Wybrane problemy sterowania napędem wielosilnikowym

Selected problems of multi-motor drive control

AUTOR:
Marcin Dmochowski

PROWADZĄCY PRACĘ:
dr inż. Marek Wnuk, K-7

OCENA PRACY:
Pracę tę dedykuję moim drogim Rodzicom.
Pragnę złożyć serdeczne podziękowania Panu dr inż. Markowi Wnukowi za pomoc, wskazówki oraz okazaną wyrozumiałość.
# Contents

1 Introduction .................................. 6
   1.1 The importance of drive synchronisation ......... 6
   1.2 Approaches to the drive synchronisation issue .... 6
   1.3 The focus of this Master’s Thesis ............... 7

2 BLDC motor control theory ..................... 9
   2.1 Commutation in BLDC motors .................. 10
      2.1.1 Using Hall sensors or an encoder .......... 11
      2.1.2 Sensorless operation .................... 12
   2.2 Speed control ................................ 14
      2.2.1 Variations of PWM switching ............ 16
   2.3 Torque control ................................ 17

3 Motor driver’s overview ......................... 18
   3.1 Components .................................. 18
      3.1.1 Microcontroller .......................... 18
      3.1.2 N-MOSFET pre-driver .................... 19
      3.1.3 N-MOSFET transistors ................... 19
      3.1.4 Current measurement ..................... 20
      3.1.5 Magnetic encoders ........................ 20
   3.2 Schematics .................................. 20
   3.3 Required microcontroller’s peripherals ......... 25
   3.4 BLDC control algorithm ....................... 27

4 Implementation .................................. 28
   4.1 Printed circuit board .......................... 28
   4.2 Control algorithm ............................. 28
      4.2.1 Speed and current control loop .......... 30
      4.2.2 Speed synchronisation .................... 33

5 Research of driver’s capabilities ................. 35
   5.1 PI controller tuning ........................... 36
   5.2 Speed synchronisation ......................... 45

6 Conclusions .................................... 56

A Abstract in Polish .............................. 58
   A.1 Problem synchronizacji napędu ................ 58
   A.2 Teoria sterowania silników BLDC ............... 58
   A.3 Założenia projektowe dla sterownika .......... 60
A.4 Realizacja praktyczna ............................................. 60
A.5 Wyniki badań ....................................................... 61
A.6 Konkluzje ............................................................... 61

Literature 63
Chapter 1

Introduction

1.1 The importance of drive synchronisation

The issue of multi-motor drive synchronisation is ever present among numerous issues associated with the design of various robotic and mechanical systems. A prominent example to this is the IRB1400 manipulator. It is capable of achieving unidirectional pose repeatability of 0.05 mm at rated full load and $1 \frac{m}{s}$ velocity with all six robot axes in motion [1]. Such precision comes not only from its advanced control algorithms, but also from the fact that IRB1400’s electric motors are very well synchronised. While decent high-level control system allows such a manipulator to complete various pre-defined complicated tasks, good low-level drive control guarantees high precision of operation and is just as important.

An example where human safety depends on quality of the drive control system would be an electric car with four electric motors propelling each of the four wheels separately. Recently released Mercedes-Benz SLS AMG Electric Drive uses such a drive system with four synchronous motors [24]. In order to avoid undesirable slips in a situation where they should not be occurring (for example while driving in a straight line) the control system ought to be able to synchronise motor speeds with high precision. Avoiding the aforementioned slips improves the driver’s safety. It is worth noting that this issue is not present in cars with a single combustion engine, though such cars employ a gearwheel-based differential which brings other problems into consideration. They are described in section 1.2.

Another example where drive synchronisation is an important issue is a modern industrial production line in a plant dealing with continuous production of materials in form of a web, strip or fiber. In such an industrial plant the quality of final product greatly depends on the quality and precision of the implemented control system [18].

1.2 Approaches to the drive synchronisation issue

The traditional approach to position and speed synchronisation of multiple rotating shafts is coupling them together with a system of gears and linkages, used for example in some old-fashioned production lines and cars with a single combustion engine. This however causes several important problems to appear [8]:

- gearwheels are inefficient, noisy and greasy;
• the angular position error of a given shaft in relation to another coupled shaft increases (albeit marginally) as mechanical parts wear down;

• the layout of a drive system based on gears and linkages is difficult to change;

• maximal speed greatly depends on the quality of gearwheels.

