Reaction Wheels for Picosatellites

Angel Mario Cano Garza
Reaction Wheels for Picosatellites

A Thesis Presented to:

University of Würzburg

and

Luleå University of Technology

In (Partial) Fulfillment

of the Requirements for the Degree

Masters of Science

SpaceMaster: Master Course in Space Science and Engineering

Angel Mario Cano Garza

December 2008
Acknowledgments

This thesis could not have been what it is if not for the support of people from the Chair VII of the University of Würzburg in Germany and Luleå University of Technology in Sweden, I am thankful for the given opportunity of developing a topic in a very exciting area of application of small satellites.

Special thanks to my examiner Prof. Hakan Kayal who always kept a very open and enthusiastic attitude being there for frequent meetings and discussions since his arrival to the chair, it was delighting working with him. Thanks to my supervisor Prof. Klaus Schilling for putting all the pieces of the puzzle there for me in the department and being there when often I lost a piece or two.

Thanks to the people of Luleå University in Kiruna and Luleå, I take the chance to thank for the work done through all of the course, from the beginning Sven Molin worked a lot with us, then all the staff in Kiruna in our second semester, it was a wonderful time. Special thanks to Dr. Victoria Barabash and my examiner there Dr. Johnny Ejemalm who have been in very close collaboration despite the geographical distance.

Special thanks to Dr. Matthias Nienhaus at Mymotors & actuators GmbH for his exceptional support and suggestions in regards to the selected actuator.

Thanks to Dieter Ziegler for his support and patience in the robotics hall where I spent many days, his flexibility was key in the hardware development part of the project. Also thanks to the people in the University’s workshop for fabricating the flywheels.
Abstract

Picosatellite is the name given to the category of satellites that ranges from 0.1 to 1 kg mass. A popular standardized format is the CubeSat which is a cubed shaped satellite measuring 10 cm on its sides; introduced by Stanford University and California Polytechnic State University in the fall of 1999.

Since the first CubeSat launch in 2003 the choice for attitude control has been magnetotorquers, due to mechanical simplicity and weight restrictions, these offer low pointing accuracy when compared to momentum exchanged devices. However planned for launch on end of April 2008 are two missions that present reaction wheel devices, AAUsat-2 and CanX-2 in the first using commercially available motors and in the second using a product by Sinclair Interplanetary. A third project to use reaction wheels is BeeSat of Technical University of Berlin which will test the use of a newly developed coin sized reaction wheel developed by TU Berlin.

The objective of this thesis is to analyze the requirements for a wheel based ADCS (Attitude Determination and Control System) for a picosatellite using as much as possible COTS (commercial off the shelf) components and suggest a design to be used in the near future by the University’s own CubeSat program. Results are obtained by testing and simulations. When not available, test methodologies are to be designed.

The organization of the thesis seeks to answer the following questions in regard to a wheel based ADCS: When do we need it? What do we need? What is available? What can we produce? How can we produce it? And what performance is expected?
Contents

Nomenclature 4

1 Introduction 5
  1.1 Theoretical Background .................................. 5
    1.1.1 Basic concepts and equations .......................... 6
  1.2 Definition of Requirements .............................. 7
    1.2.1 Optical Payload .................................... 7
  1.3 Analysis of Wheel-Satellite motion ........................ 9
  1.4 External Disturbing Torques .............................. 10
    1.4.1 Gravity Gradient ................................... 11
    1.4.2 Solar radiation ...................................... 11
    1.4.3 Earth Magnetic Field ................................ 12
    1.4.4 Aerodynamics ....................................... 12
    1.4.5 Total Disturbance .................................. 13

2 Available solutions 14
  2.1 Ordinary RWA ............................................. 14
  2.2 DC Brushless Motor ....................................... 14

3 Reaction Wheel Assembly 17
  3.1 Mechanical Design ....................................... 17
    3.1.1 Flywheel ............................................ 17
    3.1.2 Mass Vs. Inertia .................................... 18
    3.1.3 The real flywheel .................................. 20
  3.2 Electronic Design ....................................... 20
    3.2.1 DC brushless motor .................................. 20
    3.2.2 Generating a linearly variable frequency ............ 22
    3.2.3 Built-in encoder performance ........................ 25
    3.2.4 Current sensing .................................... 26
  3.3 System Overview ......................................... 28

4 Controlling the RWA 32
  4.1 Computing the input to the controller .................... 32
  4.2 Dynamic response of the RWA ............................ 32
    4.2.1 Step response ....................................... 33
  4.3 Viscous friction compensation by speed control ........ 34
  4.4 Stability .................................................. 35
Nomenclature

\[ \dot{H}_s \] Satellite torque
\[ \dot{H}_w \] Reaction Wheel torque
\[ \theta_{s,a} \] Satellite displacement after accelerating phase
\[ \theta_{s,c} \] Satellite displacement after constant speed phase
\[ \theta_{s,d} \] Satellite displacement after decelerating phase
\[ \theta_{s,\text{total}} \] Satellite total displacement
\[ I_r \] Rotor mass moment of inertia
\[ I_s \] Mass moment of inertia of the satellite about the rotational axis
\[ n \] Percentage of total maneuver with torque action
\[ T_o \] Required torque
ADCS Attitude Determination and Control System
BLDC Brushless Direct Current
CCD Charged Coupled Device
COTS Commercial off the shelf
DDS Direct Digital Synthesis
RWA Reaction Wheel Assembly
UWE University of Würzburg Experiments
Chapter 1

Introduction

Spacecraft attitude control is necessary for a wide range of applications, basically every task in which controlled pointing is necessary i.e. Remote sensing and Communications. Not only for the end use a spacecraft needs attitude control, some payloads require that no direct sun light hits on them, like a sensitive image sensor, in contrast the solar panels are needed to be directed to the sun within a certain angle range, typically to a 5 deg accuracy [Bro02].

There are several methods for achieving attitude control, each of them has their strengths and weaknesses, it is the end application and payloads which determines the method to use, when high pointing accuracy is required as in the case of a space telescope, three-axis active stabilization is the selected scheme, for this scheme the typical actuators in normal sized satellites are thrusters and/or momentum exchange devices, this thesis is about the implementation of the second type of actuator, specifically a reaction wheel in small scale satellites. Picosatellites have not successfully proven to adopt this attitude control scheme yet, a very popular standardized version in this range of satellites is the CubeSat which has dimensions $10 \times 10 \times 10$ cm not exceeding 1 kg of mass, the University of Würzburg works in the UWE series, part of their CubeSat program. The aim of this thesis is to make a detailed analysis for including these momentum exchange devices in a picosatellite, at the time of initiation of this thesis no successful picosatellite incorporated this type of control scheme on orbit, however attempts are scheduled for this year to be launched, to name two examples: CanX-2 from University of Toronto (Triple CubeSat) and BeeSat from TU Berlin.

The approach to be followed seeks to develop as much as possible in-house at the University using COTS components.

1.1 Theoretical Background

If no forces are applied to a body this would tend to equilibrium, however even in space exist natural forces that in turn make bodies tumble, forces such as: Solar pressure, aerodynamics, gravity gradients and magnetic torques[Bro02] which in the context of attitude control are called disturbing forces. And for a good reason, even when these forces are relatively small, they have an undesired effect on the spacecraft attitude, when these forces act in asymmetrical directions produce torques, the effects of these torques
accumulate over periods of time and bring our satellite to an undesired attitude, thus we need to apply correcting torques for canceling the effects of such disturbances. Apart of counteract these disturbing forces we may also want to change the attitude of our spacecraft to perform an observation or point the antennas to a ground station for sending collected data. All these forces interacting in our system are described by the law of conservation of angular momentum, as a result we have that the torque delivered by a reaction wheel will be transferred to the satellite into a torque of the same magnitude with opposite direction[Sid97], as described by

$$\dot{H}_w = -\dot{H}_s.$$  \hspace{1cm} (1.1)

Eq. (1.1) expresses the torques in function of angular momentum, lets take a look at some basic equations that will help us see things more clearly.

1.1.1 Basic concepts and equations

In a rotational system such as the one depicted in figure 1.1 the torque generated by the motor is

$$T_m = \dot{H}_m = I\dot{\omega},$$ \hspace{1cm} (1.2)

in which $H$ is the angular momentum of the rotating parts, in this example the rotor (inside the motor), the shaft and the load, where $\dot{\omega}$ is the angular acceleration. The angular momentum, also known as momentum of momentum is described by

$$H_m = I\omega,$$ \hspace{1cm} (1.3)

where $I$ stands for mass moment of inertia of the rotating parts about the rotational axis, indicated with a dashed line in figure 1.1 and $\omega$ is the angular speed, the mass moment of inertia $I$ is a function of the geometric distribution of mass and shape of the solid body in question, it will be addressed further in the sections related to the mechanical design of the RWA.

To enable 3-axis attitude control at least 3 wheels aligned in orthonormal axes are needed as depicted in figure 1.2 on the facing page, it is common that 4 wheels are used for
redundancy purposes, however these topics will not be addressed in this work, instead the focus is on the design of a single RWA, it would be then the task of the attitude control engineer to make use of the actuator in different configurations, the interested reader can find more about spacecraft attitude dynamics and control in [Sid97, Cho91, Hug04].

1.2 Definition of Requirements

In order to be able to think about requirements for the RWA we need to assume a mission scenario, the following scenario was selected: A satellite with an optical payload capable of taking pictures of the Earth, specifically pictures that show the city of Würzburg.

Within this scenario some assumptions needed to be made, the first one is that our picosatellite is to be within the CubeSat standard, which already provides us with a geometric shape and a mass limit, which is of great help when dealing with mass moments of inertia. A second assumption is that our satellite is to be a triple CubeSat, the reasoning behind this decision is that a 3-axis stabilization system is needed for enabling relatively complex instruments, which at the time of the writing of this thesis are not usual for a single CubeSat satellite, instruments such as optical payloads present inherent physical restrictions such as focal lengths, an optical payload is a typical example of an application requiring precise pointing capability, for this reason we believe that making the final design available to a triple Cubesat, capable of longer focal lengths, makes more sense. However the final design is to comply with those mass, volume and power constraints of a single CubeSat. Figure 1.3 on the next page shows an artist conception of what such satellite might look like, in top of the satellite can be seen a circle that would be the aperture lens of a camera.

The minimum slew maneuver has been set to a 180 deg rotation in a total time of $10 \text{ min}$, which gives us a slew rate of $0.3 \text{ deg sec}^{-1}$, and the pointing accuracy is derived from an assumed optical payload, namely a CCD imager, this analysis is the topic of the next section.

