Design of a Ball Handling Mechanism for RoboCup

T. Bouwens, 0570536

DCT 2009.051

Bachelor Final Project

Supervisor: Prof.dr.ir. M. Steinbuch
Coach: J.J.T.H. de Best

Eindhoven University of Technology
Department of Mechanical Engineering
Dynamics and Control Technology group

Eindhoven, June, 2009
Abstract

This report describes the design, realization and testing of a new ball handling mechanism for Tech United, the Robocup team of the Eindhoven University of Technology. The goal of this project is to enable the robot to receive the ball and dribble with the ball.

The report will describe in detail the previous concepts that were implemented in the different versions of the robots. It will describe the improvements made on the new concept. The new concept involves motorized wheels attached to levers mounted under an angle of 90 degrees with respect to each other and optical sensors measuring the angle of the lever in respect to the robot. These wheels exert forces on the ball in order to control it. A theoretical model of the new concept has been inferred, using the Lagrange equations of motion, resulting in a linearized model around a desired equilibrium point. This model is verified by measuring frequency response on the experimental setup.

From this comparison a controller has been devised. This controller uses a cascaded structure, meaning it implements both a position and a velocity control loop to control the angle of the lever and therefore the position of the ball. The velocity control loop is added to add the option of introducing feed forward information of the movements of the robot itself. These movements are the biggest known disturbances on the ball handling mechanism. This cascading results in a closed loop system with a bandwidth of approximately 12 Hz.
Contents

1. Introduction 4

2. Design criteria 5
   Rules and regulations 5
   Tech United criteria 5

3. Previous concepts 6
   Static or passive control: 6
   Active controlled mechanism without ball position feedback 7
   Active controlled mechanism with ball position feedback 8

4. Current concept 10
   Layout 10
   Sensors 10

5. Identification and Control 11
   Theoretical model 11
   Practical Implementation 15
   Identification 16

6. Results 21

7. Conclusion and Recommendations 23
   Conclusion 23
   Recommendations 23

8. References 24

9. Appendix 25
   M-files used in FRF measurements 25
1. Introduction

RoboCup\(^1\) is an international project that promotes A.I. (Artificial Intelligence), robotics, and related fields. It is an attempt to foster A.I. and intelligent robotics research by providing a standard problem where a wide range of technologies can be integrated and examined. RoboCup chose to use the soccer game as a central topic of research, aiming at innovations to be applied for socially significant problems and industries. The ultimate goal of the RoboCup project is to develop a team of fully autonomous humanoid robots that can win against the human world soccer champion by 2050.

Robocup consists of various leagues. There are separate soccer leagues for various wheeled, legged and humanoid robots, but there are also simulation and rescue leagues. The Robocup team of the Eindhoven University of Technology, Tech United\(^2\), competes in the Middle Size League, in which wheeled robots autonomously play soccer.

This report describes the design, realization and testing of a ball dribbling mechanism for the Tech United TURTLE (Tech United Robot Limited Edition) robots. Various efforts have been made to build a mechanism to control the ball, but this still is an ongoing process. The goal of this project is to describe the previous concepts and evaluate and control the current concept.

The outline of the report is as follows. First, the design criteria will be presented in chapter 2. Chapter 3 will discuss the previous concepts of stopping and dribbling with the ball. This is followed by a description of the current concept in chapter 4 and the system identification and controller design in chapter 5. Finally, the results of these tests and the design itself will be evaluated in chapter 6.
2. Design criteria

This section will describe the design criteria to which the ball handling mechanism has to comply with. Firstly the rules and regulation of the Robocup competition and secondly the criteria set by the Tech United Robocup team are described.

Rules and regulations

This section describes the Rules and Regulations of the Mid-League soccer game regarding ball handling mechanisms. These rules are as close as possible to the FIFA rules but also contain extra parts that are relevant for ball handling. These rules are as followed:

- During a game, the ball must not enter the convex hull of a robot by more than a third of its diameter except when the robot is stopping the ball. The ball must not enter the convex hull of a robot by more than half of its diameter when stopping the ball.

- The robot may exert a force onto the ball only by direct physical contact between robot and ball. Forces exerted onto the ball that hinder the ball from rotating in its natural direction of rotation are allowed for no more than 4 seconds and a maximal distance of movement of one meter. Exerting this kind of forces repeatedly is allowed only after a waiting time of at least 4 seconds. Natural direction of rotation means that the ball is rotating in the direction of its movement.

- Violating any of the above rules is considered ball holding.

Tech United criteria

The goal of the ball handling mechanism for the Tech United TURTLE (Tech United Robot Limited Edition) is to catch the ball and keep it during a dribbling motion of the robot while obeying the rules mentioned above.

Another important criteria is the active ball handling. Most competitors use a static construction, consisting of points and/or wedges to guide the ball, relying on the robot itself to control and handle the ball.

The designed ball handling mechanism for the TURTLE has to consist of parts that are actively driven to exert extra forces onto the ball in order to change velocity and position in relation to the robot.
3. Previous concepts

This section will describe the previous concepts that were implemented in the Tech United Turtles. They can be subdivided into 3 different categories. Firstly, the static or passive control mechanism. Secondly, the active controlled mechanism without ball position feedback and thirdly the active controlled mechanism with ball position feedback.