A more sophisticated approach to the issue of position and speed synchronisation is using multiple motors and an electronic control system. Multi-motor drives are widely used in both robotic and mechanical systems. Examples include manipulators, mobile robots, modern production lines [18, 9], electric trains and cars [24]. Using multiple motors in lieu of a system consisting of gears and linkages rids the object under consideration of all problems associated with these. For this very reason electronic-controlled multi-motor drive systems are worth developing.

1.3 The focus of this Master’s Thesis

The purpose of this Master’s Thesis is to create a multi-motor drive control module and then to research its capabilities for drive synchronisation. In the aforementioned robotic and mechanical systems there are several types of drive synchronisation, which can be divided into the following categories:

• keeping two or more motors’ speeds at the same value (applies for example to production lines and electric vehicles);

• keeping speeds at values proportional to each other with pre-defined ratios (applies for example to manipulators);

• keeping a constant difference between two or more motors’ speeds (applies for example to quadcopters while manoeuvring);

• keeping torque values for two or more motors at the same value (applies for example to electric cars).

• keeping torque values at values proportional to each other with pre-defined ratios (applies for example to some of production lines).

A special case of a mechanism which belongs to the fourth category is a differential mechanism [8], which is a planetary gear train. Torque is transferred to each of the output shafts equally. Moreover, if the carrier train ratio is fixed at a value of 1, the average angular velocity of the two output shafts is equal to the speed of the propelling shaft. However, the most important thing is that the speed of the output shafts is inversely proportional to the resistance they encounter. It is worth noting that in more sophisticated differential mechanisms (for example in limited slip differential) torque disparity between output shafts is allowed to some extent.

The drive system constructed as a research subject consists of two brushless direct current (BLDC) motors with encoders. The research shall focus mainly on the system’s capabilities while operating according to the second of these categories, where speeds of both motors should remain proportional to each other at all times, and to the third category, where a constant difference between speed values should be kept.
The driver was designed with operation in a quadcopter in mind, therefore a complete control system consisting of two such control modules (each driving two BLDC motors) would be sufficient for such an application. It is however flexible enough that it can be also used in any other of aforementioned robotic or mechanical systems.
Chapter 2

BLDC motor control theory

A brushless direct current motor is in essence a three phase synchronous electric machine. The term direct current is in this case derived from the fact that no alternating current power source is required in order to power a BLDC motor along with its control unit, in contrast to synchronous motors and permanent magnet synchronous motors, which are for this reason called alternating current motors [28]. However, a power converter, specifically a voltage power inverter, is needed in order to control a BLDC motor. Inverters used with BLDC motors consist of three pairs of electronic switches (one pair per output phase) and create three voltage signals which are applied to the motor’s stator windings [13]. A simplified overview of a system consisting of a BLDC motor, a voltage power inverter and a control unit is shown in figure 2.1.

Figure 2.1 An overview of a system consisting of a BLDC motor, a voltage power inverter and a control unit (microcontroller). Source: [13].
2. BLDC motor control theory

2.1 Commutation in BLDC motors

Direct current motors are also characterised by the fact that commutation is required in their circuits for proper operation since it allows for creation of a rotation magnetic field \[23\]. In brushed direct current motors this is done using brushes. These however have several disadvantages. The most important two of these are as following:

- brushes get used up fairly quickly in relation to other mechanical parts, therefore the motor requires frequent maintenance;

- during the rotor’s rotation, brushes cause sparking due to periodically losing and gaining contact with segments of the motor’s commutator.

Sparking occurring in brushed direct current motors renders them unsuitable for applications in environments where fire or explosion hazard could be introduced by sparks. Brushless direct current motors can be used in such places as the commutation process in their case is done by switching the power inverter’s electronic switches (transistors or thyristors) on or off. For this reason BLDC motors are sometimes referred to as electronically commuted motors.

In case of a BLDC motor the voltage signals applied to its stator windings can be rectangular, which allows simple voltage power inverter design with six switching elements. By applying rectangular voltages a rotational magnetic field with low resulting torque ripple is created inside the motor’s air gap. This is achieved due to the fact that the permanent magnets’ placement on the rotor and their magnetisation are chosen in such a way that the voltages induced in stator windings during rotor’s movement (called back-EMF) are trapezoidal \[15\]. Example waveforms of stator windings voltage (electric potential at the beginning of a winding in relation to the potential in the windings’ common point) for all three phases are depicted in figure 2.2.