1.2.1 Optical Payload

We must keep in mind that the maximum pointing accuracy that can be achieved is limited on sensor pointing capabilities, this means we cannot bring the satellite to a more
precise attitude than the one that we can determine, it will be assumed that the mea-
sured attitude is the real attitude, neglecting errors from sensor limitations. The assumed
optical payload is based on the e2v CCD57-10 CCD sensor and a 100 mm focal length $f$.
A maximum of 10% of image smear is allowed for an image to be considered acceptable.
As per the orbit, a circular orbit is assumed at an altitude $h=700$ km.

The sensor image area is composed of $512 \times 512$ pixels each measuring $13 \times 13$ µm
resulting in an area of $6.656 \times 6.656$ mm, using the amplification equation

$$\frac{f}{h} = \frac{l}{X},$$  \hfill (1.4)

where $X$ stands for swath width, solving (1.4) we find $X = 46.6$ km, this translates in
a spatial resolution of 91 m, the configuration is illustrated in figure 1.4 on the facing page.

The required pointing accuracy, 4.6 km at $h = 700$ km is

accuracy = 0.381 deg = 22.8 arcmin.

At $h = 700$ km we have a ground track velocity $V_{gt} = 6,762 \frac{m}{s}$, the shutter time is
selected as the time it takes for the ground projection of one pixel to pass at ground track
velocity[Wer99], thus

$$\tau_{shutter} = \frac{\Delta x_{1pix}}{V_{gt}} = \frac{91m}{6,762\frac{m}{s}} = 13.5 \text{ ms},$$

and the required stability is

$$\text{stability} = \frac{0.1 \times IFOV}{\tau_{shutter}} = \frac{0.1 \times 0.00745 \text{ deg}}{13.5 \text{ ms}} = 0.0553 \frac{\text{deg}}{\text{sec}} = 199.3 \frac{\text{arcsec}}{\text{sec}}.$$

Figure 1.3: Artist conception of candidate satellite. Credit. Stephan Busch.
1.3 Analysis of Wheel-Satellite motion

Taking a look in our maneuver requirements we a have a minimum maneuver of 180 deg performed in 10 min, this, and the worst case disturbances (next topic), determines the minimum torque authority or capacity to be delivered by the wheel. In order to be able to analyze the wheel-satellite motion we need to define the kind of maneuver that we wish to perform, the maneuver that we are to use for deriving the design parameters is to apply a constant torque from the RWA, then a period of no torque, or coast and finally apply an opposite torque in a symmetrical fashion, figure 1.5 on the next page depicts the torques applied and how these affect the position of the satellite.

The satellite total displacement is given by

\[ \theta_{s,\text{total}} = \theta_{s,a} + \theta_{s,c} + \theta_{s,d}, \]  

and the torque \( T_o \) needed for a maneuver is[CH65] given by

\[ T_o = \frac{\theta_{s,\text{total}} I_s}{n - n^2}, \]  

where \( n \) is a value from 0 to 0.5 and describes the percentage of time that the wheel will accelerate and decelerate, thus a value of 0.5 represents a time optimal maneuver[Sid97]
lacking of a no torque phase and \( I_s \) is the satellite mass moment of inertia about the axis of rotation.

With eq. (1.6) we can see how the torque needs change when varying each of the parameters, the mass moment of inertia of the satellite \( I_s \) is derived easily taking the geometric shape of a triple CubeSat about its axes and a 3 kg mass, it is found to be \( I_s = 0.025 \text{ kg} \cdot \text{m}^2 \) in the heavier axes. Having \( T_o \) it is time to look on the RWA side to see how we are to deliver \( T_o \), figure 1.6 on the facing page shows the output torque needed \( T_o \) vs \( n \) with the minimum target maneuver of 180 deg in 10 min.

To deliver this torque from the RWA we have

\[
T_o = \ddot{\theta}_w I_w, \tag{1.7}
\]

and we can find the needed angular acceleration by simply solving \( \dot{\theta} \text{max} = \ddot{\theta}_w n \cdot t_{total} \), the maximum speed and acceleration will depend on the chosen actuator and we must be careful they are well within its capabilities.

Once having \( T_o \) and \( \ddot{\theta}_w \) we solve eq. (1.7) to find the needed mass moment of inertia of the reaction wheel \( I_w \), this is a major input parameter for the design of the flywheel.

### 1.4 External Disturbing Torques

The RWA needs to be able to provide more torque than the one received by disturbing torques, as mentioned earlier there are four main sources of disturbing torques from the environment, these are: Gravity gradient, Solar radiation, Earth’s Magnetic Field and Aerodynamics. Next we calculate approximate worst case values based on equations from [Wer01] and [Bro02]:

![Figure 1.5: Relation between torques generated by the wheel and the satellite motion.](image-url)
1.4.1 Gravity Gradient

The effects generated by the Earth gravity gradient are described by

\[ T_g = \frac{3 \mu}{2R^3} |I_z - I_y| \sin(2\theta), \]  

(1.8)

where \( T_g \) is the max gravity torque; \( \mu \) is the Earth’s gravity constant \((3.986 \times 10^{14} m^3/s^2)\); \( R \) is orbit radius (m), \( \theta \) is the maximum deviation of the Z-axis from local vertical in radians, and \( I_z \) and \( I_y \) are moments of inertia about z and y (or x, if smaller) axes in kg·m².

The value of \( \theta \) is very important as it represents the inclination in which the torque arm will act, so the bigger this value the bigger the torque will be, as we need a pointing accuracy of 0.381 deg = 0.00665 rad. Solving

\[ T_g = \frac{3 \times 3.986 \times 10^{14} m^3/s^2}{2 \times (7078 \times 10^3 m)^3} \left[ 0.005 - 0.025 kg \cdot m^2 \right] \sin(2 \times 0.00665), \]

we get

\[ T_g = 4.485 \times 10^{-10} Nm. \]

1.4.2 Solar radiation

The effects generated by solar radiation are described by

\[ T_{sp} = F(c_{ps} - cg), \]  

(1.9)

where \( F = \frac{F_s}{c} A_s(1 + q) \cos i \) and \( F_s \) is the solar constant, 1,367 W/m², \( c \) is the speed of light, \( A_s \) is the surface area, \( c_{ps} \) is the location of the center of solar pressure, \( cg \) is the center of gravity, \( q \) is the reflectance factor (0-1, 0.6 typical.), and \( i \) is the angle of
incidence of the Sun.

First we need to find $F$, using the typical value of $q = 0.6$ and the worst case incidence angle which is normal to the surface $i = 0 \, \text{deg}$,

$$F = \frac{1367 \, W/m^2}{3 \times 10^8 \, m/s} (0.3 \times 0.1 \, m^2)(1 + 0.6) \cos(0),$$

we assume worst case, where the center of mass is located in the center of the satellite and the center of solar pressure acts in the furthest possible area of the longest face, this gives us

$$T_{sp} = 2.187 \times 10^{-7} \, N(0.15 - 0) = 3.28 \times 10^{-8} \, Nm.$$

### 1.4.3 Earth Magnetic Field

The effects generated by the Earth magnetic field are described by

$$T_m = DB,$$

where $T_m$ is the magnetic torque on the spacecraft; $D$ is the residual dipole of the spacecraft in $A - m^2$, and $B$ is the Earth’s magnetic field in Tesla. $B$ can be approximated as $\frac{2M}{R}$. $M$ is the magnetic moment of the Earth, $7.96 \times 10^{15} \, \text{tesla} \cdot m^3$ and $R$ is the radius from the dipole center to spacecraft in $m$.

Obtaining an accurate value of the residual dipole of the spacecraft can only be made by testing the actual satellite. Typical values range from 0.2 to 20 $A - m^2$ [Bro02], now we must keep in mind that these are values for regular sized satellites, so even the smallest suggested value may be too big. It was decided to take a look to the values of magnetically stabilized small satellites, such as the Korean Hausat-1 or the Danish AAUSat, these satellites when active have values of 0.022 and 0.075 $A - m^2$ respectively [Sch06], a value of 0.01 $A - m^2$ is suggested for picosatellites by [Gie06]. Because our study is for a triple Cubesat a value of 0.1 $A - m^2$ seems a safe guess. We solve

$$B = \frac{2 \times 7.96 \times 10^{15} \, \text{tesla} \cdot m^3}{((6378 + 700) \times 10^3 m)^3} = 4.49 \times 10^{-5} \, \text{tesla},$$

and substitute it in (1.12) to find

$$T_m = DB = (0.1 \, A - m^2)(4.5 \times 10^{-5} \, \text{tesla}) = 4.49 \times 10^{-6} \, Nm.$$

### 1.4.4 Aerodynamics

The Aerodynamic effects are described by

$$T_a = F(c_{pa} - cg) = FL,$$
where \( F = 0.5[\rho C_d A V^2] \). \( F \) being the force; \( C_d \) the drag coefficient (usually between 2 and 2.5); \( \rho \) the atmospheric density; \( A \), the surface area; \( V \), the spacecraft velocity; \( c_{pa} \), the center of aerodynamic pressure; and \( cg \) the center of gravity.

The worst case scenario is when there is solar maximum, then at \( h = 700 \text{ km} \), \( \rho \approx 4 \times 10^{-12} \text{ kg/m}^3 \).[Wer01]. At this altitude \( V = 7504.35 \text{ m/s} \). First we solve

\[
F = 0.5\left[4 \times 10^{-12} \text{ kg/m}^3 \times 2.5(0.3 \times 0.1 \text{ m}^2)7504.35 \text{ m/s}^2\right] = 8.45 \times 10^{-6} \text{ N},
\]

substituting \( F \) in (1.13) we find

\[
T_a = 8.45 \times 10^{-6} \text{ N}(0.15 \text{ m}) = FL = 1.27 \times 10^{-6} \text{ Nm}.
\]

### 1.4.5 Total Disturbance

The previous disturbing torques have been calculated assuming the worst case conditions, table 1.1 lists the resulting disturbing torques and a total which would represent the case in which all these forces act in the same direction at the same time, this however is very unlikely to happen in reality. Having measured the torque needs both for our target slew maneuver and the candidate disturbance torques, the actuator to choose typically should be able to produce at least double of this torque in order to have a 100% control authority margin[Bro02].

Table 1.1: Summary of disturbing torques and their worst case magnitude.

