.

#### 2.2.1 IntelliMouse

When placed above a moving surface, the IntelliMouse sensor detects movements in two orthogonal directions. The sensor $S_m$ is oriented above the surface at $(x, z) = (X_m, Z_m)$ with local orthogonal co-ordinate system $(x_l, z_l)$, which is rotated about an angle $\theta$ with respect to the global co-ordinate system (see Figure 2.3). So, it will measure movement in $x_1$ and $z_1$ direction.
The rotation about a constant angle $\theta$ can be presented with the following transformation:

\[
\begin{bmatrix}
\dot{x}_1 \\
\dot{z}_1 \\
\end{bmatrix} =
\begin{bmatrix}
\cos(\theta) & -\sin(\theta) \\
\sin(\theta) & \cos(\theta)
\end{bmatrix}
\begin{bmatrix}
\dot{x}_0 \\
\dot{z}_0 \\
\end{bmatrix}
\]  

(2.7)

\[
\begin{bmatrix}
\dot{x}_1 \\
\dot{z}_1 \\
\end{bmatrix} =
\begin{bmatrix}
\cos(\theta) & -\sin(\theta) \\
\sin(\theta) & \cos(\theta)
\end{bmatrix}
\begin{bmatrix}
1 & -\frac{Z_m}{L} & \frac{Z_m}{L} \\
\frac{X_m}{L} & 1 & -\frac{X_m}{L}
\end{bmatrix}
\begin{bmatrix}
\dot{k}_1 \\
\dot{k}_2 \\
\end{bmatrix}
\]  

(2.8)

With (2.6) the observations of the IntelliMouse can now be written as a function of $\dot{k}_1$ and $\dot{k}_2$ according to (2.8). By knowing the values of $X_m$, $Z_m$, $L$, and $\theta$ and measuring $x(t)$ and $z(t)$, $\dot{k}_1$ and $\dot{k}_2$ can thus be determined with (2.8). In this way the state of the paper beneath the pinches can be reconstructed.

\[
\begin{bmatrix}
\dot{k}_1 \\
\dot{k}_2 \\
\end{bmatrix} =
\begin{bmatrix}
\frac{1}{2} & \frac{1}{2} \\
\frac{1}{2} & \frac{1}{2}
\end{bmatrix}
\begin{bmatrix}
\dot{x}_1 \\
\dot{z}_1 \\
\end{bmatrix} =
\begin{bmatrix}
1 & 1 \\
1 & -1
\end{bmatrix}
\begin{bmatrix}
\dot{z}_1 \\
\dot{x}_1 \\
\end{bmatrix}
\]  

(2.9)

As an example, (2.9) shows the symmetric variant for $(X_m, Z_m) = (L/2, L/2)$ and $\theta = 0$. From (2.4) it can be concluded that displacement of a point in z-direction is caused by a change in angle $\phi$ (in other words by rotational velocity). However, when the registration unit is analysed again it learns that paper movement in z-direction can be realised without an effective change in $\phi$. One condition has to be fulfilled here, namely there has to be composed actuation in x- and $\phi$-direction. This will be explained using Figure 2.4: A sheet is transported in x-direction while the pinches cause a difference in angle $\Delta \phi = +\phi_0$ unequal to zero. The paper is still transported in x-direction keeping this angle $\phi + \phi_0$. After some time the paper is rotated back by the pinches around the same angle $-\phi_0$. The entire sheet is now displaced ($\Delta z$) in z-direction.
However, the mouse is placed above the paper-sheet and it will, in such case, not directly
detect any effective change in $\phi$ between initial- and end point. Although, the total sheet has
moved in $z$-direction (and of course in $x$-direction), the IntelliMouse will only detect an
effective displacement in $x$-direction and zero effective displacement in $z$-direction. In the
following paragraph an equation is derived to compute this sort of $z$-displacement dependent
on $x$ and $\phi$.

2.2.2 Edge-detection sensor

In Figure 2.3 a sensor $S_e$ can be seen ($X_{Se}$ represents the distance between pinch line and
sensor). With help of a rotation matrix a line (for example the paper-edge) in a moving co-
ordinate system can be translated into a line in a stationary co-ordinate system. The sensor
will detect and measure $z$-movement ($z_k$) along a line with constant $x$-value ($X_{Se}$). The
intersection of these two lines will give $z$-measurement of the edge-detection sensor (see
Appendix 2.1):

\[
\frac{dz_e(t)}{dt} = \dot{x}_c \phi(t) - \dot{\phi}(t) X_{Se}
\]  

(2.10)

In (2.10) $\dot{x}_c$ represents the paper velocity in $x$-direction at the point $(X_c, Z_c) = (0, L/2)$.
Looking between the pinches $(X_{Se} = 0)$ will give:

\[
\frac{dz_e(t)}{dt} = \dot{x}_c(t) \phi(t)
\]  

(2.11)

From this it can be concluded $z$-displacement between the pinches is the integral of the
product of transportation velocity (between the pinches) and inclination. So after
reconstructing $x_c$ and $\phi$ also $z_k$ can be computed by integrating (2.11). This equation can also
be written as follows:

\[
\phi = \frac{dz_e}{dx}
\]  

(2.12)
2.3 PMDC-Motors

The standard PMDC-motors can be described by two coupled differential equations. The first one is a mechanical differential equation dependent on the motor's angle $\alpha$:

$$J_\alpha \frac{d^2 \alpha}{dt^2} = T_e - D \frac{d\alpha}{dt} - C_f \text{sgn} \left( \frac{d\alpha}{dt} \right)$$

(2.13)

The second differential equation is an electrical one, which is dependent on the current $i$:

$$L_i \frac{di}{dt} = V_{in} - V_{emk} - Ri$$

(2.14)

The motor constant $K$ couples the equations above:

$$K = \frac{V_{emk}}{\alpha} = \frac{T_e}{i}$$

(2.15)

(2.13), (2.14) and (2.15) can be represented in the following block diagram (Figure 2.5). It shows the transfer from voltage input into angle output.

![PMDC-motor schematic](image-url)
3. Control design

In the foregoing chapter all transfers in the system have been dealt with: From input motor to output motor, from output motor to paper movement and from paper movement to observations of the sensors. Now the system is mathematically described, the possibilities of both controlling and observing should be checked. This is done in paragraph 3.1 and 3.2 respectively. After that, in paragraph 3.3 a suitable control law is developed. The variables to be controlled are reconstructed in paragraph 3.4. In paragraph 3.5 a trajectory is formulated, which enables moving the paper from any arbitrary initial state into any arbitrary end state. Finally in paragraph 3.6 a simulation is made controlling a piece of paper along the developed trajectory.

3.1 Controllability

Positioning the paper means it has to be moved from an initial state into an end state. One problem occurs here: With this registration unit it is not possible to actuate the paper sideward (= z-direction). However it is possible to get the paper from one z-position into another: Successively rotating, transporting, rotating back and transporting back will result in a z-displacement (see also paragraph 2.2). So it is impossible to control the paper along any arbitrary trajectory in three degrees of freedom. However the paper can be moved from any initial condition into any end condition. Following and controlling a paper point on the fly will lead to complicated differential equations (see (2.6)) in the control law. That is why a point on the pinch line is chosen as the one to be controlled, namely the point between the pinches (co-ordinates \((X_c, Z_c) = (0, L/2)\)). This point can be compared with the centre of the steer-axle of a two-wheeled mobile robot. At this point, (2.11) can be used to compute z-displacement. Knowing the variables to be controlled, the controllability of the kinematics can be determined: The state-variables are \(x_c\), \(\phi\) and \(z_c\), while \(k_1\) and \(k_2\) are the inputs of this non-linear system (the PMDC-motors are not taken into account yet). It can be written in the following form (using (2.1) and (2.5))

\[
\dot{\mathbf{x}} = \begin{bmatrix} \dot{x} \\ \dot{\phi} \\ \dot{z_c} \end{bmatrix} = \begin{bmatrix} 1 \\ \frac{1}{2} \\ \frac{1}{L} \end{bmatrix} k_1 + \begin{bmatrix} -1 \\ \frac{1}{L} \\ \frac{1}{2} \end{bmatrix} k_2
\]

(3.1)

The following question arises: Is this system, with three state variables and two inputs, fully controllable? It is if the following condition holds [1] and Appendix 3.1:

\[
\dim(\text{span}(g_1(\bar{x}), g_2(\bar{x}), [g_1(\bar{x}), g_2(\bar{x})])) = n = 3
\]

(3.2)

with

\[
[g_1(\bar{x}), g_2(\bar{x})] = \frac{\partial g_2(\bar{x})}{\partial \bar{x}} g_1(\bar{x}) - \frac{\partial g_1(\bar{x})}{\partial \bar{x}} g_2(\bar{x}) = \begin{bmatrix} 0 & 0 & -\frac{1}{L} \end{bmatrix}^T
\]

(3.3)

This means (3.2) is valid and the system is fully controllable. Physically this third vector \([g_1(\bar{x}), g_2(\bar{x})]\) which is independent of both \(g_1(\bar{x})\) and \(g_2(\bar{x})\), means it is possible to move the system in a third direction (z-direction) by a certain combination of the two inputs!
3.2 Observability

A system is fully observable if all state variables can be reconstructed from the sensors outputs. In this case it means the vector $\tilde{x}$, (3.1), should be reconstructed. The variables $x_c$ and $\phi$ can be reconstructed combining (2.6) and (2.8):

$$\begin{bmatrix} x_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \frac{Z_m}{L} & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \frac{Z_m}{L} & 1 \end{bmatrix} \begin{bmatrix} x_c \\ \phi \end{bmatrix} \quad (3.4)$$

And thus

$$\begin{bmatrix} x_1 \\ \phi \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \frac{Z_m}{L} & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \frac{Z_m}{L} & 1 \end{bmatrix}^{-1} \begin{bmatrix} x_1 \\ z_1 \end{bmatrix} \quad (3.5)$$

$$z_c = z_c - \phi X_s e \quad (3.6)$$

The reconstruction of the third variable $z_c$ can be done with help of (3.6). The output part of the system of (3.1) can be defined as

$$\tilde{y} = C\tilde{x} \quad (3.7)$$

in which:

$$\tilde{y} = \begin{bmatrix} x_1 \\ z_1 \\ z_c \end{bmatrix}^T \quad (3.8)$$

$$\tilde{x} = \begin{bmatrix} x_c \\ \phi \\ z_c \end{bmatrix}^T \quad (3.9)$$

$$\begin{bmatrix} x_1 \\ z_1 \\ z_c \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ \frac{Z_m}{L} & 1 & 0 \\ \frac{Z_m}{L} & 1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ \phi \\ z_c \end{bmatrix} \quad (3.10)$$

$$\tilde{x} = C^{-1}\tilde{y} \quad (3.11)$$

All of the state variables $x_c$, $\phi$ and $z_c$ can thus directly (linearly) be reconstructed if the matrix $C$ is regular. Appendix 3.2 shows that $\text{det}(C) = -X_m$. So, the IntelliMouse can be placed everywhere but between the pinches ($X_m = 0$) to obtain a fully observable system. This is a logical result because at the pinch line z-velocity caused by angle-rotation is equal to zero and $\phi$ cannot be reconstructed then.