**Static or passive control:**

Static or passive control means that the ball handling mechanism is not an actively actuated mechanism, but consists of points and/or wedges, as seen in Fig 3.1. These guide the ball in the desired direction without preventing the ball to roll in its natural direction of rotation.

The first problem with this concept is that the robot has to move around the center of the ball to change direction, as seen in Fig 3.2, which leads to more complex trajectory planning. Catching the ball is the second problem. While the robot is catching the ball the velocity of both robots should be synchronized to prevent the ball from bouncing away from the robot.
Active controlled mechanism without ball position feedback

Active control means that the ball handling mechanism consists of parts that are actively driven to exert extra forces onto the ball in order to change velocity and position in relation to the robot.

The first version of this concept had motorized rubber belts, shown in Fig 3.3, to push or pull the ball according to the movement of the turtle. This velocity is estimated by using the onboard encoders mounted on each of the Omni wheels. Changing the ball's direction still involved moving the robot therefore a complex trajectory was still necessary. The control of the ball mechanism however was limited to going forwards and backwards because the belts were placed parallel to each other.

A simplified schematic can be seen in Fig 3.5a. The parallel belts are represented by a single actuator, where the red arrow is the applied torque and the white arrow is the direction of rotation.

The next version had two 4-bar linkage systems with motorized wheels. The levers are placed under an angle of 90 degrees with respect to each other as seen in Fig 3.4. This was an improvement over the belts because they did not get cut that easily and they contacted the ball at an angle relative to each other, thus enabling a more directional control of the ball.

A simplified schematic can also be seen in Fig 3.5b. The 4-bar linkages are represented by two actuators, where the red arrow is the applied torque and the white arrow is the direction of rotation. Because of the different rotation directions some sideways movement of the ball is possible.
Active controlled mechanism with ball position feedback

In the previous concept the balls position with respect to the robot was unknown and it was assumed that the ball was in the correct position to be able to move the ball forwards or backwards. In order to get the desired position feedback a simple potentiometer was mounted at a hinge point of the 4-bar linkage such that the configuration can be measured. The signal of this sensor was used to control/adjust the velocity of the wheels.

Adding a sensor that gave the position of the ball with respect to the robot added the feedback necessary for controlling the ball without having to move the robot itself.

First of all, it creates the opportunity to drive sideways and turning around the center of the ball instead of the center of the ball, while still possessing the ball.

Secondly, the position of the ball with respect to the robot can be controlled within the range of the ball handling mechanism. This property can be exploited during a kick, where the ball can be pulled against the kicker or can be placed to the left or to the right of the kicker, which enables the robot to aim during a shot, without rotating the robot itself. Furthermore, during a dribble the ball will rotate in its natural direction at all times, independent of the texture of the field.

Finally, the trajectory planning problem becomes much simpler since it is not necessary to rotate around the vertical axis of the robot in order to change the direction during a dribbling motion, as seen in Fig 3.6.

Fig 3.6 Robot trajectory planning for active control.
The 4-bar linkage turned out to be too big for the new TURTLE robots and was replaced with a single lever, as seen in Fig 3.7. During testing this system worked great, making sure the ball was kept at the correct position during moving and enabling more complex movements to avoid opponents while keeping the ball in possession. It turned out to be the opponents and other robots that were the next big problem. The mechanism was not robust enough to cope with collisions with other robots and would break or, resulting in improper functioning.

The next version of the concept was designed to handle these collisions, while still being able to control the ball. A spring was attached to the lever to absorb the impacts, see Fig 3.8. The spring was stiff enough to prevent influencing the ball handling behavior, but compliant enough to flex on impact. The flex proved to be insufficient to prevent damage but a softer spring would influence the ball handling, so a different concept was desired.
4. Current concept

This section will describe the current concept of the ball handling. This concept was produced using the proven ball handling principal used by the previous TURTLEs but improving the robustness. Firstly, the new layout with the improvements of the ball handling mechanism will be described and finally the new sensors that are used are discussed.

Layout

The previous concept had the levers mounted halfway up the robot which needed extra bracing on the robot, making the construction more spacious and complex. The new design has the levers of the ball handling mechanism mounted on a rigid base plate, making a very compact construction possible. The levers are replaced by stronger A-arms and more protection is added to the wheels, as shown in Fig 4.1. The A-arms are mounted on the base plate using rubber-cushioned ball bearings to still be able to absorb energy during collisions.

Sensors

The potentiometers are replaced with optical sensors because the attachment of the lever has changed from a fixed hinge to the rubber-cushioned bearings, (see fig 4.1) This prevents the potentiometer from being mounted at the hinge-point. The optical sensors now detect the distance between the robot and the wheel directly, instead of the angle of the lever.
5. Identification and Control

*Theoretical model*

This section describes the theoretical model for the new ball handling mechanism. The 2-dimensional simplified model of the ball handling mechanism is schematically depicted in Fig. 5.1. In this figure, the cart represents the soccer robot.