![Figure 2.2 Example waveforms of stator windings voltages during a full electrical revolution. Source: \[13\].](image)

In order to achieve a full revolution of the motor’s rotor, it is necessary to complete an amount of electrical revolutions equal to the number of motor’s pole-pairs per phase.
A cross-section of an example BLDC motor with three pole-pairs per phase is presented in figure 2.3. This motor’s rotor rotates by one third of a full revolution during one complete control cycle.

![BLDC Motor Cross-Section](image)

Figure 2.3 Cross-section of a BLDC motor with three pole-pairs per phase. Source: [10].

2.1.1 Using Hall sensors or an encoder

There are two approaches to ensure that phase voltages applied to stator windings are switched at correct moments. The simpler approach involves using sensors to detect rotor’s angular position. Both Hall effect sensors and an encoder are suitable for that task.

Three Hall effect sensors are required in order to obtain necessary information on rotor’s position. The control unit should react accordingly to signals provided by Hall effect sensors and apply appropriate voltage waveforms to stator windings. This ensures that the angle between rotor flux and stator flux remains as close to 90° as possible, therefore maintaining high torque value. In case of a BLDC motor this angle varies from 60° to 120°, which causes slight torque ripple [14]. Figure 2.4 contains a simplified overview for a motor with one pole-pair per phase powered by an inverter with six switching elements (thus providing six-step control, also referred to as six commutation stages). The arrows represent the direction of stator’s flux vectors in all six steps of rotor’s electrical revolution. Whenever the angle between the direction of rotor flux and the current direction of stator flux reaches 60°, the state indicated by one of the Hall effect sensors changes. The control unit should then switch voltages applied to stator windings accordingly. It is crucial that the voltage switch occurs precisely at this moment as otherwise torque ripple would increase. The angle between rotor flux and stator flux changes to a value of 120° with the voltage switch and the process repeats.
2. BLDC motor control theory

2.1.2 Sensorless operation

Back-EMF voltage (voltage induced in stator windings) fulfills a vital role in the motor’s commutation process if neither angular position sensors nor Hall sensors are present. At any given moment only two of the three stator windings are powered as shown in figure 2.2. Measuring the value of the back-EMF voltage in the unpowered winding allows the control unit to obtain some information on rotor’s position, specifically the moment when the angle between rotor flux and stator flux reaches 90°. At that moment the value of back-EMF voltage reaches half of supply voltage, which is equal to zero if the voltage supply is symmetrical and because of that this event is called a zero-crossing [12].
Figure 2.5 depicts control signals including phase selection signal which indicates if the presented phase’s back-EMF voltage is to be used in the zero-crossing event detection process, and also phase comparator output signal which changes value whenever a zero-crossing event is detected. Moreover, the phase voltage itself and a reference signal, which is voltage equal to half of supply voltage, are presented.

After detecting a zero-crossing event the control unit waits for the same amount of time that has elapsed since the start of current six-step control stage, and moves on to the next control step. This is represented as a $90^\circ$ shift between commutation and zero-crossing signals as presented in figure 2.5. Assuming that the rotor maintained its angular velocity at constant level, the angle between rotor flux and stator flux should have reached $60^\circ$ by that time.

Sensorless operation has both advantages and disadvantages over using Hall effect sensors or encoders.

- Using sensors of any kind implies using signal wires in addition to power wires, which would not be feasible in some cases.

- Sensors are more precise for multiple reasons. First of all, they do not require comparing an analog signal (phase voltage in unpowered phase) to another analog (reference) signal. They also can indicate the exact moment in which the control unit should move on to the next six-step control stage.

- If the rotor rapidly accelerates or decelerates, higher than nominal torque ripple occurs while operating without sensors. This is caused by the fact that the control unit waits for the exact amount of time that has elapsed between the start of current control stage and the zero-crossing event before moving on to the next control step. Therefore the voltage switching occurs too late or too soon while accelerating and decelerating respectively.
• If the rotor’s angular velocity is to be controlled using pulse width modulated control signals (which is the case in most simple BLDC motor systems), multiple false zero-crossing events can occur [10, 12]. This is further described in section 2.2.