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{2}{L} & 0 \\ 0 & \frac{2X_s e}{L} & 1 \end{bmatrix} \tilde{x} = \begin{bmatrix} 1 & 0 \\ \frac{2}{L} & 0 \\ \frac{2X_s e}{L} & 1 \end{bmatrix} \tilde{y} \quad (3.12)$$
3.3 Control

To make the errors of the paper positioning as small as possible, a control law has to be
designed. In the first paragraph of this chapter a suitable reconstruction out of the
IntelliMouse observations is made to turn the MIMO-system into two SISO-systems. In 3.3.2,
an extra control in third direction (z,) is added to the one created in paragraph 3.3.1.

3.3.1 Decoupling

Until now the dynamics of the PMDC-motors have been neglected. The total system can be
represented in the block-diagram as can be seen in Figure 3.1. The part included by the dotted
line is a so-called black box: The system has two inputs and dependent on the system in the
black box, these inputs result in three outputs.

In printers and copiers a piece of paper is always imported with a (nearly) constant velocity.
Knowing this the following control strategy is proposed: Input-Output feedback control is
used for the actuation directions; adding the error in z-direction a term in this Input-Output
control enables correction in z-direction. First the Input-Output control is determined by
decoupling the system. Assuming the motor equations are linear (Coulomb friction is avoided
in first instance) the differential equations can be transformed into transfer functions in the
frequency domain.

\[
\begin{bmatrix}
X_1 \\
Z_1
\end{bmatrix} =
\begin{bmatrix}
H_{11} & H_{12} \\
H_{21} & H_{22}
\end{bmatrix}
\begin{bmatrix}
U_1 \\
U_2
\end{bmatrix}
\]  \hspace{2cm} (3.13)

From (3.13) it is obvious that the outputs (X1 and Z1) are dependent on both U1 and U2! This
is called a MIMO-system (Multiple In, Multiple Out). It is more convenient to control a
system, consisting of SISO-systems (Single In, Single Out). So, a method has to be found to
transform the MIMO-systems into two SISO-systems. When the block diagram of Figure 3.1
is again taken into account we see the transfer functions of the two PMDC-motors are
decoupled.

\[
\begin{bmatrix}
K_1 \\
K_2
\end{bmatrix} =
\begin{bmatrix}
M_{11} & 0 \\
0 & M_{22}
\end{bmatrix}
\begin{bmatrix}
U_1 \\
U_2
\end{bmatrix}
\]  \hspace{2cm} (3.14)

Integrating (2.8) and transforming it into frequency domain results into:
When (3.14) is substituted in (3.15) the following can be derived using (3.13):

\[
\begin{bmatrix}
X_1 \\
Z_1
\end{bmatrix}
= \begin{bmatrix}
\cos(\theta) & -\sin(\theta) \\
\sin(\theta) & \cos(\theta)
\end{bmatrix}
\begin{bmatrix}
1-\frac{Z_m}{L} & \frac{Z_m}{L} \\
\frac{X_m}{L} & -\frac{X_m}{L}
\end{bmatrix}
\begin{bmatrix}
K_1 \\
K_2
\end{bmatrix}
\]  

(3.15)

When (3.14) is substituted in (3.15) the following can be derived using (3.13):

\[
\begin{bmatrix}
H_{11} & H_{12} \\
H_{21} & H_{22}
\end{bmatrix}
= \begin{bmatrix}
\cos(\theta) & -\sin(\theta) \\
\sin(\theta) & \cos(\theta)
\end{bmatrix}
\begin{bmatrix}
1-\frac{Z_m}{L} & \frac{Z_m}{L} \\
\frac{X_m}{L} & -\frac{X_m}{L}
\end{bmatrix}
\begin{bmatrix}
M_{11} & 0 \\
0 & M_{22}
\end{bmatrix}
\]  

(3.16)

and

\[
\begin{bmatrix}
M_{11} & 0 \\
0 & M_{22}
\end{bmatrix}
= \begin{bmatrix}
\cos(\theta) & -\sin(\theta) \\
\sin(\theta) & \cos(\theta)
\end{bmatrix}
\begin{bmatrix}
1-\frac{Z_m}{L} & \frac{Z_m}{L} \\
\frac{X_m}{L} & -\frac{X_m}{L}
\end{bmatrix}^{-1}
\begin{bmatrix}
H_{11} & H_{12} \\
H_{21} & H_{22}
\end{bmatrix}
\]  

(3.17)

The symmetric case leads to:

\[
\begin{bmatrix}
M_{11} & 0 \\
0 & M_{22}
\end{bmatrix}
= \begin{bmatrix}
1 & 1 \\
1 & -1
\end{bmatrix}
\begin{bmatrix}
H_{11} & H_{12} \\
H_{21} & H_{22}
\end{bmatrix}
\]  

(3.18)

\(H_{11}, H_{12}, H_{21}, \) and \(H_{22}\) (in s-domain) can be determined by measuring \(x_1\) and \(z_1\) at known inputs \(U_1\) and \(U_2\). With help of (3.17) transfer functions \(M_{11}\) and \(M_{22}\) can be calculated and a control law can be designed. Linear control is done using feed forward in acceleration, velocity and Coulomb friction and using lead filters as feedback. Fitting suitable control laws to these motors is done in Appendix 3.4.

### 3.3.2 Control of \(z_c\)

With help of the control laws tuned in Appendix 3.4 the paper movement beneath both pinchers is controlled, too. From this it can be concluded \(x_c\) and \(\phi_c\) are controlled as well, because they are linearly dependent of \(k_1\) and \(k_2\). However a third state variable \(z_c\) exists, which has to be controlled. As has been said before, the edge detection sensor can directly detect errors in \(z\)-direction. So control of \(z_c\) has to be added to the control law, developed in the foregoing paragraph. Without determining transfer functions a clever design of the total control is made assuming a constant velocity in \(x\)-direction is present: In the preceding paragraph a control law has been developed for each motor. Feedback of the errors in \(k_1\) and \(k_2\) can be formulated as follows (the lead filters are represented here as PD-controllers for convenience):

\[
u_1 = u_{1,ref} + P_{11}e_{k_1} + D_{11}\dot{e}_{k_1}
\]

(3.19)

\[
u_2 = u_{2,ref} + P_{22}e_{k_2} + D_{22}\dot{e}_{k_2}
\]

(3.20)

in which:

\[
e_{k_1} = k_{1,d} - k_{1,r}
\]

(3.21)

\[
e_{k_2} = k_{2,d} - k_{2,r}
\]

(3.22)
Using (3.1):

\[ u_1 = u_{1g} + P_{11}(e_{x_c} - \frac{L}{2}e_{\phi_c}) + D_{11}(\dot{e}_{x_c} - \frac{L}{2}\dot{e}_{\phi_c}) \] (3.23)

\[ u_2 = u_{2g} + P_{22}(e_{x_c} + \frac{L}{2}e_{\phi_c}) + D_{22}(\dot{e}_{x_c} + \frac{L}{2}\dot{e}_{\phi_c}) \] (3.24)

The following conclusions can be drawn from this: An error in \( x_c \) will add voltages with equal sign to both motors. In contrast, an error in \( \phi_c \) will add voltages with opposite sign.

\[ e_{x_c} = z_{cd} - z_{cr} \] (3.25)

\[ z_{cd} = \text{Desired trajectory of } z_c \]

\[ z_{cr} = \text{Real trajectory of } z_c \]

An error in \( z \)-direction can be undone by temporarily adjusting the trajectory of \( \phi \) while paper transport continues. For example, if \( z_{cr} \) is too large with respect to \( z_{cd} \) the trajectory in \( \phi \) has to be made smaller according to (2.12): Derivative \( \frac{\partial z}{\partial x} \) (therefore \( x \)-transport is necessary) has to get smaller to decrease \( z \)-velocity. The error in \( z_c \) will get smaller in this way; however, an error in \( \phi_c \) is created. The controller will also compensate this by rotating the paper back to desired \( \phi \). In this way the succession of rotating, transporting and rotating back is accomplished! The compensation for \( z_c \) is integrated in the control law as a P-action. \( \phi_{cd} \) should get smaller if \( z_{cr} \) is too large (\( e_{z_c} \) is negative). So the P-action should be added to \( \phi_{cd} \).

The control laws for \( M_1 \) and \( M_2 \) become:

\[ u_1 = u_{1g} + P_{11}(e_{x_c} - \frac{L}{2}e_{\phi_c} + P_{11z}e_{z_c}) + D_{11}(\dot{e}_{x_c} - \frac{L}{2}\dot{e}_{\phi_c}) \] (3.26)

\[ u_2 = u_{2g} + P_{22}(e_{x_c} + \frac{L}{2}e_{\phi_c} + P_{22z}e_{z_c}) + D_{22}(\dot{e}_{x_c} + \frac{L}{2}\dot{e}_{\phi_c}) \] (3.27)

The block diagram of the controlled system can be seen in Appendix 3.5.