![Theoretical model diagram](image)

*Fig 5.1: The theoretical model.*

A lever is mounted on the robot that can freely rotate around a fixed axis on the robot by means of a hinge. The height of this point is located at a distance $h$ below the center of the ball as can be seen in Fig. 5.1. The lever has a length $l$, a mass $m_l$ and a rotational inertia $J_l$. At the end of the lever a wheel with mass $m_w$ and rotational inertia $J_w$ is mounted. The radius of the wheel is denoted by $R_w$. A torque $T$ can be applied between the wheel and the lever by means of a motor to control the angle of the lever and to counteract the viscous damping $d$ that is present between the lever and the wheel. During dribbling the wheel is assumed to be in contact with the ball at all times. The ball has a radius $R_b$, a mass $m_b$ and a rotational inertia $J_b$. Furthermore, it is assumed that there is no slip between the wheel and the ball nor between the ball and the floor.

The motion of the robot is prescribed and is denoted by $s(t)$. The idea is to measure the angle of the lever $\phi_l$ and maintain this angle at a preferred angle, which corresponds with a desired distance between the ball and the robot.
Lagrange equation of motion

Using Lagrange equation of motion to describe the system, the position vectors of the bodies’ center of mass, i.e. $\ell_l, \ell_w, \ell_b$, are derived as a function of the generalized coordinate $\phi$, and the prescribed coordinate $s(t)$ and are given by

\[
\ell_l = \left( s(t) + \frac{1}{2} l \cos(\phi_l), \frac{1}{2} l \sin(\phi_l) \right),
\]

\[
\ell_w = \left( s(t) + l \cos(\phi_w), l \sin(\phi_w) \right),
\]

\[
\ell_b = \left( s(t) + l \cos(\phi) + \sqrt{(R_b + R_w)^2 - (l \sin(\phi) - h)^2}, \frac{h}{R_b} \right).
\]

[1]

Also the rotation vector of each body, i.e. $\theta_l, \theta_w, \theta_b$ can be calculated as a function of the generalized coordinate $\phi$, and the prescribed coordinate $s(t)$, given by

\[
\theta_l = \phi_l,
\]

\[
\theta_w = s(t) + l \cos(\phi) + \sqrt{(R_b + R_w)^2 - (l \sin(\phi) - h)^2} - \arcsin \left( \frac{l \sin(\phi) - h}{R_w} \right),
\]

\[
\theta_b = s(t) + l \cos(\phi) + \sqrt{(R_b + R_w)^2 - (l \sin(\phi) - h)^2}.
\]

[2]

Using the position vectors the velocity vectors of the bodies can be calculated as

\[
v_l = \frac{dr_l}{d\phi_l} \frac{d\phi_l}{dt} + \frac{dr_l}{ds} \frac{ds}{dt} = \frac{dr_l}{d\phi_l} \dot{\phi}_l + \frac{dr_l}{ds} \dot{s},
\]

\[
v_w = \frac{dr_w}{d\phi_l} \frac{d\phi_l}{dt} + \frac{dr_w}{ds} \frac{ds}{dt} = \frac{dr_w}{d\phi_l} \dot{\phi}_l + \frac{dr_w}{ds} \dot{s},
\]

\[
v_b = \frac{dr_b}{d\phi_l} \frac{d\phi_l}{dt} + \frac{dr_b}{ds} \frac{ds}{dt} = \frac{dr_b}{d\phi_l} \dot{\phi}_l + \frac{dr_b}{ds} \dot{s}.
\]

[3]
The rotational velocities can be calculated similarly using the rotation vectors, given by

\[
\omega_i = \frac{d\theta_i}{d\phi_i} \frac{d\phi_i}{dt} + \frac{d\theta_i}{ds} \frac{ds}{dt} = \frac{d\theta_i}{d\phi_i} \dot{\phi}_i + \frac{d\theta_i}{ds} \dot{s},
\]

\[
\omega_w = \frac{d\theta_w}{d\phi_i} \frac{d\phi_i}{dt} + \frac{d\theta_w}{ds} \frac{ds}{dt} = \frac{d\theta_w}{d\phi_i} \dot{\phi}_i + \frac{d\theta_w}{ds} \dot{s},
\]

\[
\omega_b = \frac{d\theta_b}{d\phi_i} \frac{d\phi_i}{dt} + \frac{d\theta_b}{ds} \frac{ds}{dt} = \frac{d\theta_b}{d\phi_i} \dot{\phi}_i + \frac{d\theta_b}{ds} \dot{s}.
\]

With these results the kinetic energy of the system becomes

\[
K = \sum_{i \in \{l, w, b\}} \frac{1}{2} m_i \mathbf{v}_i^T \mathbf{v}_i + \frac{1}{2} J_i \omega_i \omega_i. \quad [5]
\]

The potential energy can be written as

\[
V = \frac{1}{2} m_l g l \sin(\phi_l) + m_w g l \sin(\phi_s) + m_b g h, \quad [6]
\]

where g is the gravity constant.

With the potential and kinetic energy known the Lagrange’s equations of motion can be calculated as

\[
\frac{d}{dt} \left( \frac{dK}{d\phi_i} \right) - \frac{dK}{d\phi_i} + \frac{dV}{d\phi_i} = Q_{nc}^T, \quad [7]
\]

where \( Q_{nc} \) are the non-conservative forces.