• The back-EMF voltage induced in the motor’s unpowered winding reaches low levels when the rotor’s angular velocity is low. This encourages the use of sensors especially at low speeds. Another issue is associated with starting the motor, as in case of sensorless operation this has to be done with no information on the rotor’s position whatsoever. Start-up can be achieved by first aligning the rotor to some known position before attempting to use the usual six-step control.

2.2 Speed control

In direct current motors the rotor’s angular velocity is proportional to the average value of voltage applied to armature and inversely proportional to magnetic flux [23]. In BLDC motors the armature consists of stator windings and constant magnetic flux is created by rotor’s magnets, therefore in case of a BLDC motor the rotor’s angular velocity is proportional to the average value of phase voltage applied to stator windings. This fact provides two possibilities for speed control, which are:

• changing the value of voltage provided by power source, consequently changing the value of voltage applied to stator windings;

• using a constant voltage power source and employing PWM signals for switching the voltage power inverter’s electronic switches on and off during one control step, thus altering the average value of voltage applied to stator windings.

Both methods have their advantages and disadvantages. Using a variable voltage power source with a linear voltage regulator simplifies the design, although such a power source would be expensive and would suffer from high energy losses, making the whole system inefficient. For this reason altering the average voltage by using PWM signals is preferable. Figure 2.6 depicts waveforms of phase voltages and currents obtained while using PWM signals to switch the electronic switches on and off.

Using PWM switching increases the efficiency of the system as a whole, however it has several negative effects as well. First of all, switching transistors or thyristors, which function as electronic switches in power inverters, on and off leads to energy losses in these components. This occurs due to the fact that there is a short delay between applying voltage meant to switch on such a component and actually having it fully switched on [17]. During that transitional state the component is partially switched on, which in turn causes energy loss and heating up of the switching components. For this reason the frequency of the PWM switching signals cannot reach very high values. On the other hand, these signals cannot have too low frequency either, as their period must be much lower than the electromechanical constant of the motor in question [29]. Moreover, the higher the frequency of the PWM switching signal, the lower the motor’s additional torque ripple (introduced by PWM switching) becomes. Because of this finding a frequency which strikes a good balance between these adverse effects is a nontrivial task.

Another issue introduced with PWM switching signals is associated with sensorless commutation. Everytime a switching event in active phases occurs, the back-EMF voltage
Figure 2.6 Example waveforms of phase voltages and currents obtained while using PWM control signals. The switching components are named as shown in figure 2.1. Source: [13].

Figure 2.7 Phase voltage in unpowered phase (top), phase voltage in powered phase (middle) and zero-crossing event detection synchronisation signal (bottom). Source: [13].
in the unpowered phase gets disturbed [13], as shown in figure 2.7. The zero-crossing event detected using this method occurs at the moment marked with an arrow.

Due to the aforementioned disturbances, it is necessary to use an additional zero-crossing event detection synchronizing signal. During the high state of this signal, sampling of the back-EMF voltage is allowed as it gives the voltage in the unpowered phase time for settling down after a disturbance caused by switching the electronic switches associated with powered phases. This prevents the control unit from detecting a zero-crossing event prematurely.

### 2.2.1 Variations of PWM switching

There are several variants of PWM switching that can be employed with BLDC motors. First of all, PWM switching can be done unipolarly or bipolarly. Bipolar switching is better when employing sensorless commutation, although it causes higher electromagnetic emission of the motor than unipolar switching [12]. In this mode of operation both electronic switches that operate in a given control step get turned on and off simultaneously. While using unipolar variant, the bottom-side switch is turned on for the whole time of a given control step and the PWM signal is applied only the high-side switch.

PWM switching methods can also be divided with respect to presence of the ability to brake the motor electrically. If there is no need to provide the possibility of electrical braking, PWM switching can be done in noncomplementary mode. This means that during one PWM period only the switching elements that are supposed to be switched on are in fact turned on. For example during a control step with positive voltage applied to phase A and ground applied to phase B, only $S_{At}$ and $S_{Bb}$ switches are utilised (refer to figure 2.1). Figure 2.8 contains an overview of four quadrants of motor operation. Utilising noncomplementary switching provides the ability to operate in quadrants I and III, hence it is referred to as two-quadrant operation.