Taking upper example into consideration the unit will successively decrease and increase \( \phi_c \) to compensate the \( z_c \)-error. So points behind the pinches will first move to the right (error in \( z \)-direction gets bigger here). If values from the edge-sensor \( z_c \), which is placed behind the pinches, would be fed back the error in \( z_c \) would just become larger and the system will intend to make \( \phi_c \) even smaller.
3.4 Reconstruction of $x_c$, $\phi$ and $z_c$

The outputs of the sensors will not deliver the state variables directly. Using (3.12) solves this problem. However an approach for estimating $\phi$, which is something different, is made here. The edge-detection sensor detects $z$-displacement caused by constant $\phi$. Therefore it would be suitable if $\phi$ would be estimated by the IntelliMouse at relatively high rotational velocities and by the edge-detection sensor at zero or small rotational velocities. (3.12) is used to estimate $\hat{\phi}_m$ out of the IntelliMouse and rewritten to reconstruct $\hat{\phi}_e$ out of the edge-detection sensor:

\[
\hat{\phi}_m = \frac{2 \cdot z_i}{L} \tag{3.28}
\]

\[
\hat{\phi}_e = \frac{\dot{z}_c + \hat{\phi}_m X_{se}}{\dot{x}_c} \tag{3.29}
\]

Because the edge-detection sensor detects slow changes (low frequencies) a low-pass filter is placed behind the estimation of $\hat{\phi}_m$. The IntelliMouse detects higher frequent movements and therefore a high-pass filter is used. The sum of these filters has to be equal to one to give a proper estimation of $\phi$.

\[
\ddot{\phi} = \frac{1}{\alpha_1 + 1} \ddot{\phi}_e + \frac{\alpha_1}{\alpha_1 + 1} \ddot{\phi}_m \tag{3.30}
\]

Now $z_c$ can be estimated using (3.11):

\[
z_c = z_c - \frac{2 X_{se}}{L} z_i = z_c + X_{se} \ddot{\phi} \tag{3.31}
\]

3.5 Trajectory

A piece of paper is imported at initial state $(x_{c0}, z_{d0}, \phi_0)$. The unit has to be able to move the sheet from this initial state into any desired end state $(x_{c\infty}, z_{d\infty}, \phi_\infty)$. It is convenient to design a trajectory for $z_c$, dependent on $x_c$, because in this way the trajectory in $\phi$ is easy to determine using (2.12). The second and third derivative from $z_c$ with respect to $x_c$ have to be equal to zero. The input of the PMDC-motor is a voltage, which will result in an instant acceleration. That is why, normally, positioning to a point means first and second derivative at initial- and end state have to be equal to zero. So first and second derivative of $\phi$ (with respect to $x_c$) have to be zero. This means second and third derivatives of $z_c$ (with respect to $x_c$) have to be zero. Together with initial $z$-position $z_{c0}$ and initial inclination $(dz_c/dx_c)_0 = \tan(\phi_0)$, end position $z_{c\infty}$ and end inclination $(dz_c/dx_c)_{\infty} = \tan(\phi_\infty)$ this leads to 8 initial conditions. A seventh order polynomial $(z_c(x_c) = \sum c_i x_c^{8-i})$ for which 8 constants ($i = 1 \ldots 8$) is therefore developed.

Generating the trajectory in Matlab Workspace and loading it during real-time control will take a lot of time. Because second and third derivative will always be equal to zero, at both beginning and end, 4 degrees of freedom will remain. The trajectory then can be built up from 4 basic curves in which $x$ will be normalized into $x_0$ (between zero and one). The first two are trivial: $z_1(x_c) = 1$ and $z_2(x_c) = x_c$. The other two can be defined by the $S_n$-curve and $R_n$-curve. The $S_n$-curve $(z_{3r}(x_c))$ changes $z$-position (from 0 into 1) without changing the inclination (zero in the beginning and in the end). The $R_n$-curve $(z_{3d}(x_c))$ changes the inclination ($\tan(\phi_0) = 1$, $\tan(\phi_\infty) = -1$), without changing the $z$-position. In Figure 3.4 all four basic curves can be seen.
These four basic curves are placed into a matrix. When this matrix is multiplied with a vector containing the boundary conditions the trajectory $z(x)$ is obtained.

$$z_c(x_c) = \begin{bmatrix} \frac{dz_1t}{dx} & \frac{dz_2t}{dx} & \frac{dz_3t}{dx} & \frac{dz_4t}{dx} \end{bmatrix}$$

$$B = \begin{bmatrix} z_{c0} & T_m & z_{ce} - z_{c0} - T_m & T_d \end{bmatrix}$$

$$T_m = \frac{\tan(\phi_0) + \tan(\phi_e)}{2}$$

$$T_d = \frac{\tan(\phi_0) - \tan(\phi_e)}{2}$$

Given all boundary conditions and the desired trajectory in $x_c$ dependent on time, the trajectories of $\dot{k}_1$ and $\dot{k}_2$ can easily be determined using (3.1). By making a model of (2.6) in Simulink the trajectory of the edge points of the paper sheet can now be generated (Appendix 3.6). The path of the paper sheet can then be simulated in Matlab. In Figure (3.5) this is done for the following boundary conditions (upper left edge point) and constant transportation velocity: $x_{c0} = 0\text{m}$, $x_{ce} = 0.1\text{m}$, $dx/dt = 0.25\text{m/s}$, $z_{c0} = 0\text{m}$, $z_{ce} = 0.005\text{m}$, $\phi_0 = 0$, $\phi_e = 0$. In this figure a sheet is shown at successive time steps ($dt = 0.04s$).
Figure 3.5  Simulation of desired paper track
3.6 Simulation
The developed control law and reconstruction method can now be checked in a simulation. Several simulations are done to adjust and optimise the control law. In this paragraph the results of the most satisfying controller are shown. Because the IntelliMouse is placed behind the pinches, control is not possible at the moment the paper is imported by the pinches. When the IntelliMouse detects any movement, control can be started. In the simulation this is implemented as follows: The pinches are accelerated to constant velocity of 0.3 m/s when the paper is imported beneath them. At time $\tau$, the paper comes beneath the sensor and the velocity in x-direction will be about 0.3 m/s. This time $\tau$ is saved and used to shift the control-trajectory in time (for example: $x_{\text{cd}}(t)$ becomes $x_{\text{cd}}(t-\tau)$). At this time $\tau$ the control can start. The trajectory in x-direction dependent on time will be constant velocity (0.3 m/s). The trajectory in $z_c$ dependent on $x_c$ will be a seventh order polynomial developed in paragraph 3.5. The boundary conditions are: $\frac{dx}{dt} = 0.3$ m/s, $z_\phi = 0$ m, $z_\phi = 0.003$ m, $\phi_0 = 0$, $\phi_\phi = 0$. Furthermore, an initial error in $z_c$ (-0.0001 m) is generated to check the control in z-direction. The initial errors in $x_c$ and $\phi$ are both equal to zero. Both noise and slip are added to the motor velocities. In this simulation $\tau$ is equal to 0.65 seconds and $dt=0.001$ s. The $z_c$-correction will be made in 1/3 of a second. So the paper will be transported $0.3/3 = 0.1$ m in x-direction and 0.003 m in z-direction with initial and end inclination equal to zero! When the $z_c$-correction is done the control will go on and only movement in x-direction (0.3 m/s), until time $t = 2$ seconds, is desired. The Simulink and M-files are shown in Appendix 3.7. In Figure 3.6 both desired and real trajectories as well as the errors of respectively $x_c$, $\phi_c$ and $z_c$ are plotted. The control laws consists of feed forward in acceleration, velocity and Coulomb friction and feedback using a lead filter and I-action at low frequencies:

$$H_i = \frac{s + 1}{s} \quad (3.38)$$

As can be seen the reconstructed trajectories of the state variables will match the desired trajectories quite well: All errors are quite small and will eventually go to zero. The $z_c$-control can be said to work well because the initial error is corrected. Satisfying results are obtained and now it is time to design a real registration unit and adjust the developed control law to it.
Figure 3.6  Desired and reconstructed state variables
4. System Identification
Identifying the system is necessary to create satisfying control and reconstruction. In the first paragraph of this chapter the orientation of pinches and sensors is determined. After that, in paragraph 4.2 transfer functions are measured and fitted. In paragraph 4.3, Coulomb friction of both motors is identified. Finally the edge detection sensor is calibrated in paragraph 4.4.

4.1 Orientation parts
Both motors have been aligned in the unit as good as possible. They can be moved in both y- and z-direction (see Figure 4.1). The IntelliMouse is fixed to a steel “bridge” which can be moved in x-direction.

The variables, which need to be known, are (see Chapter 1): L, X_m, Z_m, X_se and θ. Appendix 5.1 shows the experiment that is done to determine them.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>L(mm)</th>
<th>X_m(mm)</th>
<th>Z_m(mm)</th>
<th>θ (rad)</th>
<th>X_se(mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>151.04</td>
<td>89.89</td>
<td>71.78</td>
<td>0.014</td>
<td>38</td>
</tr>
</tbody>
</table>

Table 4.1 Orientation specific parts

4.2 Determination of the Coulomb friction
According to (2.13) the one non-linear term in the dynamics of a motor is Coulomb friction: The torque caused by Coulomb friction on the motor-axis is directed oppositely to the velocity (see Figure 4.2).

In practice, Coulomb friction is not equal in both velocity directions. This can be shown by the following experiment: a sine wave as an input to M_1 and zero input to the M_2. The observations of the IntelliMouse can be seen in Figures 4.3a and b.
Figure 4.3a  Response sine wave, Motor1, input $0.5 \sin(\pi t)$

The flat areas at the minima and maxima are caused by the tolerance in the gear casing.