<table>
<thead>
<tr>
<th>Type of disturbance</th>
<th>Magnitude [Nm]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gravity Gradient</td>
<td>( T_g = 4.485 \times 10^{-10} )</td>
</tr>
<tr>
<td>Solar Radiation</td>
<td>( T_{sp} = 3.28 \times 10^{-8} )</td>
</tr>
<tr>
<td>Earth Magnetic Field</td>
<td>( T_m = 4.49 \times 10^{-6} )</td>
</tr>
<tr>
<td>Aerodynamics</td>
<td>( T_a = 1.27 \times 10^{-6} )</td>
</tr>
<tr>
<td><strong>Total Disturbance Torque</strong></td>
<td>( T_{Total} = 5.79 \times 10^{-6} )</td>
</tr>
</tbody>
</table>
Chapter 2

Available solutions

An obvious step to design something is to take a look to what has been already done, next we take a look to the anatomy of a commercially available Reaction Wheel Assembly.

2.1 Ordinary RWA

Figure 2.1 on the facing page is a sectional overview showing the main components of a commercially available RWA for regular sized satellites featuring a pancake shape, apart of the shape what can be seen as most relevant functional parts, there is a rotor flywheel which adds most of the inertia, Hall sensors for position sensing a DC motor, ball bearings, cover and case.

2.2 DC Brushless Motor

Nowadays the vast majority of the reaction wheel are built with brushless DC motors as actuator, which is an improvement over the basic brushed DC motor, its implementation as a RWA actuator boomed in the middle of the 60’s [CH65] for its noticeable improvement over the brushed DC motor and the more complex to control AC motor, among the advantages over their brushed counterpart we find that these allow for longer life as they do not suffer of brush friction, also the brushless DC motor presents a higher linearity.

For these reasons we selected the DC brushless motor as the actuator for our design, next we looked in the market of high quality motor manufacturers for candidates which would deliver the required torque and other characteristics for our application such as operational temperature ranges, maximum speed and price to name a few. The most interesting models are listed in table 2.1 on the next page.

After analyzing our options we decided for the Faulhaber 2209 even when it offers a lower maximum torque it still suffices with our requirement as seen in figure 1.6 on page 11 and table 1.1 on the previous page offering torque for faster maneuvers, it is heavier and more expensive than its counterparts but a big advantage is the integrated speed controller using a 12-bit capacitive encoder (actual value) which allows us to run accurate low speeds, this is a common problem with hall effect encoders and often reaction
wheels need to be operated at higher speeds where the hall sensors work better, another advantage was that the motor comes with an easy assembly configuration, with the use of three M3 screws, on figure 2.3 on the following page we can take a look to the back of the motor and also a very convenient 10 pin flexible cable and connector for power, control inputs and encoder output signals, a connector board was also provided for rapid prototyping. A higher rotor inertia $I_r$ is also favorable as it contributes for a lighter flywheel and more balanced rotating parts. An attractive feature is that a vacuum proof version of the 2209 can be ordered, a 1:1 diagram[Fau08] of the Faulhaber 2209 is shown in figure 2.2 on the next page.
Figure 2.2: 1:1 Diagram of the FH 2209 with dimensions in [mm]. Credit. Faulhaber.

Figure 2.3: Back of the Faulhaber 2209 micro motor.
Chapter 3

Reaction Wheel Assembly

RWA is the name given to the combination of a motor, an inertia flywheel and control electronics, in this chapter we talk about how these elements are designed or chosen and their interactions for being able to realize the desired task. The chapter is divided in three parts: Mechanical Design, Electronic Design, and a System Overview.

3.1 Mechanical Design

3.1.1 Flywheel

As mentioned earlier, one of the key characteristics for the flywheel design is its mass moment of inertia, another important consideration is that the material is non magnetic to avoid induced noises.

The material chosen for the fabrication of the flywheel is Aluminum EN AW-2007 which was previously used for the fabrication of the UWE-1 satellite chassis, one of its very attractive characteristics is that it is thermally stable, its density is $\rho = 2.85 g/cm^3$ [BatNA].

Design Constraints

A good starting point in the design phase of the flywheel is to detect those parameters that cannot be changed or that have a certain limit in order to simplify analysis, the selected constraints are the wheel diameter and the base where the shaft of the motor is to be mounted which at the same time will be used to limit the height of the flywheel. In the case of the diameter, it should not exceed that diameter of the motor as we may wish to encapsulate the RWA to make it vacuum proof and/or for radiation or thermal shielding also simply to use as little volume as possible in the satellite, the diameter of the motor is 22 mm so we will take that as our maximum flywheel diameter. In the case of the base of the shaft, figure 3.1 on the following page shows its design, note that the diameter of the shaft is 1.5 mm.
Having these constraints leaves us with a very clear idea of what our flywheel will look like.

### 3.1.2 Mass Vs. Inertia

The design of the flywheel was done targeting for the optimum point where our inertia needs are met without adding unnecessary mass to the end product. From

\[ I_z = \frac{1}{2} M (R_1^2 + R_2^2), \quad (3.1) \]

which describes the mass moment of inertia of a hollow cylinder about the axis depicted with arrowed dashed lines in figure 3.2 on the next page we can see that some of the design elements will add mass and inertia in a proportional manner, these cannot be optimized, our focus will be on those elements that mass and inertia are not proportional, namely the outer ring \( EO - EI \) shown in figure 3.3 on the facing page, since the moment of inertia is function of the distance to the axis of rotation, an example of an element that cannot be optimized is the main disk thickness as it varies parallel to the rotational axis.

On the shaft base no additional mass was added for inertia gain since the inertia/mass ratio is low due to its proximity to the axis of rotation, the top of the shaft base is the only element that is not a hollow cylinder but a solid one, using

\[ I_z = \frac{1}{2} MR^2, \quad (3.2) \]

we find the mass moment of inertia of that element. That said, lets take a look at the composition of the flywheel in figure 3.3 on the next page, our design is composed of solid and hollow cylinders combined, the outer ring, the base of the shaft, the top of the base of the shaft and the main disk. Having the dimensions of the cylinders we calculate the volume and then the mass \( M \) can be calculated with the density of the material.

The volume equations are
Figure 3.2: Solid and hollow cylinders construct the flywheel.

$$V_B = (\pi r_{B_o}^2 - \pi r_s^2)h_s + \pi r_{B_o}^2(h_B - h_s),$$  \hspace{1cm} (3.3)

where $V_B$ is the base volume, $r_{B_o}$ the outer radius of the base, $h_B$ the height of the base, $r_s = 0.75 \text{ mm}$ is the shaft radius and $h_s$ is the length of shaft that is inside the base.

$$V_M = (\pi r_{M_o}^2 - \pi r_{B_o}^2)h_M,$$  \hspace{1cm} (3.4)

where $V_M$ is the main disk volume and $r_{M_o}$ is the outer radius of the main disk.

$$V_E = (\pi r_{E_o}^2 - \pi r_{M_o}^2)h_E,$$  \hspace{1cm} (3.5)

where $V_E$ is the outer ring volume $r_{E_o}$ is the outer radius of the exterior ring. Finally the total volume is given by

$$V_{fw} = V_B + B_M + V_E.$$  \hspace{1cm} (3.6)

As stated before, there are constraints such as the radius of the exterior ring, not to be larger than $11 \text{ mm}$ or the shaft radius, other values cannot be optimized for inertia/mass ratio these are the values of such elements:
Figure 3.4: Inertia mass ratio Vs. $r_{E_i}$ [mm].

Table 3.1: Characteristics of the three fabricated flywheels.

<table>
<thead>
<tr>
<th>Maneuver</th>
<th>Rei [mm]</th>
<th>Ri [mm]</th>
<th>Mdt [mm]</th>
<th>Weight [g]</th>
<th>Inertia [gcm²]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slow</td>
<td>9</td>
<td>10</td>
<td>1</td>
<td>1.899</td>
<td>1.221</td>
</tr>
<tr>
<td>Medium</td>
<td>6</td>
<td>10</td>
<td>2</td>
<td>4.201</td>
<td>2.456</td>
</tr>
<tr>
<td>Fast</td>
<td>6</td>
<td>11</td>
<td>2</td>
<td>5.329</td>
<td>3.703</td>
</tr>
</tbody>
</table>

$r_s = 0.75$ mm, $h_s = 5$ mm, $h_B = 6$ mm, $r_{B_o} = 2$ mm, $r_{E_o} \leq 11$ mm, $h_M = 2$ mm and $r_{E_o} < r_{E_i} \geq r_{B_o}$, where $r_{E_i}$ is the internal radius of the exterior ring.

The final flywheel inertia is given by

$$I_{fw} = I_r + I_B + I_d + I_e, \quad (3.7)$$

where $I_r = 2.34g\cdot cm^2$ is a constant value and represent the rotor inertia, $I_B$ is the inertia of the shaft base, $I_d$ is the main disk inertia and $I_e$ is the inertia contribution of the exterior ring.

With those constrains the main variable to optimize is $r_{E_i}$, figure 3.4 shows a plot of the inertia/mass ratio with the previously chosen constrained values and varying $r_{E_i}$.

### 3.1.3 The real flywheel

Once the design phase was completed three flywheels were ordered for fabrication, table 3.1 depicts their properties, these were chosen for slow, medium and fast maneuvers, figure 3.5 on the facing page shows the three wheels, Appendix A contains the drawings.

### 3.2 Electronic Design

#### 3.2.1 DC brushless motor

As said earlier on section 2.2 the selected motor was the 2209 from faulhaber in this section we explain its characteristics in depth.
The motor operates at 5 V and its maximum consumed current is 90 mA, which translates in 450 mW peak power consumption. It is capable of running up to 10,000 rpm and producing torques up to 160 µNm, table 3.2 shows a detailed list of specifications.

### Integrated Speed Controller

One of the main advantages of the 2209 against its counterparts is the integrated speed controller which makes use of a high resolution (12 bit) capacitive encoder while its counterparts made use of three hall effect sensors with no integrated controller. Since the beginning of the project we were very excited to try out this motor as it promised outstanding performance thanks to its encoder and controller. As said earlier, it is well known that hall effect sensors have problems reading at slow speeds and often need to be run at a higher speeds for improving accuracy.
Quadrature signals from the encoder are available at pins Qa and Qb, shown in figure 3.6, in total 1024 pulses each revolution, and a pulse every 90° from the Index output.

In order to control the speed of the motor a square signal needs to be applied in the CLK pin, the speed of the motor is given by

\[ n \text{[rpm]} = \frac{f_{clk} \text{[Hz]}}{1024} \times 60, \]  
(3.8)

where \( f_{clk} \) is the frequency of the applied square signal. This is the only way for us of controlling the speed of the motor, opposite to the motors without control electronics where a varying voltage controls the speed, this in turn added the need to be able to generate a linearly varying frequency square signal.

### 3.2.2 Generating a linearly variable frequency

When selecting the motor, in the data sheet was specified that the motor could operate using a voltage between 2.7 and 5 V and that for achieving very good synchronization the clock signal could be fed with a continuous cycle. This to my understanding was that a variable voltage could be applied or a clock signal for stepper like functionality, so my idea was to apply a PWM signal for controlling the speed of the motor.