Utilising both the necessary electronic switches and their complements provides two major advantages. Following up on the example from the previous paragraph, this would mean either switching on $S_{Ab}$ and $S_{Bt}$ after $S_{At}$ and $S_{Bb}$ are switched off in a single PWM period in case of bipolar operation or, in case of unipolar operation, switching on $S_{Ab}$ after $S_{At}$ is switched off (providing that current PWM duty cycle is lower than 100%). Using this approach an average voltage value of 0 V is reached at 50% or 0% duty cycle in case of bipolar or unipolar operation respectively.

Complementary operation provides two major advantages, which are the possibility of braking the motor electrically by having it act as a generator (thus returning energy to the power source) and ensuring linear operation in respect to PWM duty cycle [12]. However, it also brings a disadvantage, which is the fact that dead time needs to be inserted between switching the high-side and low-side components in order to avoid short-circuits which would have fatal consequences. This complicates the control algorithm slightly, although advantages gained far outweigh this fact.
2.3 Torque control

A direct current motor’s torque is proportional to both the armature current and magnetic flux [23], therefore in case of a BLDC motor it is proportional to the average value of current flowing through stator windings, measured either separately for each phase or on the common direct current power wire leading to or from direct current power source.

As there is no direct method of controlling the current flow, torque controllers can limit the motor’s torque value by lowering the average value of phase voltage applied to stator windings. Refer to section 2.2 for average voltage value altering methods used with BLDC motors.

There are motors’ applications where operation with constant torque value regardless of speed is preferable [14]. As long as the motor’s torque is below the limiting value, the rotor’s speed is matched to the desired speed value by the speed controller. However, whenever the motor’s torque reaches the limiting value and would rise even further, the torque controller overrides the speed controller and lowers the rotor’s speed, thus keeping torque value at the limiting point.

Measuring the values of current for each phase also provides additional diagnostic information. If measured average values are not roughly equal across respective phases, the motor’s stator windings are most probably damaged. A more sophisticated version of this technique is employed as one of numerous safety measures for small synchronous generators in power plants [31].
Chapter 3

Motor driver’s overview

3.1 Components

Several assumptions were made before selecting the driver’s components and designing it. The most important are as following.

- The driver’s control unit must possess capabilities sufficient to simultaneously control two BLDC motors, receive and interpret quadrature signals from encoders for both motors, and measure values of three voltages.

- An encoder provides precise information on the rotor’s position, therefore the electronic commutation will be done using these sensors. Sensorless operation mode is considered optional.

- The control unit must be also equipped with two separate SPI communication interfaces and UART communication interface. Moreover, it needs to have general purpose input/output pins with external interrupt functionality in a quantity sufficient to send and receive all auxiliary signals to and from other components respectively.

- The voltage power inverter should not be a single component and should instead consist of a transistor driver and six transistors.

- The transistors’ resistance in switched on state should be as low as possible in order to avoid excess heating while simultaneously having relatively small dimensions. N-channel MOSFET transistors satisfy this criterion very well.

- Measurement of current drawn from direct current power source by the motors should be done as efficiently as possible, which would imply using an integrated circuit utilising Hall effect for current measurement. Such a solution requires no shunt resistors.

- The direct current power source for the driver and motors is a 3-cell Li-Poly battery.

All of the driver’s components were chosen taking these assumptions under consideration.

3.1.1 Microcontroller

The driver’s control unit is a Freescale Kinetis K60 microcontroller (MK60FX512VLQ15). The main reason behind this choice is the fact that there are two timers with quadrature decoder functionality and two eight-channel general purpose timers available, which
3. Motor driver’s overview

is enough for simultaneous control of two BLDC motors with encoders. The ARM Cortex-M4 core of this microcontroller possesses more than enough computational power. It is also equipped with three SPI modules, two I\(^2\)C modules, six UART modules, a USB interface and enough general purpose input/output pins. It operates at a supply voltage of 3.3 V \[20\].