Figure 4.3b  Response sine wave, Motor2, input $0.5 \sin(\pi t)$

To determine the Coulomb friction a “stair-wave” is used as an input. It has the shape of a sine wave, however the velocity will increase with steps, which hold on for some time. The system will start to move at one of these steps (Coulomb friction compensation). This can be seen in Figure 4.4a. In Figure 4.4b both input and $\dot{z}_i(t)$ are presented.
The values of Coulomb friction for $M_1$ and $M_2$ in terms of voltages (input) can be seen in Table 4.2. Velocity is defined positive in positive x-direction.

<table>
<thead>
<tr>
<th></th>
<th>Positive velocity</th>
<th>Negative velocity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motor 1</td>
<td>-0.32V</td>
<td>+0.29</td>
</tr>
<tr>
<td>Motor 2</td>
<td>-0.35V</td>
<td>+0.43</td>
</tr>
</tbody>
</table>

Table 4.2 Coulomb friction
4.3 Transfer functions of PMDC-motors

By knowing the parameters, determined in paragraph 4.1, the transfer from paper movement at the pinches into observations of the IntelliMouse can be determined with help of (3.14). Taking (3.17) into consideration only $H_{11}$, $H_{12}$, $H_{21}$ and $H_{22}$ need to be known to decouple the system into transfer functions $M_1$ and $M_2$. Because (3.13) shows two equations and four unknown transfer functions the method from paragraph 4.1 is used here too: One motor will be driven and the other one will not. When only Motor1 (Pinch1) is driven and thus $U_2$ is zero, (3.13) becomes:

$$
\begin{bmatrix}
    x_1 \\
    z_1
\end{bmatrix} =
\begin{bmatrix}
    H_{11} \\
    H_{21}
\end{bmatrix} U_1
$$

(4.1)

An excitation in time ($u_1(t)$) will result in $x_1(t)$ and $z_1(t)$ measured by the IntelliMouse. Using the command `tfe` in Matlab these transfer functions are transformed into the frequency domain, which leads to $H_{11}$ and $H_{21}$. The same is done driving only Motor2 (Pinch2), leading to $H_{12}$ and $H_{22}$. Because some time is needed to get a reliable estimation the "stair-wave with optimal offset" summed with white noise is used as an excitation here.

\[ u_1 = \"stair-wave\" \]

\[ u_2 = 0 \]

\[ w \]

\[ + \]

\[ \text{System} \]

\[ x_1 \]

\[ z_1 \]

\[ \text{Figure 4.5 Measuring transfer function} \]

Constant velocity would be preferred here because then the motor has permanently overcome its Coulomb friction. However, the paper sheet has restricted length and furthermore it will move out of the registration track if only one motor is driven in one direction. White noise is necessary to determine the response of the moving system at all frequencies. Both excitation ($u(t)$) and response $z_1(t)$ can be seen in Figure 4.6a. The observations of the IntelliMouse ($x_1(t)$ and $z_1(t)$) are presented in Figure 4.6b. The calculated transfer functions $H_{11}$ and $H_{21}$ can be seen in respectively Figure 4.6c and d.
In *Figure 4.6a* the white noise cannot be seen, however when the signal is derived with respect to time it can (*Figure 4.6b*). Compare *Figure 4.6b* with *Figure 4.4b*.

*Figure 4.6a*  
Response on input “stair wave” + noise

*Figure 4.6b*  
Input and $z_1\text{dot}$ response
Figure 4.6c Results of $H_{11}$

Figure 4.6d Results of $H_{21}$
Using the determined transfer functions $H_{11}$, $H_{12}$, $H_{21}$ and $H_{22}$, the parameters determined in paragraph 4.1, $M_1$ and $M_2$ can be derived now. In (3.18) the other two elements of the matrix at the left side are equal to zero (decoupling). So, it is to be expected these elements are very small with respect to $M_1$ and $M_2$! The results of $M_{11}$, $M_{12}$, $M_{21}$ and $M_{22}$ are shown in Figure 5.7. To identify the system a suitable fit of $M_{11}$ and $M_{22}$ has to be made. Figure 2.5 shows the structure of the transfer. Using this structure in the \textit{frfit} command, the following transfer functions are found:

$$M_{11}(s) = \frac{17.238}{s^2 + 33.999s} \quad (4.2)$$

$$M_{22}(s) = \frac{15.071}{s^2 + 33.355s} \quad (4.3)$$

These are also plotted in Figure 4.7. The last parts of the estimated transfer functions are not taken into consideration because the coherence decreases here! This means the reliability of the estimation decreases.
Figure 4.7a  Gain $M_{11}$, $M_{12}$, $M_{21}$, and $M_{22}$

Figure 4.7b  Angle $M_{11}$, $M_{12}$, $M_{21}$, $M_{22}$
4.4 Calibration edge-detection sensor

The output of the edge-detection sensor is a voltage. The magnitude of this output is dependent on the light-intensity on the photodiode. The following experiment is done (see Figure 4.8): A paper sheet is fixed to a ruler and with steps of 0.1 mm the paper sheet is moved across the prism. At every step the output is measured. Knowing the output is proportional to the input (in the working range) a linear fit can be made to the experimental data.

![Experiment: Calibration edge-detection sensor](image)

The results of both experiment and fit are shown in Figure 4.9. The fit is given in (4.4).

\[
d = 0.0015 - 0.0009 \cdot v
\]  

(4.4)

![Calibration edge-detection sensor](image)
5. Controlling the registration unit

Using the system identification a suitable control law and reconstruction can be developed for the registration unit. In the next three paragraphs the control law is checked for the registration unit: In paragraph 5.1 paper transportation is desired, in paragraph 5.2 paper rotation. Finally, in paragraph 5.3, a combination of transportation and rotation in the form of an S-curve is used as desired trajectory. Due to lack of time z-control is not added to the control law here and only 

5.1 Controlling x-transport

In this paragraph only x-transportation is desired. The paper is placed beneath the pinches and the IntelliMouse by hand. Then control can start: The trajectory of $x_c(t)$ is a seventh order polynomial with the following initial conditions: $t_0=0; t_f=6; x_{c,0}=0; x_{c,1}=0.18; (dx_c/dt)_0 = 0; (dx_c/dt)_f = 0$. The results are presented in Figure 5.2.

Because the IntelliMouse is oriented at some angle with respect to the global co-ordinate system it will detect $z_1$-movement as can be seen in Figure 5.2a and can be calculated by integrating (2.8). Looking at this figure the following facts are noticed: the error in $x_1$ direction is about 0.5mm. Using (1.8) the displacement in $x_1$-direction is desired to be $0.18\cos(0.014)=0.18m$. The error is just $100(0.0005/0.18) = 0.28\%$ of the total $x$-displacement. Furthermore the error in $z_1$-direction is about $100\mu m$.
When the trajectory of the error is taken into consideration two periodic signals can be seen. Knowing the radius of a pinch is equal to 0.015m, the pinch circumference can be calculated using $2\pi \cdot 0.015 = 0.0942m$. Because the displacement is 0.18m, the pinches will rotate $\frac{0.18}{0.0942} \approx 2$ times during the measured 6 seconds. One signal in the error plot shows this frequency. The elliptical (not perfectly round) shape of the pinches could cause the noticed signal. Looking at the error again a 50 Hz noise signal can be seen. This is the frequency of
the lighting system. Zooming in the error, the resolution of the IntelliMouse sensor \((=63.5 \times 10^{-6} \text{ m})\) can be noticed. Looking at Figure 5.2.c the edge detection detects movement on z-direction although this is not valid at the IntelliMouse measurements. This can be explained as follows: Because the initial conditions are not determined here and the paper sheet is imported by hand, the initial angle of the paper could not be exactly equal to zero. Now, during x-transport, the paper-edge moves towards or away from a static observer like the edge-detector. This illustrates the need of edge detection at once.

5.2 Controlling \(\phi\)-rotation

In the foregoing paragraph just x-transportation is desired, in this paragraph just \(\phi\)-rotation: The trajectories for the displacements at the both pinches are the same in magnitude, however they differ in direction. They are accelerated to some velocity, for some time this velocity is held and then it is decelerated until velocity zero. To avoid the paper sheet from colliding into the fixing of the motors and of the IntelliMouse sensor, a circle-shaped paper is formed (see Figure 5.3).

To check control, during rotation a triangle-formed pen is put on top of the paper. Despite of this disturbance, still the desired trajectory is followed. The results are shown in Figure 5.4.
Because the z-co-ordinate of the IntelliMouse sensor is smaller than half the distance between the pinches and the velocity of Motor1 is positive, the sensor will measure a positive $x_I$-displacement (Orientation IntelliMouse is negligible with respect to this). This is presented in Figure 5.5 and can also be seen in Figure 5.4b.

Again both periodic signals (caused by not perfectly round pinches and the lighting system) can be noticed. The gain is as high as possible here (almost unstable system) and that is why noise is shown clearly. Several other experiments are done to determine the highest velocity. The highest velocity obtained at the IntelliMouse (in a stable control loop) has been 24 cm/s. At higher velocities the IntelliMouse was not able to correctly detect movements and the system got instable. A cause for this is the fact that the paper bulges out during rotation. The required distance between sensor and paper was not present at these moments with the consequence the IntelliMouse was not able to detect well. Furthermore after several experiments the sensor could be charged electrically, which results in a wrong output.
5.3 S-curve

From the two preceding chapters it can be concluded $x_c$ and $\phi_c$ can be controlled separately. In this paragraph these two are composed into the so-called S-curve. This S-curve describes the path in $z$, dependent on $x$: During $x$-transport the paper is rotated about some angle and then rotated back around the same angle. The paper makes a S-shaped movement and a certain $z$-displacement is accomplished. With the polynomial described in paragraph 2.5, this path can be determined. In Figure 5.6 the results can be seen. The boundary conditions of the trajectory are: $t_0=0; t_6=6; x_c=0; x_c=0.18; (dx_c/dt)_0=0; (dx_c/dt)_6=0; z_c=0; z_c=0.01; \phi_c=0; \phi_c=0$. Due to some error the simulation has only been run for 5 seconds instead of six.