Luckily while in communication with our contact at MyMotors (Faulhaber division of micro drives) I commented this and he replied not to apply a PWM signal as I could damage the unit, instead one should apply a clock signal and that this was the way the motor had to be operated. At this point the idea was to use the 16-bit Timer in the ATMega128 µC and use an overflow service routine and toggle the output but soon, or maybe not soon enough I realized there was something which needed special attention, as we can see from

\[ f = \frac{1}{T}, \]  
(3.9)

having a constant \( \Delta T \) as it is the case in the timers of a µC would lead us to generate
a not proportional varying frequency, figure 3.7 depicts the needed period in seconds for a desired speed in rpm, as we can see the $\Delta T$ is not linearly varying and that is the reason the control signal could not be implemented using the timer peripheral with the previously mentioned strategy.

After a review of options for generating such a linearly variable square wave three ways of producing it were found, two analog and one digital way. The first analog way is with the use of an IC such as the MAX083 in which the output frequency is function of an input current which with the use of a resistor in series can be generated applying voltage (PWM from the $\mu$C). The second analog way is with the use of a Voltage to Frequency converter, such as the AD537, an advantage is that this technique is used in space for transmitting signals over long distances and there is a space qualified component in the frequency range of interest. The digital way is with the use of a technology called Direct Digital Synthesis DDS in which the information corresponding the shape of the wave is saved in memory on a look-up or wave table and then generated with the use of a DAC.

For our project the MAX038 IC was considered in the first place for apparently being simpler to implement, but it has several disadvantages: its operational temperature is very limited (0 to 70°C), its production is discontinued, which makes it very costly and eventually harder to find and its performance is highly temperature dependent. So it was discarded for this project.

The voltage to frequency converter approach is better than the MAX038 approach, there even exist a space qualified version and its functionality is very easy, a voltage is applied and a square signal of frequency proportional to this voltage is generated, one of its disadvantages is that it exhibits a non-linear response in a region of operation, shown in figure 3.8 on the following page [Ana00].

DDS is preferred over the analog IC’s for being more stable, having increased repeatability, price and there are IC’s with industrial version operational range which is in the range of commercially available RWA, however the implementation can be more complicated than say the V/F converter. One of its strong advantages is that it can be
used in a digital control loop giving very accurate results, also the IC in discussion the AD9833 provides us with a 28 bit resolution range, which allows for a huge spectrum of possible speeds $2.68435 \times 10^8$ to be exact, this method was chosen.

**AD9833**

The AD9833 is a DDS waveform generator that can divide its master frequency $f_{\text{Master}}$ in $2^{28}$ parts, and it’s controlled via a three wire serial interface, we use the SPI peripheral of the ATMega128 for this end. The AD9833 can generate sinusoidal, triangular and square wave outputs. The serial interface can be written up to a 40 MHz rate[Ana03], Let us take a look at the pin configuration in figure 3.9.

In order to be able to use the AD9833 we need a master frequency $f_{\text{Master}}$, which has to be the double of the highest frequency $f_{\text{Vout}}$ that we wish to generate (Nyquist Theorem), for generating $f_{\text{Master}}$ we use a ICM755 general purpose timer with the configuration shown in figure 3.10 on the facing page[Max92] where $R = 36.6 \, k\Omega$ and $C = 102.5 \, pF$ give us $f_{\text{Master}} \approx 187 \, kHz$, which translates in a $n_{\text{max}} \approx 5500 \, \text{rpm}$.

For the serial communication we use pins FSYNC, SDATA and SCLK. SDATA and SCLK go connected in the MOSI (Master-out, serial-in) and the SCK pins of the Crumb128 respectively, SCLK is a clock signal for synchronizing the communication and SDATA is the actual data.

The AD9833 uses 16-bit words for control registers, since the SPI routine of the com-
piler used (CodeVisionAVR) for programming the ATmega128 can only send and receive bytes (packages of 8-bits) we need to send two consecutively, the HIGH and LOW bytes, FSYNC is a control input that tells the AD9833 when we are to start and finish sending information, the settings for the SPI interface on the ATmega are: half cycle clock, AT-Mega is Master and MSB is sent first.

The first thing we need to do for using the AD9833 is to initialize and reset it, otherwise the information of the last time it was used remains in memory and is output causing undesired behavior, lines 516 to 528 of code in Appendix A do this job.

Data is sent between PORTA.1=0 and PORTA.1=1 which the reader might already have deduced refers to the FSYNC pin, lines 519 - 522 do the actual reset and to make sure the AD9833 will not output any residual data from previous operations we command to execute a wave with \( f = 0 \) Hz. The control register contains all the information about what type of wave is to be produced, if we are writing to the MSB or LSB or both, which oscillator to use and all other special functions, for more information refer to [Ana03].

3.2.3 Built-in encoder performance

Since the built-in encoder was enclosed and new to us there was little information about its performance and it seemed appropriate to do a comparison with real world performance so it was decided to set up an optical encoder to compare the performance of the integrated encoder.

Figure 3.11 on the next page shows the encoder wheel and the Vishay CNY70 photo interrupter, figure 3.12 on the following page shows the design of the encoder wheel, it is a modified design based in the postscript code found on [And01]. I added the cross and the white circle for making it easier for alignment. Figure 3.13 on page 27 shows output from CNY70 and figure 3.14 on page 28 shows signal from CNY70 after schmitt trigger

Figure 3.15 on page 29 shows a run at slow speeds, the speed was computed both from the quadrature signals from the integrated encoder and from the optical encoder, we can see how the integrated encoder shows some speed even when the input speed was zero, this is due to the high resolution encoder detects the small vibrations that exist and they
are counted by the $\mu$controller over the sampling period which was set to 50 $ms$, this is a quite large sampling period, but we should not forget that this is only monitoring of the speed and that the speed is controlled by the built-in controller. This behavior of always outputting pulses from the encoder even when at rest was one of the reasons to put the encoder to test.

Figure 3.16 on page 29 shows a run at higher speeds and shows clearly that the performance of the built-in encoder is consistent with the registered by the optical encoder, if the electronic version of this document is available, the reader is encouraged to zoom and notice the difference of the higher resolution built-in encoder against the low resolution optical one.

3.2.4 Current sensing

Because the torque produced by a BLDC (Brushless Direct Current) motor is proportional to its armature current as shown in figure 3.17 on page 30, a current sensor was added for enabling torque control by current feedback and not only by speed feedback, this feature is standard in commercially available reaction wheels. In fact reaction wheels often have at least two control modes: torque and speed, since hall-effect sensors are used predominantly it has been problematic to think about torque control using speed feedback, specially at low and zero crossings speeds, however recent findings by [GC06] suggest that using a 1024 count per revolution encoder speed feedback control scheme may lead to better results than with current control in reaction wheels specially in the
Figure 3.13: Output from Vishay CNY70 Opto-interrupter.

mentioned speed ranges.

MAX4172

Another reason to include a current sensor is simply for monitoring the behavior of the current as it is related to the torque output, the sensor used for monitoring the current is the MAX4172, in order to sense current a small resistor $R_{\text{SENSE}}$ is placed in the positive line going to the power source of the motor, the configuration used is shown in figure 3.18 on page 30 [Max96].

The maximum current to pass by the 2209 is 90 mA [Fau08], so our full range current to be measured bears this number in mind, with the guidelines found in [Max96] $R_{\text{SENSE}} = 1000 \text{ m}\Omega$ is suggested, two 500 m\(\Omega\) were put in series after measuring the resistance we have $R_{\text{SENSE}} = 1.25\Omega$, next we proceed to select $R_{\text{OUT}}$ depending on our full scale voltage and current, for the ATMega128 A/D converter this is 5V, $R_{\text{OUT}}$ is given by

$$R_{\text{OUT}} = \frac{V_{\text{OUT}}}{I_{\text{LOAD}} \times R_{\text{SENSE}} \times G_m}, \quad (3.10)$$

where $V_{\text{OUT}}$ is the full scale voltage, $I_{\text{LOAD}}$ is the full scale current being measured and $G_m = 10 \text{ [mA/V]}$ is the MAX4172 transconductance, $R_{\text{OUT}} = 3.88 \text{ k}\Omega$ is selected, this gives us a full scale voltage $V_{\text{OUT}} = 4.7 \text{ V}$.

Because the sensor is "seeing" the complete motor and not just its armature, the current used by the built-in electronics is also sensed, we will assume this current consumption is constant, however this is a downside because we might read more noise due to switching of the electronics and it is possible that the electronics energy consumption is not fully linear.

27
Figure 3.14: Output from Harris CD40106BE inverted Schmitt trigger.

3.3 System Overview

A diagram of the complete system is depicted in figure 3.19 on page 31 a torque mode based on current feedback would use the available current signal in top of the drawing. A picture of the suggested development system is shown in 3.20 on page 31.
Figure 3.15: Optical and built-in encoder reading in slow speeds.

Figure 3.16: Optical and built-in encoder reading in high speeds.
Figure 3.17: Torque/Current relation for a BLDC reaction wheel. *Credit, NASA/Ful69*.

Figure 3.18: MAX4172 functioning diagram and configuration.
Figure 3.19: System operation diagram.

Figure 3.20: Suggested system configuration for development.
Chapter 4

Controlling the RWA

4.1 Computing the input to the controller

Once the hardware development was ready it was time to write the code for the different maneuvers, figure 4.1 on the next page shows a flow chart for the torque function, the code can be found in Appendix C, (lines 232-292).

The torque function \texttt{torque(inptorque, tsec, torqdir)} receives three parameters: Magnitude, duration and direction. The main controller should take notice of the current speed for avoiding saturation, \texttt{inpspeed} is a global variable in the process that represents the speed, so the torque function only increases or decreases it, the function \texttt{speed(inpspeed)} (lines 220-229) receives \texttt{inpspeed} as parameter and calculates the MSB and LSB of the full range of speeds that the AD9833 can generate, finally the function \texttt{exec()} (lines 191-217) does a final formatting of the data to be sent to the AD9833 and sends or ”executes” it via the SPI interface. Because the variable \texttt{inpspeed} is a signed one, we take the sign of the variable and use it to command direction to the FH2209.

While the torque function is running, speeds are being calculated and commanded on a time basis of 10 ms or 100 Hz, figure 4.2 on page 34 shows a block diagram of the additional processes occurring while the torque function is running.

4.2 Dynamic response of the RWA

In order to characterize the response of a system, a common practice is to analyze the response to a step command, this is of great value when building the complete controller that would finally do the attitude control, from these graphics the parameters of a second order system can be deduced[Oga02].