3.1.2 N-MOSFET pre-driver

The task of control of the N-channel MOSFET transistors is fulfilled by Freescale MC33937, which is a three phase pre-driver specifically designed for three phase motor control and similar applications. It consists of three high side and three low side FET pre-drivers. Its most important features include \[21\]:

- operation at supply voltages (powering the inverter) in range of 8 V to 40 V;
- a guaranteed capability of gate drive at a current value of 1 A with overcurrent protection;
- a charge pump intended to allow full opening of high side transistors;
- built-in comparators for zero-crossing events detection, which is necessary in sensorless operation;
- a trickle charge pump used for sustaining gate voltage of high side transistors during long periods of having these transistors switched on (this charge pump provides current equal to leakage current);
- built-in amplifier for current measurement and overcurrent detection with selectable threshold;
- built-in logic introducing dead time between switching low and high side transistors of the same phase;
- SPI interface for setting various parameters, including length of dead time;
- accepting input signals at both 3.3 V and 5 V levels (output signals are provided at 5 V level);
- interrupt signals associated with fault detection and overcurrent detection.

All of these provided features make MC33937 a perfect candidate for a MOSFET driver in a BLDC motor driver. Its block diagram is presented in figure 3.1.

3.1.3 N-MOSFET transistors

The chosen transistors are AP1RA03GMT-HF-3 made by Advanced Power Electronics Corp. They are characterised by very low static drain-source resistance in switched on state, equal to 1.59 mΩ \[7\]. Their minimal guaranteed drain-source breakdown voltage is equal to 30 V, which should be a sufficient value for this project. They are also quite small, as their package is PowerSO8 and their dimensions are about 5 mm by 6 mm.
### 3.1.4 Current measurement

Current measurement is done by ACS712ELCTR-20A-T integrated circuits, which are Hall effect based linear current sensors. The optimised measurement range for these sensors is ±20 A. Their typical sensitivity in this range is equal to 100 mV/A. The typical resistance of the internal conductor is 1.2 mΩ [3]. This contributes to the main advantage of Hall effect current sensors over measuring methods based on shunt resistors, which is low pass-through resistance.

### 3.1.5 Magnetic encoders

The magnetic encoders used are ams (formerly Austria Microsystems) AS5040. Their resolution is 0.35°, which is an equivalent of 1024 points per rotor’s full rotation [4]. Two types of outputs are available: digital and incremental outputs. In the first group there is serial interface and PWM signal output. Incremental output can be set into either quadrature and index signals mode or step/direction and index signals mode. In this driver’s project only the quadrature mode is used. The encoder also has an output which provides information if the distance to the magnet is proper and if it is placed properly.

### 3.2 Schematics

The driver’s schematics were created using Altium Designer 13. They are presented in figures 3.2 to 3.7.

Each of the figures 3.2 and 3.3 contains one power inverter module along with a cur-
rent sensor. The Freescale Semiconductor MC33937 was not present in Altium Designer’s libraries. Its component library was therefore created manually with the following assumptions: digital and communication signals’ pins (which are connected to a signal bus leading to the microcontroller) should be kept on the left hand-side of the component, while analog signals’ pins and connections to the gates of N-MOSFET transistors should be kept on the right hand-side. Moreover, power supply and ground pins were placed on the top and bottom of the component respectively. Detailed descriptions of signals for this component are found in its data sheet [21].

The 10 Ω resistors inserted between the pre-drivers’ control outputs and the transistors’ gates are intended to limit the current flow while switching a transistor on or off, therefore reducing the electromagnetic emission in this part of the driver.

Current measurement is done by ACS712 Hall effect current sensors, which are inserted between the connected sources of low side N-MOSFET transistors and ground, thus providing measurement of current drawn from power source by the motor. Their analog outputs are connected to microcontroller’s pins which serve as analog-to-digital converter’s inputs and also to non-inverting input of MC33937’s current sense amplifier.
Figure 3.2 Schematics of the first complete power inverter with MC33937 FET pre-driver, ACS712 Hall effect current sensor and N-MOSFET transistor power stage.
Figure 3.3 Schematics of the second complete power inverter with MC33937 FET pre-driver, ACS712 Hall effect current sensor and N-MOSFET transistor power stage.
3. Motor driver’s overview

Figure 3.4 Schematics of connection headers for Bilbao minimodule [6], containing MK60FX512VLQ15 microcontroller.

Figure 3.5 contains schematics of a single encoder board and headers present on the driver’s board. The AS5040 encoders operate at a supply voltage of 3.3V and include an optional voltage regulator for systems with 5V supply voltage. The encoder provides quadrature signals and has a digital output which is in high state if the magnet on motor’s rotor is placed properly in relation to the encoder’s chip.