![Figure 5.6a Controlling S-curve, IntelliMouse](image1)

![Figure 5.6b Controlling S-curve, errors at IntelliMouse](image2)
As can be seen in Figure 5.6a the error in x-direction will be less than about 0.6 mm during the entire trajectory. Because in x-direction a distance of 0.18 m is covered, the relative error is very small. The error in z-direction will be less than about 0.4 mm and becomes very small (about 100 μm) at the end. Unfortunately the error at six seconds cannot be seen. However a satisfying result is obtained. Looking at the filtered measurements of the edge detections sensor it gets clear again it has a very restricted measuring range. However, within this range the observations can be called very satisfying because they approach the desired values.

In Chapter 3 the control law has been extended to obtain better results. However due to lack of time extending the control law for the registration unit is left out.
6. Conclusions

A paper sheet can be controlled from any initial state \((x_0, \phi_0, z_0)\) into any end state \((x, \phi, z)\) with help of two independently driven pinches. However, moving sideward (in \(z\)-direction) is not possible. A subsequence of rotation, transportation, back-rotation and back-transportation is needed to accomplish this. Some aspects have to be taken into consideration about observability: Using just the IntelliMouse, as a sensing device will not do, because it is not able detect the initial conditions itself. Furthermore reconstruction of the third state \(z_c\) cannot directly be done using the IntelliMouse. Namely, integrating the product of transportation-velocity and angle is needed to compute it. This will lead to a growing error An edge-detection sensor is added to the unit in order to overcome both problems. Now the system becomes fully observable, too. The used concept is the one with minimal sensing devices: Just the edge-detection sensor together with the IntelliMouse. Both sensors are placed behind the pinches so control can just start at the moment the paper is detected. Only a few limitations have to be made to the orientation of the sensors: The IntelliMouse cannot be placed between the pinches, otherwise the system is not fully observable anymore. Placing this sensor too far away will restrict the registration range. Theoretically, it could be placed anywhere in the area between it. The optimal orientation is not searched for in this investigation and a co-ordination is chosen. The edge detection sensor, which has restricted measuring range, has to be placed somewhere around the IntelliMouse in the path of the paper edge. With help of the sensor orientation, the kinematics from sensors observations into paper movements (state variables \(x, \phi, z\) between the pinches) can mathematically be derived. When the developed design is translated into a model in Simulink, a suitable control law (controlling state variables \(x, \phi, z\)) is developed. The control law in \(x\) and \(\phi\) exists from a lead filter with I-action together with a feed forward of acceleration, velocity and Coulomb friction. Controlling \(z\) is accomplished by on-line-adapting of the desired trajectory in \(\phi\), dependent on the error in \(z\). Reconstruction of \(x\) is directly done out of the IntelliMouse, \(\phi\) is determined with the observations of the IntelliMouse at higher frequencies, whereas the edge-detection sensor estimates \(\phi\) at lower frequencies. The resultant of these two \(\phi\)-estimations is the final \(\phi\)-estimation. This resultant is used to reconstruct \(z_c\) out of the edge-detection. With this model, satisfying simulation results are gathered.

Now the system can be tested in practice: A TU/e DACS is used as data-acquisition tool, being an interface between PC and unit. With help of this experimental set-up satisfying results are obtained: Without determining the initial conditions \(x\)-transportation, \(\phi\)-rotation and an S-curve are given as desired tracks, respectively. Furthermore, both \(z\)-control and I-action are left out here. That is why the reconstruction of \(\phi\) is totally done out of measurements of the IntelliMouse (edge-detection sensor is not used here to estimate \(\phi\)). The error at the end of the tracks will be smaller than 200 \(\mu\text{m}\) every time, whereas the resolution of the IntelliMouse is 63.5 \(\mu\text{m}\). The rotation control is done at a speed, which approaches the maximum velocity: the IntelliMouse can detect movements until 0.3 m/s, the registration is done at 0.24 m/s. Both \(x\)-transportation and S-curve are achieved at relative low speed (0.18m \(x\)-displacement and 0.01m \(z\)-displacement in 6 seconds). Experiments at higher speed will lead to instability. Two reasons can be given for this: First of all, the paper will bulge out during registration, with the subsequence the required distance between sensor and paper is not present. Furthermore the sensor gets electrically charged when paper is moved beneath it and it will not detect movements correctly. However, looking at the results at the mentioned speed, the IntelliMouse delivers a smooth signal (without noise), which can be used perfectly in a control loop.
7. Recommendations

Following from the conclusions, some recommendations can be made for further investigations on this subject. First of all the complete control law developed in theory can be adjusted to the real unit: Integration of I-action and z-control should reduce the error. Because the edge detection sensor has restricted measuring range, z-control should only be adjusted if the paper is within this range. Because $\phi$ is partially estimated with help of z-measurement out of the edge-detection sensor, this estimation should entirely be left to the IntelliMouse at the moments the paper-edge is not within the measuring range. Furthermore an algorithm for determining the initial conditions still needs to be derived: When this is done the paper can be imported between the pinches and will be steered towards the sensors by only feed forward in velocity and acceleration. At the moment the sheet moves beneath the IntelliMouse and edge-detection sensor, the initial conditions are determined. Dependent on these initial conditions and the desired end state, a suitable trajectory can be generated with the calculated seventh order polynomial. In this way the unit can regisrate subsequent sheets with different initial conditions, into the same end conditions. When both control law and the determination of the initial conditions are integrated it should be investigated whether the speed of registration can be enlarged. The following improvements could be made to the existing experimental set-up to accomplish this: First of all a solution for the charging problem has to be made. Something has to be invented to lead away the charge. Furthermore the system could be enlarged with vacuum holes in the aluminium plate in order to pull down the paper to the plate. In this way the required distance between paper and IntelliMouse will be kept. When all this is done the system can be integrated in a real copier or printer to add import and export units and check the performance of the combination of these.
References

Appendix 2.1 Observations edge-detection sensor

A line, for example the paper edge, in a moving co-ordinate system can be translated to a stationary co-ordinate system with help of a transformation matrix. The variables that lead to movements are $x_e$, $\phi$ and $z_e$. The paper edge transformation can be described mathematically according to:

$$l_{fe} = R(x, \phi, z)l_{mc} = \begin{bmatrix} \cos(\phi) & \sin(\phi) \\ -\sin(\phi) & \cos(\phi) \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \phi \\ z \end{bmatrix} = \begin{bmatrix} z_h \\ \phi \\ z_h \end{bmatrix}$$  \hfill (A.2.1)

The observation of the edge-detection sensor is in $z$-direction. This can also be described in a vector:

$$l_e = \begin{bmatrix} x_e \\ z_e \\ 1 \end{bmatrix}$$  \hfill (A.2.2)

The point of intersection between these two lines will give the observations of the edge-detection sensor. This will lead to the following two equations:

$$\frac{z_h}{\phi} \cos(\phi) + z_h \sin(\phi) + x = x_e$$  \hfill (A.2.3)

$$-\frac{z_h}{\phi} \sin(\phi) + z_h \cos(\phi) + z = z_e$$  \hfill (A.2.4)

Substituting these two in each other will result in the observations of the edge-detection sensor

$$z_e = \tan(\phi)(x + \sin(\phi)z_h - x_h) + \cos(\phi)z_h + z$$  \hfill (A.2.5)

Deriving this equation with respect to time will result in:
For small angles and evaluated from a point that is between the pinches ($x = 0, z = L/2, \frac{dx}{dt} = \frac{(k_1+k_2)}{2}$ and $\frac{dz}{dt}=0$) this equation becomes:

\[ \ddot{z}_c = -\dot{x}_h + \dot{x} \phi \]  
(A.2.7)
Appendix 3.1 Controllability

The system defined in (3.1) has three state variables and only two inputs. The question is if the system is fully controllable, in other words, is control in all three directions possible. Equation (3.6) has to be fulfilled here. Therefore (3.7) has to be computed:

\[
[g_1(\bar{x}), g_2(\bar{x})] = \frac{\partial g_2(\bar{x})}{\partial \bar{x}} g_1(\bar{x}) - \frac{\partial g_1(\bar{x})}{\partial \bar{x}} g_2(\bar{x})
\]

\[
g_1(\bar{x}) = \begin{bmatrix} \frac{1}{2} & -\frac{1}{L} & \phi \end{bmatrix}^T, \quad g_2(\bar{x}) = \begin{bmatrix} 1 & 1 & \phi \end{bmatrix}^T
\]

\[
\frac{\partial g_1(\bar{x})}{\partial \bar{x}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad \frac{\partial g_2(\bar{x})}{\partial \bar{x}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}
\]

\[
\frac{\partial g_2(\bar{x})}{\partial \bar{x}} g_1(\bar{x}) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{2} \\ -\frac{1}{L} \\ \phi \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -1 \\ 2L \end{bmatrix}
\]

\[
\frac{\partial g_1(\bar{x})}{\partial \bar{x}} g_1(\bar{x}) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{2} \\ \frac{1}{L} \\ \phi \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 2L \end{bmatrix}
\]

\[
[g_1(\bar{x}), g_2(\bar{x})] = \frac{\partial g_2(\bar{x})}{\partial \bar{x}} g_1(\bar{x}) - \frac{\partial g_1(\bar{x})}{\partial \bar{x}} g_2(\bar{x}) = \begin{bmatrix} 0 \\ 0 \\ -1 \\ L \end{bmatrix}
\]

\[
\text{det} \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ -\frac{1}{L} & \frac{1}{L} & 0 \\ \phi & \phi & -1 \\ 2 & 2 & -L \end{bmatrix} = -\frac{1}{L^2}
\]

so

\[
n = 3
\]

This means change in one state variable is possible, while change (between initial and endpoint) in the other ones is equal to zero: Change in just x by k_1 = k_2, change in just \phi by k_1 = -k_2. Change in just z-direction by first k_1 = k_2 (rotation), then k_1 = k_2 (transportation), then k_1 = -k_2 (back-rotation) and finally k_1 = k_2 (back-transportation).
Appendix 3.2 Observability

All state variables \((x_o, \phi, z_o)\) can be reconstructed if the inverse matrix \(C^{-1}\) exists. \(C^{-1}\) exists if \(\det(C) \neq 0\).