For the acquisition of the step responses the data was saved in the SRAM memory of the \(\mu\)-Controller and then transmitted to the PC later, in this way the errors related to the delays caused by sending the information via UART while measuring speed were greatly minimized, however because of the limited memory space this could only be done for 200 samples, using a 50 ms rate gives a total of 10 seconds, enough for seeing high
Figure 4.1: Flow chart of the torque function.

quality step responses, but not for monitoring a complete maneuver of several minutes, when downloading the 200 samples and starting a set of 200 new samples at least 2 of the next samples which represents 100 ms were corrupted, thus for monitoring the maneuvers a continuous download of the data every 100 ms was used, however this method decreases the quality of the measured speed specially when close to max. speed, where more counts per sampling step are lost, but the data is still good enough for seeing the complete development of a maneuver.

4.2.1 Step response

The step response was calculated for the worst case input, being maximum speed $n = 5000$ rpm from rest. Apart from the speed step response, since the overshooting is really small a zoom of the overshooting area and settling time is included and the current response in mA as well.

In general the overshooting was 1% of the input and the final error oscillated 20 rpm below the input, this is from 4980 to 5000 rpm which represents 0.4% of the input. It’s normal that rise time and overshoot are a compromise, this small overshoot translates in very fine pointing capability. The slow response however may lead to instability of the system if not dealt with care in the controller phase design. Typically reaction wheels have lower response times, in the order of a few ms[Sid97], our responses vary from around 3 to 6.3 sec.
4.3 Viscous friction compensation by speed control

An important aspect to pay attention in a RWA is the internal disturbing friction torques, illustrated in figure 4.12 on page 40. These are non-linear, when using current control sophisticated control models need to be implemented to counteract its effects\cite{Sid97}, however recent work by \cite{GC06} suggests that speed control may lead to better results than with current control, specially at low and zero-crossing speeds, where the non-linearities are larger, this approach makes sense since torque is also a function of speed and not only proportional to the armature current of a motor, the setup used in their experiment uses an encoder also with 1024 counts per revolution, the key as mentioned in the paper is to be able to accurately measure and control speed.

To illustrate the viscous friction compensation we are talking about, figure 4.13 on page 40 shows a 600 second maneuver with braking and zero speed crossing, notice at around 30 seconds, before the maneuver is started we can see some current consumption, around 10mA, this correspond to the built-in electronics, then from 38 to 133 a torque is applied and we can see how the viscous friction torque increases with speed, coincid-
Figure 4.4: Zoom to the overshooting region of step response with small inertia flywheel.

If we were using current control these changes should be part of a non-linear mathematical model and its fidelity would translate in performance.

4.4 Stability

For the stability study we take the step response with the small wheel, which exhibits the lowest time constant and suffices with our requirements.

The analysis methodology is the following, the overshooting of the worst case step function at max. speed $n = 5000 \, \text{rpm}$ represents a sudden acceleration pushing the wheel out of its reference speed, which can be considered as an internal disturbing torque, we will approximate a linear angular acceleration to be able to measure the magnitude and measure the disturbing effects of such torque on the satellite, from figure 4.4, we approximate an angular acceleration as depicted in figure 4.14 on page 40, and extend its effects for 5 units of time (each unit 50 $ms$), this is more than it really acts, but as a safety factor.

Calculating the angular acceleration we obtain $52.36 \, \text{rad/sec}^2$, because $I_{FW} = 3.56 \times 10^{-7} \, Kg \cdot m^2$, the Torque transfered to the satellite is $T_o = 18.64 \times 10^{-6} \, Nm$.

Taking the inertia of the satellite $I_s = 0.025 \, Kg \cdot m^2$, results in an angular acceleration
\( \omega_s = 0.000746 \, \text{rad/sec}^2 \), if we suppose this torque acts for 250 ms which is more than the settling time, this would mean it would add to the satellite \( 1.865 \times 10^{-4} \, \text{rad/sec} \) of instability, which translates in \( 0.01068 \, \text{deg/sec} = 38.44 \, \text{arcsec/sec} \).

### 4.5 Pointing Accuracy

Since the stability requirement is met the pointing accuracy requirement should not be a problem, for this a maneuver which would be the equivalent of moving the satellite in increments of 0.1 \( \text{deg} \) in steps of 3 \( \text{sec} \), the maximum speed is set to 2000 \( \text{rpm} \) because its a fast maneuver, however we can see that its executed correctly in figure 4.15 on page 41 and the corresponding current in \( \text{mA} \) in figure 4.16 on page 41.
Figure 4.6: Step response with the medium inertia flywheel.

Figure 4.7: Zoom to the overshooting region of step response with medium inertia flywheel.
Figure 4.8: Current response to a step function with medium inertia flywheel.

Figure 4.9: Step response with the big inertia flywheel.
Figure 4.10: Zoom to the overshooting region of step response with big inertia flywheel.

Figure 4.11: Current response to a step function with big inertia flywheel.
Figure 4.12: Friction model of a reaction wheel. *Adapted from [Sid97].*

Figure 4.13: Maneuver showing current compensation to viscous friction.

Figure 4.14: Linearization of step response overshooting for calculating stability.
Figure 4.15: Maneuver composed of steps of 0.1 deg in 3 seconds each.

Figure 4.16: Current plot for the 0.1 deg steps in 3 sec maneuver.
Chapter 5

Testing

5.1 Functional Tests

5.1.1 Slew Maneuver

Minimum Target Maneuver

The first maneuver, \textit{maneuver 1} to present is the one specified as our minimal target maneuver, that is 180 deg in 600 sec, n was not specified in our target maneuver, in this example \(n=0.05\), which is a high value.

For deriving the required torque values many of the calculations that have been presented were used, on Apendix D, worksheets used to derive these calculations are presented.
Figure 5.1: Maneuver 1; Displacement of the flywheel in revolutions.

Figure 5.2: Maneuver 1; Speed of the flywheel in rpm.

Figure 5.3: Maneuver 1; Current of the RWA in mA.
Chapter 6

Conclusions

Important technical aspects for the development of a RWA for the use of Picosatellites were presented, a development platform was designed and tested successfully to meet the minimum performance criteria.

6.1 Performance

The controller presents almost no overshooting with a relatively slow response, which translates in very precise pointing capabilities and should perform well, fast targets and/or disturbance torques may be a problem due to the slow response of the drive, these should be modeled and see if it presents not a big drawback. Besides harsh environmental conditions, success with our minimum target maneuver of 180 deg in 600 sec and higher can be achieved with no apparent problem, as presented in the maneuver examples. A summary of the minimum requirement and the achieved is presented in table 6.1.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required</th>
<th>Achieved</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum maneuver</td>
<td>180 [deg] in 600 [min]</td>
<td>Yes</td>
</tr>
<tr>
<td>Pointing Accuracy</td>
<td>0.381 [deg]</td>
<td>0.1 [deg]</td>
</tr>
<tr>
<td>Stability</td>
<td>199.3 [arcsec]</td>
<td>38.44 [arcsec]</td>
</tr>
</tbody>
</table>

A summary of the RWA specifications, similar to the found in commercially available wheels, is presented in table 6.2 on the facing page.

6.2 Future Work

For the successful development of a RWA these are aspects which need particular attention:
Table 6.2: RWA technical specifications.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Capability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Angular Momentum at Max. Speed</td>
<td>mN-m-s</td>
<td>64, 129, 194</td>
</tr>
<tr>
<td>Output Torque, Max</td>
<td>uN-m</td>
<td>160</td>
</tr>
<tr>
<td>Peak Power</td>
<td>mW</td>
<td>450</td>
</tr>
<tr>
<td>Power Bus Voltage</td>
<td>Volts</td>
<td>5</td>
</tr>
<tr>
<td>Wheel Speed Max.</td>
<td>RPM</td>
<td>5000</td>
</tr>
<tr>
<td>Mass</td>
<td>gr</td>
<td>11.6, 13.9, 15.03</td>
</tr>
<tr>
<td>Outside Diameter. max</td>
<td>mm</td>
<td>22</td>
</tr>
<tr>
<td>Height</td>
<td>mm</td>
<td>18.5</td>
</tr>
<tr>
<td>Operational Temperature Range</td>
<td>deg C-Lo</td>
<td>-20</td>
</tr>
<tr>
<td></td>
<td>degC-Hi</td>
<td>85</td>
</tr>
<tr>
<td>Motor Type</td>
<td>DCBL</td>
<td></td>
</tr>
<tr>
<td>Interface</td>
<td>Analog/Digital</td>
<td>DIgital</td>
</tr>
</tbody>
</table>

First and most important, capable sensors to provide the high demands that were set, a star sensor and gyros capable of the accuracies higher than the here established.

If we are to use current feedback for torque control we must account for the viscous friction non-linear behavior, and have a feedback directly from the armature winding, since the ranges of currents are very small any additional electronics may induce undesired noise for control purposes.

It is important to measure the performance with a manufactured PCB, in the suggestions of the AD9833 it was pointed that a ground plane must be available, all the cables and manual soldering induce noises, and the performance should improve once all those details are taken care of.

6.2.1 Space Qualification

Special attention is needed in the area of space qualification, the materials and devices that were suggested were feasible within those specifications for a space qualified product in terms of temperature and vacuum since there exists a vacuum proof version of the 2209 DC drive. However system tests and, very crucial to this device, vibration test should be run, this will determine the final design, particularly if changing the inertia flywheel of place, as in the pancake shaped commercially available RWA’s.

6.2.2 Case

Together with the temperature and vacuum tests, the decision whether or not is necessary to include a case will be determined. Figure 6.1 on the next page shows an exploded view of the 2209, courtesy of Dr. Nienhaus, which helps us understand how the modifications could be realized.
6.2.3 Current Control

The approach presented uses the relation between angular acceleration and torque to achieve the torques required, other more commonly used approaches found in the literature use the relation between current and torque.

Since the DC drive used has a built-in controller it is not completely clear if the current consumed by the complete system is proportional to the delivered torque, plots of the current consumption were realized, even when the results are as expected, is not sure if this signal is clean enough as for being used for control purposes.