Figure 3.5 Schematics of a single encoder board along with two connection headers placed on the main board.
The driver board has connectors for SPI interfaces in master and slave mode and for UART interface. They provide an option of expanding the driver by peripheral boards and devices. Communication with a PC is done using the microcontroller’s built-in USB interface, which emulates a COM port.

Both the driver’s electronic components and motors are powered from a common direct current supply, which is a Li-Poly battery rated at 11.1 V. The voltage divider presented in figure 3.7 allows the microcontroller to measure the battery’s voltage. The voltage regulators used are linear regulators with low-dropout voltage from LM1117 series [22]. Aside from the 3.3 V voltage required as a power supply for most of the driver’s circuitry, there is a 5 V regulator powering the ACS712 current sensors.

### 3.3 Required microcontroller’s peripherals

Out of available microcontroller’s peripherals, these are necessary in order for the driver to operate properly:

- two general purpose FlexTimers (FTM0 and FTM3) along with 12 PWM channels (FTM0_C0V through FTM0_C5V and FTM3_C0V through FTM3_C5V for the first and the second timer respectively) for voltage waveforms generation through MC33937 FET pre-driver;
- two FlexTimers (FTM1 and FTM2) set up as QuadratureDecoders for interpreting quadrature signals from encoders;
3. Motor driver’s overview

Figure 3.7 Schematics of direct current power source connector and voltage regulators.

- three analog-to-digital converter’s inputs (ADC_SE9, ADC_SE12 and ADC_SE13) for current measurement in each motor’s power supply line (through ACS712 Hall effect current sensors) and for measurement of Li-Poly battery voltage;

- six general purpose input/output pins used as ExternalInterrupt inputs for overcurrent detection interrupt events, fault detection interrupt events handling and for zero position detection signals from the encoders;

- eight general purpose input/output pins used for handling remaining auxiliary signals;

- a FreeCntr periodic interrupt module (based on LPTMR0 timer module), used for generating an interrupt-driven PI control and speed calculation with a precise time base;

- AsynchroSerial communication interface, utilising UART0 peripheral;

- SynchroMaster communication interface, utilising SPI0 peripheral along with three general purpose input/output pins serving as chip select pins – two of these are used with MC33937 FET pre-drivers;

- SynchroSlave communication interface, utilising SPI1 peripheral and one chip select pin.
3.4 BLDC control algorithm

Two main controllers for each of the motors make up the internal structure of the driver. These are the speed controller and the torque controller. The internal structure of such a module for a single motor is presented in figure 3.8.

![Block Diagram of BLDC Motor Control](image)

**Figure 3.8** A block diagram representing the driver’s internal structure. Source: [14].

In case of a typical BLDC motor PI speed and torque controllers provide good quality of control [2, 19]. It is worth noting that also PID and fuzzy controllers can be used with BLDC motors with decent results [16, 27]. The derivative part of a PID controller can be skipped as in simpler cases it has minimal influence on the motor’s speed and PID controllers are harder to tune than PI controllers [26]. For this reason the driver utilises PI speed and torque controllers.

As the block diagram of the driver’s internal structure suggests, the speed controller is subordinate to the torque controller. The reason for this is that the torque controller acts only as a limiter to the motor’s torque value. As long as the motor’s torque value stays under the limiting value, the only thing influencing the motor’s speed is the speed controller. In such case the rotor should rotate at desired speed. However, if the motor’s torque value reaches the limiting value and then its load keeps increasing, the torque controller steps in and reduces the rotor’s speed. The speed controller is no longer able to increase the speed as long as the torque value stays at the preset limit. Both speed and torque controllers change the duty cycle of PWM switching signals accordingly. Speed and torque control theory in BLDC motors has been described in sections 2.2 and 2.3. The control technique intended for this driver is four quadrant operation with unipolar switching.
Chapter 4

Implementation

4.1 Printed circuit board

The printed circuit board created using aforementioned schematics is presented in figures 4.1 and 4.2. The assembled driver is presented in figures 4.3 and 4.4.

![Printed circuit board](image)

Figure 4.1 Front view of the driver’s printed circuit board.