\[
\begin{bmatrix}
x_1 \\
z_1 \\
z_2
\end{bmatrix} =
\begin{bmatrix}
\cos(\theta) & -\sin(\theta) & 0 \\
\sin(\theta) & \cos(\theta) & 0 \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
1 & -\frac{Zm}{L} & 0 \\
\frac{Zm}{Xm} & \frac{Zm}{L} & 0 \\
0 & \frac{L}{Xm} & 0
\end{bmatrix}
\begin{bmatrix}
1 & \frac{-L}{2} & 0 \\
\frac{L}{2} & 0 & 0 \\
0 & -\frac{1}{XSe} & 1
\end{bmatrix}
\begin{bmatrix}
x_o \\
\phi \\
z_o
\end{bmatrix}
\]

\[C = A \cdot B \cdot D\]  \hspace{1cm} (A.3.5)

\[\det(C) = \det(A) \cdot \det(B) \cdot \det(D)\]  \hspace{1cm} (A.3.6)

\[\det(A) = 1 \cdot \left(\cos^2(\theta) + \sin^2(\theta)\right) = 1\]  \hspace{1cm} (A.3.7)

\[\det(B) = 1 \cdot \left(\left(1 - \frac{Zm}{L}\right) \cdot \frac{Xm}{L} - \left(\frac{Xm}{L} \cdot \frac{Zm}{L}\right)\right) = \frac{-Xm}{L}\]  \hspace{1cm} (A.3.8)

\[\det(D) = 1 \cdot \left(\frac{L}{2} + 1 \cdot \frac{L}{2}\right) = L\]  \hspace{1cm} (A.3.9)

\[\det(C) = 1 \cdot \frac{-Xm}{L} \cdot L = -Xm\]  \hspace{1cm} (A.3.10)
Appendix 3.3 Determining initial conditions

Due to lack of time, an algorithm for determining the initial conditions of an imported sheet is not developed. However, one global method is given here. When a paper is imported (with constant velocity) the initial angle $\phi_0$ can be determined with help of Figure A.3.1. Both IntelliMouse and edge detection sensor serve as front-edge-detectors. The difference in time $\Delta t$ between the detection of IntelliMouse and edge-detection sensor is a measure for the angle. The initial angle can be determined with help of equation (A.3.11).

![Diagram](image)

**Figure A.3.1** Determining $\phi_0$

$$
\phi_0 = \arctan \left( \frac{x \Delta t - \Delta S_x}{\Delta S_z} \right) \quad (A.3.11)
$$

At the moment the sheet moves beneath the IntelliMouse, it can determine the transport velocity and thus $\phi_0$.

As soon as $\phi_0$ is determined the control law should bring the paper velocity to zero. After that rotation about Pinch 1 (only Pinch 2 moves) is done: Pinch 2 is controlled to accelerate until the edge detection sensor cannot see paper anymore. The edge-detection sensor serves as a side-edge detector here. The displacement of the paper beneath Pinch 2 is saved at the time the sensor detects the side-edge. Because the displacement is known, the orientation of the point, that is beneath the edge-detector, before rotation can be reconstructed.
By rotating back the paper moves into the previous conditions and the front-edge is beneath the IntelliMouse with angle $\phi_0$. The orientation of point $A_2$ can be calculated using:

$$x_{p_2} = L_{ps} \sin(\beta_2 - \beta_1)$$  \hspace{1cm} (A.3.12)

$$z_{p_2} = L_{ps} \cos(\beta_2 - \beta_1)$$  \hspace{1cm} (A.3.13)

Knowing the orientation of $A_1$ and $A_2$, $A_3$ can be calculated with Figure (A.3.3)

$$PP_1 = PP \cos(\phi_0 + \gamma)$$  \hspace{1cm} (A.3.14)

$$PP_2 = PP \sin(\phi_0 + \gamma)$$  \hspace{1cm} (A.3.15)

$$x_{A_3} = x_{A_1} + PP_1 \sin(\phi_0)$$  \hspace{1cm} (A.3.16)

$$z_{A_3} = z_{A_1} + PP_1 \cos(\phi_0)$$  \hspace{1cm} (A.3.17)
Appendix 3.4 Control of PMDC-motor

**Feedback**

In paragraph 2.3 the transfer (from voltage input into angle output) of a PMDC-motor is given schematically (Figure 2.5). Because L and D are relatively small, these terms can be neglected. Figure 1.5 can then be formulated as follows:

\[
\dot{\alpha} + \frac{K^2}{JR} \dot{\alpha} + \frac{C_f}{J} \arctan(\dot{\alpha}) = \frac{K}{JR} \text{Vin}
\]  

(A.3.18)

First the non-linear terms are neglected, to get a transfer function in s-domain:

\[
\frac{\alpha}{\text{Vin}} = \frac{K}{JR} \frac{s^2 + K^2}{s}
\]  

(A.3.19)

The corresponding Bode-diagram can be seen in Figure A.3.4a. Values for K, J and R are taken out of literature. Signals with a gain above 1 will be amplified; signals with a gain below 1 will be weakened. A phase delay of 180 degrees or more will lead to an unstable system. For frequency values below \( \frac{1}{\tau} = \frac{K^2}{2\pi JR} \) Hz, the process diagram has a slope equal to -1 (in semilogx-scale) and corresponding angle equal to -90 degrees. Above \( 1/\tau \) Hz the slope is equal to -2 and the corresponding angle is -180 degrees. Knowing the desired bandwidth of the controlled system a controller can be designed now. First some equations have to be given to understand the following:

\[
\frac{Y}{U} = CH \quad \text{Open loop} \\
\frac{Y}{R} = \frac{CH}{1 + CH} \quad \text{Closed loop}
\]  

(A.3.20)  

(A.3.21)

In this example \( \tau \) will be about 25 Hz and for example the desired bandwidth will be 50 Hz. By just enlarging the gain (P-action), signals of 50 Hz will be amplified, however phase-delay will be more go to 180 degrees and the controlled system will be instable then. By adding a D-action phase-margin is obtained (+90 degrees). This D-action should not be adjusted to all the frequencies: At lower frequencies this will lead to a slope equal to zero in the open loop transfer function. A slope of -1 is desired here because it leads to infinity at lower frequencies in the open loop (Figure A.3.4a) and 0 dB in the closed loop (Figure A.3.4c), (error = 0).

At higher frequencies a D-action will amplify disturbances. To avoid both these problems, a lead-filter is designed (Figure A.3.4b) just around 1/\( \tau \). Slope -1 is kept until 1/\( \tau \) and the gain can be increased until \( \tau \) lies at gain zero. Looking at the closed loop transfer function now, we can see the controller will force the system to follow a desired trajectory for frequencies below \( \tau \) and to damp it for higher frequencies.

\[
C = Kp \frac{\tau_d s + 1}{\tau_d s + 1}
\]  

(A.3.22)
Figure A.3.4a  Process

Figure A.3.4b  Controller

Figure A.3.4c  Open loop
To reduce the error between desired and real trajectory, besides feedback, feed forward is used. The desired acceleration and velocity are directly added to the input. Moreover a compensational term for the Coulomb friction is used to get the error towards zero. The motor is now steered by the feed forward and controlled by the feedback. All this can be shown in the following block diagram.
Appendix 3.5 Block diagram of controlled system

Figure A.3.6  Block diagram controlled system
Appendix 3.6 Simulation piece of paper

% M-file to generate desired trajectories for points beneath pinches
% U1 and U2. A desired trajectory for the point between the pinches is
% generated and dependent on the relation between xc and phiec on the
% one hand and U1 and U2 on the other the path for U1 and U2 is
determined.

tb=0;
tc=0.4;
xe=0;
xe=0.1;
zb=0;
ze=0.005;
phiec=0;
phiec=0;
br=0.2;
l=0.3;
L=0.1;

% tb=initial time simulation
% te=end time simulation
% xb=initial xposition point beneath mouse
% xe=end xposition point beneath mouse
% zb=initial zposition point beneath mouse
% ze=end zposition point beneath mouse
% phieb=initial inclination
% phiec=end inclination
% br=paper width
% l=paper length
% L=distance between pinches

Ax=[xb^7 xb^6 xb^5 xb^4 xb^3 xb^2 xb 1; xe^7 xe^6 xe^5 xe^4 xe^3 xe^2 xe 1; 7*xb^6 6*xb^5 5*xb^4 4*xb^3 3*xb^2 2*xb 1 0; 7*xe^6 6*xe^5 5*xe^4 4*xe^3 3*xe^2 2*xe 1 0; 42*xb^5 30*xb^4 20*xb^3 12*xb^2 6*xb 2 0 0; 42*xe^5 30*xe^4 20*xe^3 12*xe^2 6*xe 2 0 0; 210*xb^4 120*xb^3 60*xb^2 24*xb 6 0 0 0; 210*xe^4 120*xe^3 60*xe^2 24*xe 6 0 0 0];

bx=[zb ze tan(phieb) tan(phiec) 0 0 0 0]';

Cx=Ax\bx;
dt=0.0005;
T=[ ];
X=[ ];
DXDT=[ ];
Z=[ ];
DZDX=[ ];
Phieb=[ ];
Dphied=];
Dphiedt=[ ];
for $t=tb:dt:te$
    $x=xb+((xe-xb)/(te-tb))(t-tb)$;
    $dxdt=((xe-xb)/(te-tb))$;
    $T=[T;t]$;
    $X=[X;x]$;
    $DXDT=[DXDT;dxdt]$;
end

for $k=1:1:length(X)$
    $z=Cx(1)*X(k)^7+Cx(2)*X(k)^6+Cx(3)*X(k)^5+Cx(4)*X(k)^4+Cx(5)*X(k)^3+Cx(6)*X(k)^2+Cx(7)*X(k)+Cx(8);$
    $phie=(7*Cx(1)*X(k)^6+6*Cx(2)*X(k)^5+5*Cx(3)*X(k)^4+4*Cx(4)*X(k)^3+3*Cx(5)*X(k)^2+2*Cx(6)*X(k)+Cx(7));$
    $phiee=atan(7*Cx(1)*X(k)^6+6*Cx(2)*X(k)^5+5*Cx(3)*X(k)^4+4*Cx(4)*X(k)^3+3*Cx(5)*X(k)^2+2*Cx(6)*X(k)+Cx(7));$
    $dphiedx=42*Cx(1)*X(k)^5+30*Cx(2)*X(k)^4+20*Cx(3)*X(k)^3+12*Cx(4)*X(k)^2+6*Cx(5)*X(k)+2*Cx(6);$
    $dphiedt=dphiedx*DXDT(k);$ 
    $Z=[Z;z];$
    $Phie=[Phie;phie];$
    $Dphiedx=[Dphiedx;dphiedx];$
    $Dphiedt=[Dphiedt;dphiedt];$
end

figure(1)
plot(T,X)
figure(2)
plot(T,DXDT)
figure(3)
plot(X,Z)
figure(4)
plot(X,Phie)
figure(5)
plot(X,Dphiedx)
figure(6)
plot(T,Dphiedt)