6.2.4 Collaboration with the manufacturer

The design of a RWA has proved not to be a trivial task, a joint collaboration between the University of Würzburg and Faulhaber Group would be very attractive in which it could be possible to optimize certain parameters for the specific use of the brushless drive as a reaction wheel, such as the torque noise and mechanical constant time. The 2209 is a an attractive platform for a future reaction wheel, however it has a relatively slow response, non typical for reaction wheels. The physical design, as mentioned earlier, would be subject of the shock tests, however if such collaboration would exist it would not be a problem to design a special case in which the inertia mas was enclosed, also the possibility to incorporate torque control using current as control input could be easily implemented.
Appendix A

Mechanical Drawings
SECTION A-A

UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH: TOOLS AND CLENS.
LINEAR ANGULAR

NAME: Angel Cano
SIGNATURE: 
DATE: 23.07.08

TITLE: Flywheel with:
Rei=9mm, Reo=10mm
Isw= 1.221 gr*cm^2

MATERIAL: Aluminum EN AW 2007

1.8999 gr

SCALE: 1:1
SHEET 1 OF 1

SolidWorks Student Edition for Academic Use Only
Appendix B

Schematics
Appendix C

Source code

In this section the source code for the program in the micro-controller is presented.

Pages 8 to middle of 12 is auto generated by the Codevision AVR Code Wizard, showing the settings for the micro-controller.
```c
#include <mega128.h>
#include <delay.h>
#include <stdio.h>
#include <math.h>
#include <spi.h>

// Declare your global variables here
unsigned char k; // keyboard control
unsigned int value, maskval;
unsigned int valuelsb, masklsb, maskvallsb;
unsigned char valueH, valueL;
unsigned char valuelsbH, valuelsbL;
unsigned long int tsec, i;
unsigned long int inptorque;
unsigned char valuelsbH, valuelsbL;
unsigned long int cnta, cnto, cntidx, totcntidx, totcnto, totcnta;
float nrpm, oldnrpm, tor, nrpmo, nrpmi, ts, inpspeedrpm, inpspeedradsec;
float current;
int torqdir;
unsigned long int timecontrol=0; // 10 ms timebase control variable
const unsigned int mask=0b0100000000000000;
const unsigned long int posdigrange=134217728, msbstep=16384;
const signed long int negdigrange=-134217728;

#define ADC_VREF_TYPE 0x40

// Read the AD conversion result
unsigned int read_adc(unsigned char adc_input)
{
    ADMUX=adc_input|ADC_VREF_TYPE;
    // Start the AD conversion
    ADCSRA|=0x40;
    // Wait for the AD conversion to complete
    while ((ADCSRA & 0x10)==0);
    // 10ms time base Timer 0 overflow interrupt service routine
    interrupt [TIM0_OVF] void timer0_ovf_isr(void)
```
{  
// Reinitialize Timer 0 value  
TCNT0=0x6F;  
// Place your code here  
if (manon==1){  
++timecontrol;  
if (timecontrol > 1){  
printf("WARNING !!! timecontrol %d",timecontrol);  
}  
}  
// end if  
// end timer 0 routine  
/*  
// Index External Interrupt 1 service routine  
interrupt [EXT_INT1] void ext_int1_isr(void)  
{  
++cntidx;  
} */  
// end index
  
/*  
// Qa External Interrupt 4 service routine  
interrupt [EXT_INT4] void ext_int4_isr(void)  
{  
++cnta;  
} */  
// end qa
  
/*  
// Optical External Interrupt 7 service routine  
interrupt [EXT_INT7] void ext_int7_isr(void)  
{  
++cnto;  
} */  
// end optical
  
float digtorpm(signed long int inpspeed){  
float speedrpm;  
speedrpm=((float)inpspeed*5500)/posdigrange;  
return (speedrpm);  
} // end dig to rpm  
float digtoradsec(signed long int inpspeed){
107: float speedradsec;
108: 109: speedradsec=digtorpm(inpspeed)*((2*PI)/(60));
110: 111: return (speedradsec);
112: } // end dig to radsec
113: 114: 115: // Timer 1 MEASURE SPEED
116: interrupt [TIM1_OVF] void timer1_ovf_isr(void)
117: {
118: // Reinitialize Timer 1 value
119: TCNT1H=0x7C;  //ts=0.146484375;
120: TCNT1L=0x29;
121: 122: TCNT1H=0xFF;
123: TCNT1L=0xFD;
124: 125: TCNT1H=0xD2;  // ts =0.05s
126: TCNT1L=0xFF;
127: 128: TCNT1H=0xFB; for ts 0.005
129: TCNT1L=0x7F;
130: 131: // Place your code here
132: 133: totcntidx=cntidx;
134: totcnta=cnta;
135: totcnto=cnto;
136: 137: current=((float)read_adc(0)*5)/1024)*(100/4.65);
138: 139: cntidx=0;
140: cnta=0;
141: cnto=0;
142: 143: 144: //ts=0.146484375;
145: //ts=0.008681;
146: 147: ts=0.05;
148: 149: nrpm=(totcnta*60)/(ts*1024);
150: 151: oldnrpm=nrpm;
152: 153: //tor=(nrpm-oldnrpm)/ts;
154: 155: }//measure speed
156: 157: 158: 159: // Timer 3 PRINT STUFF
interrupt [TIM3_OVF] void timer3_ovf_isr(void)
{
    // Reinitialize Timer 3 value
    //TCNT3H=0xE3;
    //TCNT3L=0xDF; // print every 500ms
    TCNT3H=0xFA;
    TCNT3L=0x5F; // print every 100ms
    TCNT3H=0xFF;
    TCNT3L=0x6F; // print every 10ms
    // Place your code here
    inpspeedrpm=digtorpm(inpspeed);
    inpspeedradsec=digtoradsec(inpspeed);
    //printf("a,%u, idx,%u, opt,%u,%f,RPMa,%f, RPMo, LSB,%u, MSB,%u,
    //inpRPM,%f , Rad/sec, %f \r
    //totcnta, totcntidx, totcnto, nrpm, nrpmo, valuelsb, value, inpspeedrpm, inpspeedradsec,\n    //printf("RPMa,%f,inpRPM,%f, tor, %f\r",nrpm,inpspeedrpm,tor);
    //printf("VEL, %f, Current, %f , input,%f\r" ,nrpm,current,inpspeedrpm);
    
    void exec(void){ // translates desired speed and sends to AD9833
        maskval=mask|value;
        maskvallsb=mask|valuelsb;
        valueL=maskval;
        valueH=maskval>>8;
        valuelsbL=maskvallsb;
        valuelsbH=maskvallsb>>8;
        // printf("\n |valueLSB %u |maskedvallsb %u | HEX2write %X%X
        //printf("\n |valueMSB %u |maskedval %u | HEX2write %X%X
        // printf("\n |valueLSB %u |maskedvallsb %u | HEX2write %X%X
        PORTA.1=0;
        spi(valuelsbH);spi(valuelsbL);//LSBs
        spi(valueH);spi(valueL);//MSBs
        PORTA.1=1; // finish writing to AD9833
        if(inpspeed >=0){
            PORTA.4=1;
        }
    }
else  {  
    FORTA.4=0;  
};  
// end exec function  

void speed(signed long int inpspeed){  // executes speed coming from a single number and not MSB and LSB
    if (inpspeed <=posdigrange && inpspeed >=negdigrange){
        value=labs(inpspeed)/16384;  
        valuelsb= labs(inpspeed)%16384;  
        exec();
        }  //end if
    }
// end speed  

void torque(unsigned long int inptorque, unsigned long int tsec , int torqdir){  // initial speed, tormagnitude, time, torque dir  
    unsigned long int tmsec,tentsmsec,i;  
    tmsec=tsec*1000;  
    tentsmsec=tmsec/10;  
    if (torqdir==1)  
    
        for(i=0;i<=tentsmsec;i++){
            waitingup:
            if (timecontrol==1)  
            
                inpspeed=inpspeed+inptorque;  
                //printf("i:%d of tentmsec: %d\r",i,tentsmsec);  
                speed(inpspeed);  
                timecontrol=0;  
                }  // end if
            else  
            
                goto waitingup;  
                }  //end else
        }  // end for hmsec
    manon=0;  
    };
//end if torque up  

- 5 -
if (torqdir==0) { // torque down
    manon=1;
    for (i=0;i<=tentsmsec;i++){
        waitingdown:
        if (timecontrol==1) {
            inpspeed=inpspeed-inptorque;
            //printf("i:%d of tentmsec: %d\r",i,tentsmsec);
            speed(inpspeed);
            timecontrol=0;
            }// end if
        else { goto waitingdown;
        };// end else
    }// end for hmsec
    manon=0;
};//end if torque up