These forces consist of the applied torque \( T \) and the torque caused by the damping \( d \) between the lever and the wheel modeled as \( T_d = d(\dot{\phi}_w - \dot{\phi}_s) \)

These non-conservative forces are formulated as

\[
Q_{nc} = \left( \frac{d\theta_w}{d\phi_i} \frac{d\phi_i}{\phi_i} - \frac{d\theta_l}{\phi_i} \right) (T_d - T). \quad [8]
\]

Finally, a non-linear model can be constructed which has the form

\[
\ddot{\phi}_s = f(\phi_s, \dot{\phi}_s, \dot{s}(t), s(t), T) \quad [9]
\]
Linearization

In the previous section the non-linear model of the ball handling mechanism was derived. For the design of a stabilizing feedback controller of the feedback controller the non-linear model is linearized around the following desired equilibrium point

$$\phi_{eq} = \frac{x}{4}, \quad \dot{\phi}_{eq} = 0, \quad \ddot{\phi}_{eq} = 0, \quad \dddot{\phi}_{eq} = 0, \quad T_{eq}$$  \[10\]

In the choice of this equilibrium point it is checked whether or not the chosen equilibrium point satisfies the preferred distance between the ball and the robot. Other equilibrium points may be chosen as well, since this does not influence the presented control method. The torque $T_{eq}$, associated with the chosen equilibrium point, can be calculated by substituting the values of [10] into [9] and solving for $T_{eq}$ from

$$0 = f(\phi_{eq}, \dot{\phi}_{eq}, \ddot{\phi}_{eq}, \dddot{\phi}_{eq}, T_{eq})$$  \[11\]

Linearization around the point of operating results in the following second order linear model

$$G: \begin{cases} \dot{x} = Ax + Bu \\ y = Cx \end{cases}$$  \[12\]

where the state vector is given by $x = [\phi - \phi_{eq}, \dot{\phi}]^T$, the input $u$ is given by $T - T_{eq}$ and the output $y$ is defined as $\phi - \phi_{eq}$. The system matrix $A$, input matrix $B$ and output matrix $C$ are given by

$$A = \begin{bmatrix} \frac{df}{d\phi} & \frac{df}{d\phi} \\ \frac{d^2f}{d\phi^2} & \frac{d^2f}{d\phi^2} \end{bmatrix}, \quad B = \begin{bmatrix} \frac{df}{dT} \\ \frac{d^2f}{dT} \end{bmatrix}, \quad C = [1 \quad 0]$$  \[13\]

where $A$ and $B$ are both evaluated at the equilibrium point, [10].
Practical Implementation

This section will describe the controller structure used for the ball handling system. The structure will consist of two control loops, as seen in Fig 5.2.

One control loop for the angle of the lever and the other for the rotational velocity of the wheel. The optical sensor attached to the lever gives the position feedback for the controller. This feedback is then used to control the motor/wheel with the velocity control loop. The motor has a tachometer attached to the shaft for rotational feedback.

The main reason for this cascading is the robot itself. The movement of the robot is the biggest known disturbance on the ball handling mechanism and using a velocity control loop this movement can be anticipated, making the handling more accurate. Furthermore it adds flexibility to the controller. Even if the robot does not possess the ball, the velocity of the wheels can be controlled, for example for catching the ball.

The rotational velocity control loop gives an input torque $T(t)$ to the plant $H_{1,2}$, as seen in Fig 5.3, based on the error between the output velocity $\phi_w(t)$ and reference velocity $\phi_{w,r}(t)$.

The position control loop gives an input reference velocity $\phi_{w,r}(t)$ to the plant $H_2$, based on the error between the output lever angle $\phi_l(t)$ and reference angle $\phi_{l,r}(t)$, as seen in Fig 5.4.


Identification

This section will describe the identification of the three plants $H_{1,1}$, $H_{1,2}$ and $H_2$ in order to determine the controllers for an effective ball handling system.

These plants represent transfer functions. In order to determine these transfer functions or the transfer function estimates (tfe) a Fourier analysis of the input and output signals can be used. This analysis works on the basis that every signal consists of multiple trigonometric functions.

Using a white noise signal as a disturbance on the system, the response of the system can be measured. White noise is a random signal with a flat power spectral density. In other words, the signal contains equal power within a fixed bandwidth at any center frequency. Using the Fourier analysis on both the noise signal and the input signal, the response of the system to a specific input frequency can be calculated. This produces a Frequency Response Function (FRF) plot of the plant. The FRF shows the gain and phase plotted versus the frequencies.

In a Fourier analysis the input signal $u(t)$, a reference signal $r(t)$ and a white noise signal $w(t)$ have to be measured.

According to Fig 5.5 these input signals are $T(t)$ and $\phi_{w,r}(t)$ for the velocity an position control loop. The white noise signal $w(t)$ is the same in both cases. The reference signal is 0 for both loops.

The Fourier analysis uses a Fourier transformation on the signals, which has the form of

$$F(s) = \int_{-\infty}^{\infty} f(t) e^{-st} dt$$

where $s = \sigma + j\omega$.