$U1=DXDT-(Dphiedt*L/2);$ 
$U2=DXDT+(Dphiedt*L/2);$ 
figure(7)
plot(T,U1)
figure(8)
plot(T,U2)
In this Simulink file the path for a point on the paper is calculated dependent on \(U_1\) and \(U_2\).
% Simulation papersheet
% First run scurve578.m and XpZp.mdl!!
% Xp and Zp will be generated into the Workspace. Dependent on the
% dimension of the paper the four
% angular points of the piece of paper are calculated here!

function y=papiersim(Xp,Zp,Phie)

close all

br=0.2;
l=0.3;
Xpl=[];
Zp1=[];
Xp2=[];
Zp2=[];
Xp3=[];
Zp3=[];
Xp4=[];
Zp4=[];

for k=1:length(Xp)
    xp1=Xp(k)+br*sin(Phie(k));
    zp1=Zp(k)+br*cos(Phie(k));
    xp2=xp1-l*cos(Phie(k));
    zp2=zp1-l*sin(Phie(k));
    xp3=xp2-br*sin(Phie(k));
    zp3=zp2-br*cos(Phie(k));
    Xpl=[Xpl;xp1];
    Zpl=[Zpl;zp1];
    Xp2=[xp2;xp2];
    Zp2=[zp2;zp2];
    Xp3=[xp3;xp3];
    Zp3=[zp3;zp3];
end

U=[];
V=[];

for h=1:80:length(Xp)
    U(1)=Xp(h);
    V(1)=Zp(h);
    U(2)=Xp1(h);
    V(2)=Zp1(h);
    U(3)=Xp2(h);
    V(3)=Zp2(h);
    U(4)=Xp3(h);
    V(4)=Zp3(h);
    U(5)=Xp(h);
    V(5)=Zp(h);
figure(1)
plot(V,U)
axis([-0.1 0.3 -0.3 0.1])
title('z-correction')
xlabel('x(m)')
ylabel('z(m)')
hold on
pause(0.1)
U=[];
V=[];
end
Appendix 3.7 Simulation system
Appendix 4.1 Entire Unit
The entire system including mechanism, steering and sensors exists of several components, which can be seen in Figure A 4.1.

- **Control**: Dependent on the desired and real state of the paper control has to generate a suitable trajectory for the PMDC-motors.
- **TUeDACS**: The TueDACS is the data-acquisition tool of the unit. It translates trajectories coming out of the PC into PWM-signals recognisable for the motors and quadrature signals of the sensors into digital signals recognisable for the PC.
- **Servo Amplifiers**: Generate the trajectory of the motors given the desired PWM.
- **PMDC Motors**: The motors will accelerate dependent on the PWM en their own dynamics.
- **Kinematics Mechanism**: Dependent on the kinematics of the mechanism the pinch velocities will lead to the state of a paper sheet.
- **Sensors**: The sensors will observe the paper state dependent on their kinematics.
- **Reconstruction**: The sensor data are translated into the variables to be controlled.

In the next subparagraphs the specific hardware components of the system are discussed.

4.1.1 **TUeDACS**
The TUeDACS/1 Quadrature/Analog/Digital interface (QAD) is specifically designed for use in real-time closed-loop motion control systems.

The QAD consists of 2 quadrature counters, two 16-bit AD converters, two 16-bit DA converters, a general-purpose 8-bit I/O port, and a programmable, crystal-controlled clock generator which can serve as a time base for the AD and DA converters. The TUeDACS/1 QAD interface communicates with the host computer by means of the PhyDAS Serial Highway protocol. In this experiment the two quadrature counters are used for reading the outputs ($x_1$ and $z_1$) of the mouse sensor. One ADC is used for reading the analogue output of the edge-detection sensor. And both DACs are used for generating the trajectory for the
PMDC-motors. The hardware of the TUeDACS can be seen in Figure A 4.2. In Appendix 4.3 the interface is presented in a block diagram, in which the used parts are indicated.

4.1.2 Servo-amplifier
Both PMDC-motors are connected to servo-amplifiers. These will amplify the signal generated by the TUe DACS to drive the motors. In this experiment the reference signal (from DACS) is connected to the amplifier. Using an internal feedback loop this signal is amplified and sent to the PMDC-motor. The amplifiers require an unregulated DC-power supply. In Figure A 4.3 one used servo-amplifier is shown and the used connections have been indicated.

![Figure A 4.3 Servo amplifier](image)

4.1.3 PMDC-motor
The PMDC-motors are driven by the servo-amplifier, which generate their specific trajectory. The information about performances can be found in literature (Appendix 4.4). A suitable motor is chosen and they are fixed to the aluminium plate of the registration unit. The pinches are fixed onto the motor axes and enclosed with rubber rings. These rings are somewhat pushed against the aluminium plate so paper transportation is enabled between them.

![Figure A 4.4 PMDC-motor](image)

4.1.4 IntelliMouse
For detection of motion and velocity in copiers and printers a range of sensors is available. However most of these sensors are very expensive- or indirect measurement devices. Agilent brings a sensor without both these disadvantages on the market. This “IntelliMouse-sensor” uses the principle of correlation for, direct and contact free, detection and measurement of motion. In this investigation a mouse of type HDNS 2000 is used. The PC-board (Figure A 4.5) contains 4 specific components:
The HDNS-2000 contains the sensor-IC housed in a 16-pin optical package designed for through hole mounting on the PC-board. The sensor is comprised of three major functional blocks: an Image Acquisition System (IAS), Digital Signal Processor (DSP) and PS/2 or quadrature output converter. Images are reflected through a lens on the sensor, which consists of an array of photocells. The pictures are stored via the IAS by detecting the different potentials of the photocells. The DSP compares two succeeding images (respectively reference and sample image) and calculates the displacements in x- and z-direction. The output converter provides, in this case, a two-channel quadrature output, for direct interface to mouse microcontrollers.

The HDNS-2100 provides the optical path for the system. It consists of several optical components: the imaging lens through which the sensor acquires surface images; integral light pipe through which a LED provides the surface illumination; and lensed prism to focus the LED light at the optimal angle of incidence.

The HDNS-2200 is a clip, which holds the LED and aligns all of the specific components to the base plate. This design ensures optical alignment and requires no precision alignment. The HLMP-ED80 is the LED, which provides the illumination needed by the HDNS-2000 sensor for proper tracking.

The DSP of the HDNS-2000 reads out and processes the photo-elements at a rate of 1500 frames per second with a resolution of 400 cpi (counts per inch). Now, the maximal velocity, the sensor is able to detect, can be determined:

\[
\text{Resolution} = \frac{25.4 \cdot 10^{-3}}{400} = 63.5 \mu\text{m}
\]

The reference image within this sensor is an 8x8 array, so there are four pixels left to determine the displacement. So, per frame maximal four counts can be given and the time difference between two frames is equal to \(\frac{1}{1500}\) s:

\[
\text{Maximal velocity} = 4 \cdot 63.5 \cdot 10^{-6} \cdot 1500 = 0.381 \text{m/s}.\]

Recommend working conditions learn the sensor works properly up to 40°C and has a maximal acceleration of 0.15g (= 1.4715 m/s²). Besides, the navigation requirements for the optical engine are a flat reflecting surface with random texture or pattern characteristics. Finally the sensor has to be set between 2.3 and 2.5 mm above the surface for optimal working. The sensor generates two quadrature outputs, one for x-direction and the other for z-direction movement. One quadrature output is determined by two block signals. The phase difference between these two indicates the direction of movement and the length of the blocks indicates the velocity in the specific direction.
4.1.5 Edge-detection sensor

The used edge detection sensor is a light-to-voltage optical converter: Output voltage is directly proportional to the light intensity on the photodiode. This is valid in a restricted range (about 1.5 mm). The photodiode is most sensitive for light with a wavelength of 880 nm. To this a suitable LED is chosen. Because the paper has to move beneath the sensor the following problem arises here (see Figure A 4.6a): When the paper is moved between photodiode and LED it will knock into the PC-board. In Figure A 4.6b the solution to this problem is given: The PC-board containing photodiode and LED is totally placed above the paper track. A prism is placed beneath it in such a way that the optical features of the prism refract the LED-light on the photodiode! In this way the paper can move free between sensor and prism and all electronics are placed on one PC-board.

Figure A 4.6  Edge detection sensor
Appendix 4.2 Choice Registration Unit

In Figure A 4.7 a schematic representation of the registration unit can be seen. The paper is transported in positive x-direction by pinches P1 and P2, which can be driven independently by PMDC-motors M1 and M2 respectively. Difference in motor velocities will cause a rotation \( \phi \). The axes of M1 and M2 have to be aligned so the centre of rotation will be on the line through the motor axes. Besides, the velocities, at the points beneath the pinches, have to be perpendicular to this line. When this is not achieved the paper will bulge or stretch out while moving. For convenience this fictitious line is chosen parallel to the z-axis. The centre of rotation will always be on this “pinch line”. Placing the IntelliMouse now brings up one limitation: It has to have an x co-ordinate unequal to zero; otherwise it is placed on the pinch line and it will not be able to detect z-displacement! The IntelliMouse Sm is placed above the paper track beyond the pinches, so registration can start at the moment the paper comes beneath the Mouse. For the same reason the edge-detection sensor is placed behind the pinches. The larger the distance between IntelliMouse and pinch line the more accurate the measurements. However placing the Mouse further away will result in a smaller registration range. An optimum has to be found here. Such a compromise is also valid for the distance between the pinches: Bringing both pinches closer to each other will make the system more sensitive for velocity differences. A little disturbance will lead to both relatively large angular velocity and error, which is not suitable. However the distance between the pinches has to be smaller than the width of the paper so the pinches will keep in touch with the paper despite of z-adjustment.