}// end function torque

void notorque(unsigned long int tsec){ //
tmsec,tentsmsec,i;
tmsec=tsec*1000;
tentsmsec=tmsec/10;
manon=1;
for (i=0;i<=tentsmsec;i++){
    waitingnotorque:
    if (timecontrol==1) {
        timecontrol=0;
    } // end if
    else { goto waitingnotorque;
    };// end else
   }// end for hmsec
manon=0;
}// end function notorque
void main(void)
{
    // Declare your local variables here
    // Input/Output Ports initialization
    // Port A initialization
    // Func7=In Func6=Out Func5=Out Func4=Out Func3=In Func2=Out
    Func1=Out Func0=In
    // State7=T State6=0 State5=0 State4=0 State3=T State2=0 State1=0
    State0=T
    PORTA=0x00;
    // Port B initialization
    // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=Out Func1=Out
    Func0=Out
    // State7=T State6=T State5=T State4=T State3=T State2=0 State1=0
    State0=0
    PORTB=0x00;
    // Port C initialization
    // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
    Func0=In
    // State7=T State6=T State5=T State4=T State3=T State2=T State1=T
    State0=T
    PORTC=0x00;
    // Port D initialization
    // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
    Func0=In
    // State7=T State6=T State5=T State4=T State3=T State2=T State1=T
    State0=T
    PORTD=0x00;
    // Port E initialization
    // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
    Func0=In
    // State7=T State6=T State5=T State4=T State3=T State2=T State1=T
    State0=T
    PORTE=0x00;
    // Port F initialization
    // always quadrature output
    PORTF=0x00;
    DDRF=0x7f;
}
372: // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
373: Func0=In
374: // State7=T State6=T State5=T State4=T State3=T State2=T State1=T
375: State0=T
376: PORTF=0x00;
377: DDRF=0x00;
378:
379: // Port G initialization
380: // Func4=In Func3=In Func2=In Func1=In Func0=In
381: // State4=T State3=T State2=T State1=T State0=T
382: PORTG=0x00;
383: DDRG=0x00;
384:
385:
386: // Timer/Counter 0 initialization
387: // Clock source: System Clock
388: // Clock value: 14.400 kHz
389: // Mode: Normal top=FFh
390: // OC0 output: Disconnected
391: ASSR=0x00;
392: TCCR0=0x07;
393: TCNT0=0x6F;
394: OCR0=0x00;
395:
396: // Timer/Counter 1 initialization
397: // Clock source: System Clock
398: // Clock value: 230.400 kHz
399: // Mode: Normal top=FFFFh
400: // OC1A output: Discon.
401: // OC1B output: Discon.
402: // OC1C output: Discon.
403: // Noise Canceler: Off
404: // Input Capture on Falling Edge
405: // Timer 1 Overflow Interrupt: On
406: // Input Capture Interrupt: Off
407: // Compare A Match Interrupt: Off
408: // Compare B Match Interrupt: Off
409: // Compare C Match Interrupt: Off
410: TCCR1A=0x00;
411: TCCR1B=0x03;
412: TCNT1H=0x7C;
413: TCNT1L=0x29;
414: ICR1H=0x00;
415: ICR1L=0x00;
416: OCR1AH=0x00;
417: OCR1AL=0x00;
418: OCR1BH=0x00;
419: OCR1BL=0x00;
420: OCR1CH=0x00;
421: OCR1CL=0x00;
422:
423: // Timer/Counter 2 initialization
424: // Clock source: System Clock
425: // Clock value: Timer 2 Stopped
426: // Mode: Normal top=FFh
427: // OC2 output: Disconnected
428: TCCR2=0x00;
429: TCNT2=0x00;
430: OCR2=0x00;
431:
432: // Timer/Counter 3 initialization
433: // Clock source: System Clock
434: // Clock value: 14.400 kHz
435: // Mode: Normal top=FFFFh
436: // Noise Canceler: Off
437: // Input Capture on Falling Edge
438: // OC3A output: Discon.
439: // OC3B output: Discon.
440: // OC3C output: Discon.
441: // Timer 3 Overflow Interrupt: On
442: // Input Capture Interrupt: Off
443: // Compare A Match Interrupt: Off
444: // Compare B Match Interrupt: Off
445: // Compare C Match Interrupt: Off
446: TCCR3A=0x00;
447: TCCR3B=0x05;
448: TCNT3H=0xE3;
449: TCNT3L=0xDF;
450: ICR3H=0x00;
451: ICR3L=0x00;
452: OCR3AH=0x00;
453: OCR3AL=0x00;
454: OCR3BH=0x00;
455: OCR3BL=0x00;
456: OCR3CH=0x00;
457: OCR3CL=0x00;
458:
459: // Timer(s)/Counter(s) Interrupt(s) initialization
460: TIMSK=0x05;
461: E TIMSK=0x04;
462:
463: // External Interrupt(s) initialization
464: // INT0: Off
465: // INT1: Off
466: // INT2: Off
467: // INT3: Off
468: // INT4: On
469: // INT4 Mode: Rising Edge
470: // INT5: Off
471: // INT6: Off
472: // INT7: Off
473: EICRA=0x00;
474: EICRB=0x03;
475: EIMSK=0x10;
476: EIFR=0x10;
477:
// USART0 initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART0 Receiver: On
// USART0 Transmitter: On
// USART0 Mode: Asynchronous
// USART0 Baud rate: 115200
UCSR0A=0x00;
UCSR0B=0x18;
UCSR0C=0x06;
UBRR0H=0x00;
UBRR0L=0x07;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;

// ADC initialization
// ADC Clock frequency: 115.200 kHz
// ADC Voltage Reference: AVCC pin
ADMUX=ADC_VREF_TYPE;
ADCSRA=0x87;

// SPI initialization
// SPI Type: Master
// SPI Clock Rate: 3686.400 kHz
// SPI Clock Phase: Cycle Half
// SPI Clock Polarity: High
// SPI Data Order: MSB First
SPCR=0x58;
SPSR=0x00;

// Global enable interrupts
#asm("sei")

//reset AD9833

PORTA.1=0;
spi(0b00000001);//
spi(0b00000000);//
PORTA.1=1;
PORTA.1=0;
spi(0b00100000);spi(0b00000000);//MSB of 28, MSB first
spi(0b00000000);spi(0b00000000);//LSB of 28, MSB first
spi(0b00100000);spi(0b00000000);//control register
PORTA.1=1;
printf("done reset");
while (1)
{
k=getchar();

if (k=='m' && inpspeed <= posdigrange && inpspeed >= negdigrange) {
inpspeed=inpspeed+(msbstep*50);
speed(inpspeed);
} //end if

if (k=='p' && inpspeed <= posdigrange && inpspeed >= negdigrange) {
inpspeed=inpspeed+1000;
speed(inpspeed);
} //end if

if (k=='l' && inpspeed <= posdigrange && inpspeed >= negdigrange) {
inpspeed=inpspeed+50;
speed(inpspeed);
}

if (k=='j' && inpspeed <= posdigrange && inpspeed >= negdigrange) {
inpspeed=inpspeed+msbstep;
speed(inpspeed);
} // end if

if (k=='2') {
  PORTA.4=0;
  printf("Dir= %u",PORTA.4);
} //end if

if (k=='3') { // clockwise
  PORTA.4=1;
  printf("Dir= %u",PORTA.4);
} //end if

if (k=='d' && inpspeed >= (msbstep*50)) {
inpspeed=inpspeed-(msbstep*50);
speed(inpspeed);
} //end if
if (k=='e' && inpspeed >= 50) {
inpspeed = inpspeed - 50;
speed(inpspeed);
} // end if

if (k=='j' && inpspeed <= posdigrange && inpspeed >= negdigrange) {
inpspeed = inpspeed + msbstep;
speed(inpspeed);
} // end if

if (k=='i' && inpspeed <= posdigrange && inpspeed >= negdigrange) {
inpspeed = inpspeed + 1;
speed(inpspeed);
} // end if

if (k=='y') {
torque(24403, 20, 1);
notorque(30);
torque(24403, 20, 0);
} // end if

if (k=='z') {
inpspeed = 134217728;
speed(inpspeed);
} // end if

}; // end while(1)
%! Postscript utility for printing an encoder wheel
%
/inch {72 mul} def              % #points/inch (don't change me)
/size 0.5685 inch def                % radius of encoder wheel
/segments 20 def                % number of segments (black and white)
/angle 360 segments div def

/wedge
{ /radius exch def
/angle_s exch def
/angle_e exch def
newpath 0 0 moveto
0 0 radius angle_s angle_e arc
closepath
} def

gsave
1.0 inch 1.0 inch translate
0 1 segments {
  360 segments div rotate
  angle 0 size wedge
  2 mod 0 eq {1} {0} ifelse
  setgray fill
} for
grestore

showpage
Appendix D

Worksheets
<table>
<thead>
<tr>
<th>rBo</th>
<th>rs</th>
<th>hs</th>
<th>hb</th>
<th>density (g/mm^3)</th>
<th>vsb1</th>
<th>vsb2</th>
<th>vB</th>
<th>vE</th>
<th>vM</th>
<th>vFW</th>
<th>MASS</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.75</td>
<td>5</td>
<td>6</td>
<td>0.00285</td>
<td>53.99612</td>
<td>12.56637</td>
<td>66.56249</td>
<td>1602.212</td>
<td>201.0619</td>
<td>1869.837</td>
<td>5.329035</td>
</tr>
<tr>
<td>2</td>
<td>0.75</td>
<td>5</td>
<td>6</td>
<td>0.00285</td>
<td>53.99612</td>
<td>12.56637</td>
<td>66.56249</td>
<td>358.1416</td>
<td>241.9026</td>
<td>666.6076</td>
<td>1.899829</td>
</tr>
<tr>
<td>2</td>
<td>0.75</td>
<td>5</td>
<td>6</td>
<td>0.00285</td>
<td>53.99612</td>
<td>12.56637</td>
<td>66.56249</td>
<td>678.584</td>
<td>376.9911</td>
<td>1122.138</td>
<td>3.198092</td>
</tr>
<tr>
<td>2</td>
<td>0.75</td>
<td>5</td>
<td>6</td>
<td>0.00285</td>
<td>53.99612</td>
<td>12.56637</td>
<td>66.56249</td>
<td>824.6681</td>
<td>328.2964</td>
<td>1219.527</td>
<td>3.475652</td>
</tr>
<tr>
<td>2</td>
<td>0.75</td>
<td>5</td>
<td>6</td>
<td>0.00285</td>
<td>53.99612</td>
<td>12.56637</td>
<td>66.56249</td>
<td>961.3274</td>
<td>282.7433</td>
<td>1310.633</td>
<td>3.735305</td>
</tr>
<tr>
<td>2</td>
<td>0.75</td>
<td>5</td>
<td>6</td>
<td>0.00285</td>
<td>53.99612</td>
<td>12.56637</td>
<td>66.56249</td>
<td>1088.562</td>
<td>240.3318</td>
<td>1395.456</td>
<td>3.97705</td>
</tr>
<tr>
<td>2</td>
<td>0.75</td>
<td>5</td>
<td>6</td>
<td>0.00285</td>
<td>53.99612</td>
<td>12.56637</td>
<td>66.56249</td>
<td>1206.372</td>
<td>201.0619</td>
<td>1473.996</td>
<td>4.200889</td>
</tr>
<tr>
<td>2</td>
<td>0.75</td>
<td>5</td>
<td>6</td>
<td>0.00285</td>
<td>53.99612</td>
<td>12.56637</td>
<td>66.56249</td>
<td>1314.757</td>
<td>164.9336</td>
<td>1546.253</td>
<td>4.585887</td>
</tr>
<tr>
<td>2</td>
<td>0.75</td>
<td>5</td>
<td>6</td>
<td>0.00285</td>
<td>53.99612</td>
<td>12.56637</td>
<td>66.56249</td>
<td>1413.717</td>
<td>131.9469</td>
<td>1612.226</td>
<td>4.764962</td>
</tr>
<tr>
<td>2</td>
<td>0.75</td>
<td>5</td>
<td>6</td>
<td>0.00285</td>
<td>53.99612</td>
<td>12.56637</td>
<td>66.56249</td>
<td>1503.252</td>
<td>102.1018</td>
<td>1671.916</td>
<td>4.947717</td>
</tr>
<tr>
<td>2</td>
<td>0.75</td>
<td>5</td>
<td>6</td>
<td>0.00285</td>
<td>53.99612</td>
<td>12.56637</td>
<td>66.56249</td>
<td>1583.363</td>
<td>75.39822</td>
<td>1725.323</td>
<td>5.051475</td>
</tr>
<tr>
<td>2</td>
<td>0.75</td>
<td>5</td>
<td>6</td>
<td>0.00285</td>
<td>53.99612</td>
<td>12.56637</td>
<td>66.56249</td>
<td>1654.049</td>
<td>51.83628</td>
<td>1772.446</td>
<td>5.26636</td>
</tr>
<tr>
<td>2</td>
<td>0.75</td>
<td>5</td>
<td>6</td>
<td>0.00285</td>
<td>53.99612</td>
<td>12.56637</td>
<td>66.56249</td>
<td>1715.31</td>
<td>31.41593</td>
<td>1813.288</td>
<td>5.346942</td>
</tr>
<tr>
<td>2</td>
<td>0.75</td>
<td>5</td>
<td>6</td>
<td>0.00285</td>
<td>53.99612</td>
<td>12.56637</td>
<td>66.56249</td>
<td>1767.146</td>
<td>14.13717</td>
<td>1847.846</td>
<td>5.26636</td>
</tr>
<tr>
<td>2</td>
<td>0.75</td>
<td>5</td>
<td>6</td>
<td>0.00285</td>
<td>53.99612</td>
<td>12.56637</td>
<td>66.56249</td>
<td>1809.557</td>
<td>0</td>
<td>1876.12</td>
<td>5.346942</td>
</tr>
</tbody>
</table>