Using [14] we get the following transformed signals:

$T(t) \rightarrow U_v(s)$,
$w(t) \rightarrow W_v(s)$,
$\phi_{w}(t) \rightarrow Y_v(s)$,
$e(t) \rightarrow E_v(s) \rightarrow -Y_v(s)$ for the velocity loop and

$\phi_{w,r}(t) \rightarrow U_p(s)$,
$w(t) \rightarrow W_p(s)$,
$\phi_r(t) \rightarrow Y_p(s)$,
$e(t) \rightarrow E_p(s) \rightarrow -Y_p(s)$ for the position loop.
Using the following formulas the transfer function estimate of the plant $H_{1,2}(s)$ can be determined for each frequency.

Firstly the Sensitivity function of the velocity control loop, the transfer of noise $W_v(s)$ to input $U_v(s)$, the 2 measured signals.

$$S_v(s) = \frac{U_v(s)}{W_v(s)}.$$ \[15\]

Rearranging the components, \[15\] can be written as

$$U_v(s) = W_v(s) + C_1(s)E_v(s),$$
$$U_v(s) = W_v(s) - C_1(s)Y_v(s),$$
$$U_v(s) = W_v(s) - C_1(s)H_{1,2}(s)U_v(s).$$ \[16\]

This leads to

$$S_v(s) = \frac{1}{1 + C_1(s)H_{1,2}(s)}.$$ \[17\]

Secondly the Process Sensitivity of the velocity control loop, the transfer of error $E_v(s)$ to noise $W_v(s)$, is derived.

$$E_v(s) = -Y_v(s),$$
$$E_v(s) = -H_{1,2}(s)U_v(s),$$
$$E_v(s) = -H_{1,2}(s) \cdot (C_1(s)E_v(s) + W_v(s)).$$ \[18\]

This leads to

$$PS_v(s) = \frac{H_{1,2}(s)}{1 + C_1(s)H_{1,2}(s)}.$$ \[19\]

Combining \[17\] and \[19\] the transfer function estimate of $H_{1,2}(s)$ has the following form,

$$H_{1,2}(s) = \frac{PS_v(s)}{S_v(s)}.$$ \[20\]
Plant $H_{1,2}$, as described in the previous section, is the transfer function of input motor torque $T(t)$ to the rotational velocity $\dot{\phi}_w(t)$ of the wheel.

Using [20] and measuring the input $T(t)$ and the white noise signal $w(t)$, the transfer function $H_{1,2}$ can be determined and plotted as an FRF, as seen in Fig 5.6. This is done in MatLab and Simulink, using m-files to do the calculations; these can be found in the Appendix.

From this FRF the influence of the inertia of the system can clearly be seen in the -2 slope of the amplitude versus frequency plot.
Plant $H_2$ could be determined similarly, measuring the input velocity $\varphi_{w,i}(t)$ and the output lever angle $\varphi_o(t)$, to determine the transfer between them. The robot however uses two levers in order to control the robot in a 2-dimensional environment and grab and handle the ball. This leads to a multi-input multi-output (MIMO) system, since there are now two actuators that can influence the motion of the ball. Furthermore, there are two sensors to measure the ball position.

In Fig 5.7 the FRF of each lever and the cross-coupling between levers can be seen.

By placing the levers under an angle of 90 degrees with respect to each other, seen from the top view, the cross coupling of the multi-input multi-output system can be minimized.

This minimization is validated via the measured relative gain array\(^5\) (RGA). The relative gain array provides a measure of interaction which is independent of input and output scaling.

The RGA is defined as $\Lambda(\tilde{H}) = \tilde{H} \times (\tilde{H}^{-1})^T$, where $\tilde{H}$ is the two-input two-output plant and $\times$ denotes element-by-element multiplication (the Schur product). In order to apply decentralized control it is preferred that this matrix is close to the unity matrix $I$ for all frequencies.
From Fig. 5.8 it can be seen that the system can be considered to be decoupled up to approximately 60 Hz. For this reason, the aimed bandwidth of the feedback controller will have to be less than 60 Hz in order to use a single-input single-output (SISO) controller.

The measured FRF for $H_2$ is then reduced to only the diagonal terms.

In order to get $H_{1,1}$, the transfer of input motor torque $T(t)$ to the output lever angle $\varphi_1(t)$, the same kind of measurements could be done. However after some simplifying of the original block diagram (Fig 5.9), it can be concluded that

$$H_2 = \frac{C_1}{1 + C_1 H_{1,2}} \cdot H_{1,1},$$

where the velocity control loop is depicted by

$$\frac{C_1}{1 + C_1 H_{1,2}}.$$

This leads to

$$H_{1,3} = \left( \frac{1}{C_1 + H_{1,2}} \right) \cdot H_2.$$

So $H_{1,1}$ is a combination of the measured plants $H_{1,2}$ and $H_2$. The only parameter in the equation is controller $C_1$, which has been chosen 1 for the measurements.

This leads to

$$H_{1,1} = (1 + H_{1,2}) \cdot H_2.$$
6. Results

This section describes the results obtained from the theoretical model compared with the frequency response measurements.