4.2.1 Alternatives

In this chapter some alternatives for the registration unit are mentioned with their advantages and disadvantages. In A 4.2 several reasons are already given for the orientation of the specific part of the developed unit. However, for the sake of clearness this concept is given here again: The alternatives are all compared with this one and advantages and disadvantages with respect to the developed unit are given.

**IntelliMouse together with edge-detection sensor (developed unit)**

This concept (Figures A 4.7 and A 4.11) needs only two (cheap) sensors to control and observe the entire system. Both IntelliMouse and edge-detection sensor are placed behind the pinches so control can start at the moment the paper is detected. Using the IntelliMouse as an edge detector in the first instance the initial conditions of the paper sheet can be determined. The movement of the paper between the pinches can directly be reconstructed out of the IntelliMouse data: Slip between pinches and paper is controlled. However, this could lead to instability! Integrating the data out of the IntelliMouse will increase errors. So determining z-displacement with help of (2.11) is not the best option. Therefore the edge-detection sensor is used here to measure z-displacement. However, points behind the pinches will have non-minimum-phase behaviour. The observations of the edge-detection sensor are therefore converted into the movements on the spot of the pinches, where points have minimum-phase behaviour.
Displacing sensors
Placing both sensors in front of the pinches (Figure A 4.8) will have the following disadvantage: When the sensors first “see” the paper control cannot start immediately. The paper has to be moved beneath the pinches first. And then another initialization needs to be done, because the track from sensors-to-pinches is not controlled! The advantage of this concept is the fact that the edge-detection sensor now measures minimum-phase behavior. However, as follows from concept 1, it is easy converting edge-detection observations into movements at the pinches (with help of (2.10)).

IntelliMouse + edge detection in front of pinches
When one sensor is placed behind the pinches and the other in front of them the registration track gets very small!
**IntelliMouse together with two front-edge detection sensors**

In this alternative (Figure A 4.9) both edge detection sensors are integrated in the unit just for detecting the front edge. So they are only used for determining the initial conditions. The measurement of z-direction is not used here, so errors due to integration of the IntelliMouse data are not compensated for.

![Diagram of IntelliMouse + 2 front-edge sensors](image)

**Figure A 4.9  IntelliMouse + 2 front-edge sensors**

**Encoders on motor axes together with two front-edge detection sensors**

In this concept (Figure A 4.10) two encoders on the motor axes replace the IntelliMouse. They will measure the velocity of the motor. Furthermore two front-edge sensors are integrated behind the pinches. These sensors can just give 0 (paper above them) or 1 (no paper above them) as an output. With help of these sensors the initial conditions can be determined. Slip between paper and pinches is not measured by the encoders. So slip will not be compensated for in the control loop. On the other hand when slip is taken into control it could lead to instability: If the paper does not move while the motors rotate the measured error becomes larger and larger and the motors velocity will go to infinity. However if slip is not that large and input is not amplified to strong, this problem can be avoided.

![Diagram of Encoders + 2 front-edge sensors](image)

**Figure A 4.10  Encoders + 2 front-edge sensors**
Figure A 4.11  Registration Unit
Appendix 4.3 Tue-DACS

Figure A.4.1 Block diagram Tue-DACS
Appendix 4.4 Datasheet PMDC-motor

Umkehrmotor Plattform 1.61.046.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>( M_0 )</td>
<td>( n_0 )</td>
<td>( I_0 )</td>
<td>( R )</td>
<td>( i )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12 V</td>
<td>317</td>
<td>25</td>
<td>160.0</td>
<td>1900</td>
<td>1750</td>
<td>50</td>
<td>7.9</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>300</td>
<td>26</td>
<td>145.0</td>
<td>1600</td>
<td>1600</td>
<td>50</td>
<td>7.9</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>311</td>
<td>25</td>
<td>140.0</td>
<td>1600</td>
<td>1400</td>
<td>50</td>
<td>7.9</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>316</td>
<td>25</td>
<td>120.0</td>
<td>1500</td>
<td>1200</td>
<td>50</td>
<td>7.9</td>
<td>1</td>
</tr>
<tr>
<td>18 V</td>
<td>322</td>
<td>18</td>
<td>112.0</td>
<td>1250</td>
<td>1250</td>
<td>50</td>
<td>7.9</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>324</td>
<td>18</td>
<td>100.0</td>
<td>1200</td>
<td>1200</td>
<td>50</td>
<td>7.9</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>326</td>
<td>18</td>
<td>90.0</td>
<td>1100</td>
<td>1100</td>
<td>50</td>
<td>7.9</td>
<td>4</td>
</tr>
<tr>
<td>24 V</td>
<td>331</td>
<td>25</td>
<td>100.0</td>
<td>950</td>
<td>950</td>
<td>50</td>
<td>7.9</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>332</td>
<td>25</td>
<td>100.0</td>
<td>950</td>
<td>950</td>
<td>50</td>
<td>7.9</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>334</td>
<td>25</td>
<td>85.0</td>
<td>850</td>
<td>850</td>
<td>50</td>
<td>7.9</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>336</td>
<td>25</td>
<td>80.0</td>
<td>800</td>
<td>800</td>
<td>50</td>
<td>7.9</td>
<td>4</td>
</tr>
</tbody>
</table>

Standardtypen, ab Lager in Deutschland im 48-Stunden-Service erhältlich.
Artikel, die auf Anfrage gefertigt werden, aber nicht ab Lager lieferbar sind.

Kundenspezifische Anforderungen auf Anfrage.

**AUFBAU:**
- eingebaute DC-Kleinmotor mit Permanentmagneten
- Kommutierung durch Kohlebürsten
- Kollektor 7-teilig
- Grundmotordrehzahl seriell festliegend
(siehe Seite 11 von CE-Kennzeichnung)
- Stirnradgetriebe mit Zahnrad aus Metall
- Anschlüsse als Flachstecker 2,8 x 0,5
- Gewicht 150 g

max. zulässige Radiallast (10 mm ab Anschraubfläche) \( \leq 60 \) N
max. zulässige Axiallast \( \leq 5 \) N
Axialspiel \( 0,05 \) - \( 0,6 \) mm
zulässiger Temperaturbereich \(-20 \) °C til \( +60 \) °C

Über ein Suchprogramm finden Sie im Internet den Modelltyp nach Ihren individuellen Wünschen.
Internet: www.buehlermotor.de

**Figure A.4.2** Datasheet used PMDC-motors
Appendix 5.1 Determining $\theta$

$L$, $X_m$, $Z_m$, and $X_{se}$ are measured with help of a marking gauge and $\theta$ is estimated with help of an experiment and (2.8). From this equation can be concluded that if one motor is driven (and so one pinch will rotate) two equations can be formulated: $\dot{x}_i$ and $\dot{z}_i$ dependent on $L$, $X_m$, $Z_m$, $\theta$ and the velocity of the driven pinch. Knowing $L$, $X_m$, $Z_m$ delivers two equations with two unknown variables: The pinch velocity and $\theta$. When a constant motor input is given to the motor, the paper sheet will move, corresponding $\dot{x}_i$ and $\dot{z}_i$ are measured and $\theta$ can be estimated from them substituting one equation into the other (Appendix 5.1). The results of these experiments can be seen in table 4.1.

% M-file for identifying the registration unit
% Vectors Xdot and Zdot are generated in simulink and used for
% computing $\theta$ (orientation IntelliMouse)
% $X_1$-/Z1-dot values obtained with u2=0
% $X_2$-/Z2-dot values obtained with u1=0

function y=identify(X1dot,Z1dot,X2dot,Z2dot)

% Determination $X_c$, $Z_c$ and $L$
% Values in [mm]

X1dot=X1dot(2,:);
X2dot=X2dot(2,:);
Z1dot=Z1dot(2,:);
Z2dot=Z2dot(2,:);

Z1dot=-Z1dot;
Z2dot=-Z2dot;
L=151.04;
Xc=89.89;
Zc=71.78;

C1=(Xc-(0.5*L))/L;
C2=Xc/L;

teta1=atan(((Z1dot.*(0.5-C1))-X1dot.*C2)./(X1dot.*(0.5-C1)+C2.*Z1dot));
teta2=atan(((Z2dot.*(0.5+C1))+X2dot.*C2)./(X2dot.*(0.5+C1)-C2.*Z2dot));

c=1:1:length(teta1);
c1=1:1:length(teta2);

subplot(211)
plot(c,teta1)

subplot(212)
plot(c1,teta2)

teta1=teta1(10:length(teta1));
teta2=teta2(10:length(teta2));

meanteta1=mean(teta1)
meanteta2=mean(teta2)

save fidentifyteta c c1 teta1 teta2
Appendix 6.1 Suitable control law motor

In paragraph 5.3 the transfer functions of both PMDC-motors are determined and a nice structure-fit is made of them. With help of these fits a suitable control law for both motors can be developed. In Appendix 3.4 this is already done for values out of literature. Figure A.6.1 shows the transfer functions of respectively Motor1, the lead filter and the open loop of the structure-fit is made of them. With help of these fits a suitable control law for both motors can be determined.

\[
\frac{5.25}{1.5} = 3.5Hz \quad \text{and is ended at} \quad 2 \cdot 5.25 = 10.5Hz.
\]

In this way a satisfying result is obtained in the open loop transfer.

Figure A.6.1a Process

Figure A.6.1b Controller
Figure A.6.1c  Open loop
Appendix 6.2 Simulink model to control the unit