Worksheet for calculating Flywheel Inertia

Dimensions Volumes

<table>
<thead>
<tr>
<th>lvb1</th>
<th>lvb2</th>
<th>lvE</th>
<th>lvM</th>
<th>IFW</th>
<th>g*cm^2</th>
<th>kg*m^2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.351059</td>
<td>0.071628</td>
<td>358.4549</td>
<td>11.46053</td>
<td>370.3382</td>
<td>3.703382</td>
<td>3.70338E-07</td>
</tr>
<tr>
<td>0.351059</td>
<td>0.071628</td>
<td>92.37366</td>
<td>29.30046</td>
<td>122.0968</td>
<td>1.220968</td>
<td>1.22097E-07</td>
</tr>
<tr>
<td>0.351059</td>
<td>0.071628</td>
<td>158.5851</td>
<td>36.53044</td>
<td>195.5382</td>
<td>1.955382</td>
<td>1.95538E-07</td>
</tr>
<tr>
<td>0.351059</td>
<td>0.071628</td>
<td>183.6175</td>
<td>28.1863</td>
<td>212.2265</td>
<td>2.122265</td>
<td>2.12226E-07</td>
</tr>
<tr>
<td>0.351059</td>
<td>0.071628</td>
<td>204.1138</td>
<td>21.35419</td>
<td>225.8907</td>
<td>2.258907</td>
<td>2.25891E-07</td>
</tr>
<tr>
<td>0.351059</td>
<td>0.071628</td>
<td>220.6583</td>
<td>15.83937</td>
<td>236.9203</td>
<td>2.369203</td>
<td>2.3692E-07</td>
</tr>
<tr>
<td>0.351059</td>
<td>0.071628</td>
<td>233.7948</td>
<td>11.46053</td>
<td>245.678</td>
<td>2.45678</td>
<td>2.45678E-07</td>
</tr>
<tr>
<td>0.351059</td>
<td>0.071628</td>
<td>244.027</td>
<td>8.049791</td>
<td>252.4995</td>
<td>2.524995</td>
<td>2.52499E-07</td>
</tr>
<tr>
<td>0.351059</td>
<td>0.071628</td>
<td>251.8183</td>
<td>5.452705</td>
<td>257.6937</td>
<td>2.576937</td>
<td>2.57694E-07</td>
</tr>
<tr>
<td>0.351059</td>
<td>0.071628</td>
<td>257.5916</td>
<td>3.528254</td>
<td>261.5426</td>
<td>2.615426</td>
<td>2.61543E-07</td>
</tr>
<tr>
<td>0.351059</td>
<td>0.071628</td>
<td>261.7299</td>
<td>2.148849</td>
<td>264.3014</td>
<td>2.643014</td>
<td>2.64301E-07</td>
</tr>
<tr>
<td>0.351059</td>
<td>0.071628</td>
<td>264.5754</td>
<td>1.200334</td>
<td>266.1984</td>
<td>2.661984</td>
<td>2.66198E-07</td>
</tr>
<tr>
<td>0.351059</td>
<td>0.071628</td>
<td>266.4305</td>
<td>0.58198</td>
<td>267.4351</td>
<td>2.674351</td>
<td>2.67435E-07</td>
</tr>
<tr>
<td>0.351059</td>
<td>0.071628</td>
<td>268.1764</td>
<td>0</td>
<td>268.5991</td>
<td>2.685991</td>
<td>2.68599E-07</td>
</tr>
</tbody>
</table>
### Worksheet for calculating Maneuver parameters

<table>
<thead>
<tr>
<th>θ</th>
<th>rad</th>
<th>Total time (sec)</th>
<th>time (min)</th>
<th>Is</th>
<th>n</th>
<th>To</th>
</tr>
</thead>
<tbody>
<tr>
<td>180</td>
<td>3.141592654</td>
<td>600</td>
<td>10</td>
<td>0.025</td>
<td>0.05</td>
<td>4.59297E-06</td>
</tr>
<tr>
<td>180</td>
<td>3.141592654</td>
<td>600</td>
<td>10</td>
<td>0.025</td>
<td>0.1</td>
<td>2.42407E-06</td>
</tr>
<tr>
<td>180</td>
<td>3.141592654</td>
<td>600</td>
<td>10</td>
<td>0.025</td>
<td>0.15</td>
<td>1.71111E-06</td>
</tr>
<tr>
<td>180</td>
<td>3.141592654</td>
<td>600</td>
<td>10</td>
<td>0.025</td>
<td>0.2</td>
<td>1.36354E-06</td>
</tr>
<tr>
<td>180</td>
<td>3.141592654</td>
<td>600</td>
<td>10</td>
<td>0.025</td>
<td>0.25</td>
<td>9.5872E-07</td>
</tr>
<tr>
<td>180</td>
<td>3.141592654</td>
<td>600</td>
<td>10</td>
<td>0.025</td>
<td>0.3</td>
<td>9.09026E-07</td>
</tr>
<tr>
<td>180</td>
<td>3.141592654</td>
<td>600</td>
<td>10</td>
<td>0.025</td>
<td>0.35</td>
<td>8.81479E-07</td>
</tr>
</tbody>
</table>

Calculate necessary Output torque To

Determine lw and input parameters for microcontroller program

Compare needed inertia with calculated inertia

<table>
<thead>
<tr>
<th>Determine lw</th>
<th>n*totaltime</th>
<th>time in micro</th>
<th>micro</th>
<th>Max Speed (rpm)</th>
<th>Maxspeedmax</th>
<th>acceltorq</th>
<th>Maxspeed rad/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>540</td>
<td>3000</td>
<td>5000</td>
<td>122372108</td>
<td>40791</td>
<td>523.5987756</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>480</td>
<td>6000</td>
<td>5000</td>
<td>122372108</td>
<td>20395</td>
<td>523.5987756</td>
<td></td>
</tr>
<tr>
<td>90</td>
<td>420</td>
<td>9000</td>
<td>5000</td>
<td>122372108</td>
<td>13597</td>
<td>523.5987756</td>
<td></td>
</tr>
<tr>
<td>120</td>
<td>360</td>
<td>12000</td>
<td>5000</td>
<td>122372108</td>
<td>10198</td>
<td>523.5987756</td>
<td></td>
</tr>
<tr>
<td>150</td>
<td>300</td>
<td>15000</td>
<td>5000</td>
<td>122372108</td>
<td>8158</td>
<td>523.5987756</td>
<td></td>
</tr>
<tr>
<td>180</td>
<td>240</td>
<td>18000</td>
<td>5000</td>
<td>122372108</td>
<td>6798</td>
<td>523.5987756</td>
<td></td>
</tr>
<tr>
<td>210</td>
<td>180</td>
<td>21000</td>
<td>5000</td>
<td>122372108</td>
<td>5827</td>
<td>523.5987756</td>
<td></td>
</tr>
<tr>
<td>240</td>
<td>120</td>
<td>24000</td>
<td>5000</td>
<td>122372108</td>
<td>5099</td>
<td>523.5987756</td>
<td></td>
</tr>
<tr>
<td>270</td>
<td>60</td>
<td>27000</td>
<td>5000</td>
<td>122372108</td>
<td>4532</td>
<td>523.5987756</td>
<td></td>
</tr>
<tr>
<td>300</td>
<td>0</td>
<td>30000</td>
<td>5000</td>
<td>122372108</td>
<td>4079</td>
<td>523.5987756</td>
<td></td>
</tr>
</tbody>
</table>

Angular acceleration for reaching n of maneuver rad/s²

<table>
<thead>
<tr>
<th>a in rev/s²</th>
<th>lw/To/Accel</th>
<th>Rotor Inertia</th>
<th>Fw inertia</th>
<th>T inertia</th>
<th>STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>17.45329252</td>
<td>2.77777778</td>
<td>2.63158E-07</td>
<td>2.34E-07</td>
<td>1.22097E-07</td>
<td>3.56E-07</td>
</tr>
<tr>
<td>8.72664626</td>
<td>1.38888889</td>
<td>7.7778E-07</td>
<td>2.34E-07</td>
<td>1.22097E-07</td>
<td>3.56E-07</td>
</tr>
<tr>
<td>5.817764173</td>
<td>0.92592593</td>
<td>2.94118E-07</td>
<td>2.34E-07</td>
<td>1.22097E-07</td>
<td>3.56E-07</td>
</tr>
<tr>
<td>4.36332313</td>
<td>0.69444444</td>
<td>3.125E-07</td>
<td>2.34E-07</td>
<td>1.22097E-07</td>
<td>3.56E-07</td>
</tr>
<tr>
<td>3.490658504</td>
<td>0.55555556</td>
<td>3.3333E-07</td>
<td>2.34E-07</td>
<td>1.22097E-07</td>
<td>3.56E-07</td>
</tr>
<tr>
<td>2.90882087</td>
<td>0.46296296</td>
<td>3.57143E-07</td>
<td>2.34E-07</td>
<td>1.22097E-07</td>
<td>3.56E-07</td>
</tr>
<tr>
<td>2.49327503</td>
<td>0.3968254</td>
<td>3.84615E-07</td>
<td>2.34E-07</td>
<td>1.22097E-07</td>
<td>3.56E-07</td>
</tr>
<tr>
<td>2.18161565</td>
<td>0.34722222</td>
<td>4.16667E-07</td>
<td>2.34E-07</td>
<td>1.22097E-07</td>
<td>3.56E-07</td>
</tr>
<tr>
<td>1.939254724</td>
<td>0.30864198</td>
<td>4.54545E-07</td>
<td>2.34E-07</td>
<td>1.22097E-07</td>
<td>3.56E-07</td>
</tr>
<tr>
<td>1.745329252</td>
<td>0.27777778</td>
<td>0.0000005</td>
<td>2.34E-07</td>
<td>1.22097E-07</td>
<td>3.56E-07</td>
</tr>
</tbody>
</table>
Bibliography