The parameters of the theoretical setup are given in Table 6.1:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>( l )</td>
<td>Length of the lever</td>
<td>0.2</td>
<td>[m]</td>
</tr>
<tr>
<td>( m_I )</td>
<td>Mass of the lever</td>
<td>0.4</td>
<td>[kg]</td>
</tr>
<tr>
<td>( J_I )</td>
<td>Inertia of the lever</td>
<td>( \frac{m_I l^2}{12} )</td>
<td>[kgm²]</td>
</tr>
<tr>
<td>( r_w )</td>
<td>Radius of the wheel</td>
<td>0.028</td>
<td>[m]</td>
</tr>
<tr>
<td>( m_w )</td>
<td>Mass of the wheel</td>
<td>0.1</td>
<td>[kg]</td>
</tr>
<tr>
<td>( J_w )</td>
<td>Inertia of the wheel</td>
<td>( \frac{m_w r_w^2}{2} )</td>
<td>[kgm²]</td>
</tr>
<tr>
<td>( r_b )</td>
<td>Radius of the ball</td>
<td>( \frac{0.8h}{2\pi} )</td>
<td>[m]</td>
</tr>
<tr>
<td>( m_b )</td>
<td>Mass of the ball</td>
<td>0.43</td>
<td>[kg]</td>
</tr>
<tr>
<td>( J_b )</td>
<td>Inertia of the ball</td>
<td>( \frac{2m_b r_b^2}{3} )</td>
<td>[kgm²]</td>
</tr>
<tr>
<td>( d )</td>
<td>Damping between lever and wheel</td>
<td>0.005</td>
<td>[Nmrad/s]</td>
</tr>
<tr>
<td>( h )</td>
<td>Height of rotation point of ball</td>
<td>0.1</td>
<td>[m]</td>
</tr>
<tr>
<td>( g )</td>
<td>Gravity constant</td>
<td>9.81</td>
<td>[m/s²]</td>
</tr>
</tbody>
</table>

Table 6.1: Parameters of the ball handling setup

With these parameters the linearized model \( G \)\(^{[12]}\) becomes

\[
\dot{x} = \begin{bmatrix} 0 & 1 \\ -26.43 & -27.36 \end{bmatrix} \cdot x + \begin{bmatrix} 0 \\ 659.18 \end{bmatrix} \cdot u \tag{20}
\]

\[ y = \begin{bmatrix} 1 & 0 \end{bmatrix} \cdot x \]

The poles of this system are located at 0.93 and -28.29, which indicates that the considered system is unstable.

The linearized system gives the theoretical transfer between input torque \( T(t) \) and the output angle \( \phi(t) \) which is the same transfer as the plant \( H_{1.1} \). The linearized system can be verified by comparing the FRF of the system with the measured FRF of plant \( H_{1.1} \). This comparison can be seen in Fig 6.1.
It can be seen that the model matches the measured frequency response measurement for low frequencies. However, beyond 40 Hz, the model does not match the measured FRF’s. Unmodeled artifacts of the ball handling mechanism show up in the measured frequency response function. One of these artifacts might represent the finite stiffness of the tires of the ball handling wheels. Moreover, since the control algorithm is implemented in discrete time the system also inherently suffers from time delay, which is also not taken into account in the modeling of dynamics of the ball handling mechanism.

By using loop shaping techniques⁵, an output feedback controller with a single proportional gain can be used to stabilize the system given in [20]. This feedback controller $C_f$ is given by $u = e = r - y$. Then, Fig. 6.1 can be interpreted as the open loop of the system, with a bandwidth of approximately 12 Hz. The closed loop poles are given by $-11.23 \pm 24.7i$, hence a stable closed loop system is obtained.
7. Conclusion and Recommendations

Conclusion

Previous concepts were evaluated for weak and strong points. A new setup was devised that incorporated the strong points and improved on the weak points. A non-linear model was formulated for the setup, which after linearization is used for the feedback control design, leading to a stable closed loop system with a bandwidth of approximately 13 Hz. The model is compared to experimental results and matches these results up to a frequency of 40 Hz. The difference can be explained by unmodeled artifacts in the system and time delay due to discrete control algorithms.

Recommendations

The following improvements of the ball handling mechanism for the Tech United TURTLEs can be made:

1. Evaluating the step response of the controller, further improving the handling. This will decrease the settling time of the mechanism on large disturbances, like catching the ball.

2. Implement feed forward into the control strategy, improving the dynamics of the system during moving. The motion of the robot itself can be anticipated resulting in a more accurate handling. Furthermore it adds flexibility to the controller. Even if the robot does not possess the ball, the velocity of the wheels can be controlled.
8. References


9. Appendix

M-files used in FRF measurements

Ballhandling_total.m

close all
clear all
cic

load G_ballpos1.mat
load G_ballpos2.mat
load G_ballvel1.mat
load G_ballvel2.mat

G_ball1 = (1+15*G_ballvel1).*G_ballpos1;
G_ball2 = (1+15*G_ballvel2).*G_ballpos2;

F=[0 1;21.28 -22.45];
G=[0;757.47];
H=[1 0];
J=[0];
sys=ss(F,G,H,J);

frfsys = 10*squeeze(freqresp(sys,2*pi*f));
figure

subplot(211)
semilogx(f,db(abs(G_ball2)),'Color','r','LineWidth',1)
ylabel([\text{Amplitude [dB]}])
title([\text{Plant Ball Handling}])
grid on
hold on
semilogx(f,db(abs(G_ball1)),'Color','c','LineWidth',1)
hold on
semilogx(f,db(abs(frfsys)),'Color','k','linestyle','--','LineWidth',2)
legend('Left','Right','Model')
subplot(212)
semilogx(f,angle(G_ball2)./pi*180,'Color','r','LineWidth',1)
xlabel([\text{Frequency [Hz]}])
ylabel([\text{Phase [deg]}])
grid on
hold on
semilogx(f,angle(G_ball1)./pi*180,'Color','c','LineWidth',1)
hold on
semilogx(f,angle(frfsys)./pi.*180,'Color','k','linestyle','--','LineWidth',2)
```matlab
Ballvel.m

clear all
close all
clc

Npoints = 2048;
Noverlap = 1/2*Npoints;
Fs = 1000;

load Tim_vel.mat

% Measured error ball handling 1
e1 = rt_ballveldata.signals.values(:,1);
% Measured error ball handling 2
e2 = rt_ballveldata.signals.values(:,2);
% Injected noise
w = rt_ballveldata.signals.values(:,3);
% Measured plant input ball handling 1
u1 = rt_ballveldata.signals.values(:,4);
% Measured plant input ball handling 2
u2 = rt_ballveldata.signals.values(:,5);

[c_S_w_u1,f] = mscohere(w,u1,hanning(Npoints),Noverlap,Npoints,Fs);
[c_S_w_u2,f] = mscohere(w,u2,hanning(Npoints),Noverlap,Npoints,Fs);
[c_PS_w_e1,f] = mscohere(w,e1,hanning(Npoints),Noverlap,Npoints,Fs);
[c_PS_w_e2,f] = mscohere(w,e2,hanning(Npoints),Noverlap,Npoints,Fs);
[S_w_u1,f] = tfestimate(w,u1,hanning(Npoints),Noverlap,Npoints,Fs); % S11
[S_w_u2,f] = tfestimate(w,u2,hanning(Npoints),Noverlap,Npoints,Fs); % S21
[PS_w_e1,f] = tfestimate(w,e1,hanning(Npoints),Noverlap,Npoints,Fs); % PS11
[PS_w_e2,f] = tfestimate(w,e2,hanning(Npoints),Noverlap,Npoints,Fs); % PS21

PS_w_e1 = -PS_w_e1;
PS_w_e2 = -PS_w_e2;

G_ballvel1 = PS_w_e1./S_w_u1;
G_ballvel2 = PS_w_e2./S_w_u2;

C_ballvel1 = (1./S_w_u1-1)./G_ballvel1;
C_ballvel2 = (1./S_w_u2-1)./G_ballvel2;

% Plot results

figure
subplot(211)
```
semilogx(f,db(abs(G_ballvel1)))
ylabel(['Amplitude [dB]'])
title(['Plant Ball Handling Velocity 1'])
grid on
hold on
semilogx(f,db(abs(G_ballvel2)),'r')
subplot(212)
semilogx(f,angle(G_ballvel1)./pi*180)
xlabel(['Frequency [Hz]'])
ylabel(['Phase [deg]'])
grid on
hold on
semilogx(f,angle(G_ballvel2)./pi*180,'r')

save G_ballvel1 G_ballvel1 f
save G_ballvel2 G_ballvel2 f
function frf_ballpos()

Npoints = 2048;
Noverlap = 1/2*Npoints;
Fs = 1000;

load Tim_pos.mat

% Measured error ball handling 1
e1 = rt_ballposdata.signals.values(:,1);
% Measured error ball handling 2
e2 = rt_ballposdata.signals.values(:,2);
% Injected noise
w1 = rt_ballposdata.signals.values(:,3);
% Injected noise
w2 = rt_ballposdata.signals.values(:,4);
% Measured plant input ball handling 1
u1 = rt_ballposdata.signals.values(:,5);
% Measured plant input ball handling 2
u2 = rt_ballposdata.signals.values(:,6);

if (~isequal(w1,zeros(size(w1)))) & isequal(w2,zeros(size(w2))) % noise injected in loop 1
    [c_S_w1_u1,f] = mscohere(w1,u1,hanning(Npoints),Noverlap,Npoints,Fs);
    [c_S_w1_u2,f] = mscohere(w1,u2,hanning(Npoints),Noverlap,Npoints,Fs);

    [c_PS_w1_e1,f] = mscohere(w1,e1,hanning(Npoints),Noverlap,Npoints,Fs);
    [c_PS_w1_e2,f] = mscohere(w1,e2,hanning(Npoints),Noverlap,Npoints,Fs);

    [S_w1_u1,f] = tfestimate(w1,u1,hanning(Npoints),Noverlap,Npoints,Fs);  % S11
    [S_w1_u2,f] = tfestimate(w1,u2,hanning(Npoints),Noverlap,Npoints,Fs);  % S21

    [PS_w1_e1,f] = tfestimate(w1,e1,hanning(Npoints),Noverlap,Npoints,Fs);  % PS11
    [PS_w1_e2,f] = tfestimate(w1,e2,hanning(Npoints),Noverlap,Npoints,Fs);  % PS21

    PS_w1_e1 = -PS_w1_e1;
    PS_w1_e2 = -PS_w1_e2;

    time_of_experiment1 = clock;

    save frf_ballpos1 f c_S_w1_u1 c_S_w1_u2 c_PS_w1_e1 c_PS_w1_e2 S_w1_u1
                     S_w1_u2 PS_w1_e1 PS_w1_e2 time_of_experiment1
elseif (isequal(w1, zeros(size(w1))) & (~isequal(w2, zeros(size(w2)))))

[c_S_w2_u1,f] = mscohere(w2, u1, hanning(Npoints), Noverlap, Npoints, Fs);
[c_S_w2_u2,f] = mscohere(w2, u2, hanning(Npoints), Noverlap, Npoints, Fs);

[c_PS_w2_e1,f] = mscohere(w2, e1, hanning(Npoints), Noverlap, Npoints, Fs);
[c_PS_w2_e2,f] = mscohere(w2, e2, hanning(Npoints), Noverlap, Npoints, Fs);

[S_w2_u1,f] = tfestimate(w2, u1, hanning(Npoints), Noverlap, Npoints, Fs);  % S11
[S_w2_u2,f] = tfestimate(w2, u2, hanning(Npoints), Noverlap, Npoints, Fs);  % S21

[PS_w2_e1,f] = tfestimate(w2, e1, hanning(Npoints), Noverlap, Npoints, Fs);  % PS11
[PS_w2_e2,f] = tfestimate(w2, e2, hanning(Npoints), Noverlap, Npoints, Fs);  % PS21

PS_w2_e1 = -PS_w2_e1;
PS_w2_e2 = -PS_w2_e2;

time_of_experiment2 = clock;

save frf_ballpos2 f c_S_w2_u1 c_S_w2_u2 c_PS_w2_e1 c_PS_w2_e2 S_w2_u1 S_w2_u2 PS_w2_e1 PS_w2_e2 time_of_experiment2

else
    error('Only injected noise in one of the two loops')
end
Ballpos_tot.m

```matlab
clear all
close all
c1c

try
    load frf_ballpos1.mat
    load frf_ballpos2.mat
catch
    error('First perform the two experiments')
end

S(1,1,:) = S_w1_u1;
S(1,2,:) = S_w2_u1;
S(2,1,:) = S_w1_u2;
S(2,2,:) = S_w2_u2;
S_ballpos = frd(S,f,'units','Hz');

PS(1,1,:) = PS_w1_e1;
PS(1,2,:) = PS_w2_e1;
PS(2,1,:) = PS_w1_e2;
PS(2,2,:) = PS_w2_e2;

PS_ballpos = frd(PS,f,'units','Hz');

G_ballpos = PS_ballpos*inv(S_ballpos);
[G,Hz] = FRDATA(G_ballpos);
figure;
subplot(211)
semilogx(Hz,db(abs(squeeze(G(1,1,:)))))
hold on
semilogx(Hz,db(abs(squeeze(G(2,2,:)))),'r')
subplot(212)
semilogx(Hz,angle(squeeze(G(1,1,:)))./pi.*180)
hold on
semilogx(Hz,angle(squeeze(G(2,2,:)))./pi.*180,'r')

% Plot results

% Plot measured sensitivity (amplitude)
figure;
bodemag(S_ballpos)
grid on

% Plot measured process sensitivity (amplitude)
figure;
bodemag(PS_ballpos)
grid on

% Plot measured plant (amplitude)
figure;
bodemag(G_ballpos)
grid on
```
% Calculate relative gain array (RGA)
RGA = [];
[G_ballpos_frf,f] = frdata(G_ballpos);
for i=1:length(f)
    RGA(:,:,i)=G_ballpos_frf(:,:,i).*pinv(G_ballpos_frf(:,:,i)).';
end
RGA = frd(RGA,f,'units','Hz');

% Plot relative gain array (RGA)
figure;
ymin = 0;
ymax = 5;
xmin = min(f);
xmax = max(f);
f=f(3:end);
RGA=RGA(:,:,3:end);
subplot(221)
semilogx(f,abs(squeeze(RGA(1,1,:))),'color','k','linewidth',2)
xlim([xmin xmax])
ylim([ymin ymax])
grid on
ylabel(['RGA [-]'])

subplot(222)
semilogx(f,abs(squeeze(RGA(1,2,:))),'color','k','linewidth',2)
xlim([xmin xmax])
ylim([ymin ymax])
grid on

subplot(223)
semilogx(f,abs(squeeze(RGA(2,1,:))),'color','k','linewidth',2)
xlim([xmin xmax])
ylim([ymin ymax])
grid on

ylabel(['RGA [-]'])

subplot(224)
semilogx(f,abs(squeeze(RGA(2,2,:))),'color','k','linewidth',2)
xlim([xmin xmax])
ylim([ymin ymax])
grid on

xlabel(['Frequency [Hz]'])

def_fontsize = 10;
def_tex_linewidth_cm = 221.0*3.5146e-2;
subplot(221); set(gca,'ytick',[0 1 2 3 4 5])
subplot(222); set(gca,'ytick',[0 1 2 3 4 5])
subplot(223); set(gca,'ytick',[0 1 2 3 4 5])
subplot(224); set(gca,'ytick',[0 1 2 3 4 5])

G_ballpos1 = squeeze(G_ballpos_frf(1,1,:));
G_ballpos2 = squeeze(G_ballpos_frf(2,2,:));

save G_ballpos1 G_ballpos1 f
save G_ballpos2 G_ballpos2 f