4.2 Control algorithm

The driver’s software was created using Freescale CodeWarrior for MCU 10.6 and Processor Expert 10.6. CodeWarrior is an integrated development environment aiding in creation of microcontroller’s software and Processor Expert can automatically generate peripheral initialisation code and numerous functions associated with their operation. This greatly simplifies design of software for Freescale embedded systems.
The implemented control algorithm’s flowcharts are presented in figures 4.5 and 4.6. The driver first initialises all peripherals. Most of the initialisation code was generated by Processor Expert using peripherals’ selected settings. After initialisation the microcontroller measures power source’s voltage and checks if the magnets on motors’ axles are placed correctly in relation to the encoders’ chips. Then it and attempts to set up the MC33937 FET...
pre-drivers through SPI interface. Another part of the initialisation procedure involves the motors’ alignment. For a short moment the motors are powered with voltages specific for one of the six-step control stages. This should cause the rotors to align themselves in such a way that their flux vectors are parallel to stator flux vectors created by applied voltages. At this point the quadrature decoders’ counters are set to zero. This allows the driver to use encoders for motors’ commutation as $0^\circ$ electric angle is now equivalent to zero in quadrature decoders’ counters.

4.2.1 Speed and current control loop

Prior to entering the speed and current control loop, the driver starts the motors and enables commutation interrupts. Such an interrupt is executed every time a motor’s rotor changes its electric angle by $60^\circ$, which is an equivalent of rotating by $\frac{60^\circ}{p}$, where $p$ is the number of motor’s pole-pairs per phase. In this driver the commutation interrupts are based on quadrature decoders’ counter overflow event, as their maximal value is set to $\frac{n6p}{p}$, where $n$ is equal to the amount of encoders’ pulses per full rotation of the rotor. If $n$ is not divisible by $6p$, the maximal counter values must be changed during a single rotor’s turn. This must be done in such a way that the quadrature decoders generate $6p$ interrupts during a full rotor’s turn and their counter registers contain a zero at rotor’s angular position of $0^\circ$. An occurrence of a commutation interrupt causes the driver to move on to the next stage of six-step control and to increment a variable which indicates the amount of quadrature decoders’ counter overflow events.

During operation in speed and current control loop, which is executed every 10 ms, the driver first checks if there is a new set of measurements available from the analog-to-digital converter. These are timed in such a way that in every execution of the loop there
should be new measurements available. Every current measurement is an average of 128 ADC conversion results due to the fact that PWM switching introduces a lot of noise into the current waveforms and the current flow can even become noncontinuous at low duty cycle values. The driver then checks if any of the motors draws current of a value exceeding the preset limiting value. If any motor meets that condition, it is marked using a variable. Then current error value is calculated as well, basing on the limiting value and the measurement.

The driver also has a hard limit on current values drawn by the motors. It is based on interrupt signals that are generated by the FET pre-drivers whenever the current exceeds the allowed maximal value and the driver stops the offending motor immediately.

The next step of operation is calculating the motors’ speeds. This is done by obtaining the number of pulses generated by the encoders since the start of the previous execution of the control loop. It is calculated for a single motor as the amount of its quadrature decoder’s counter overflow events multiplied by the amount of its encoder’s pulses per full rotation of the rotor and divided by six times the number of motor’s pole-pairs per phase.
This can be represented by equation 4.1.

\[ EP(k) = OF(k) \cdot \frac{n}{6p} \]  

(4.1)

\( OF(k) \) is the amount of quadrature decoder’s counter overflow events in loop iteration \( k \), \( n \) is equal to the amount of encoder’s pulses per full rotation of the rotor and \( p \) is the number of the motor’s pole-pairs per phase.

For better precision the value stored in the quadrature decoder’s counter register is added to this result and the previously read value is subtracted from it. The obtained result is then divided by the amount of encoder’s pulses per full rotation, thus giving the amount of rotations done since the previous control loop’s iteration. This number is then multiplied by the frequency of executing the control loop and by 60 in order to get a precise result expressed in \( \text{min}^{-1} \). The complete speed calculation is presented in equation 4.2.

\[ \omega(k) = \frac{60}{n \cdot T_B} \cdot (OF(k) \cdot \frac{n}{6p} + FTM(k) - FTM(k-1)) \]  

(4.2)

\( FTM(k) \) is the value contained in quadrature decoder’s counter register in iteration \( k \) and \( T_B \) is the time base.

The duty cycle of output PWM signals directed to the pre-drivers’ inputs is calculated basing on the information gathered and the controllers’ tuning parameters, which are proportional gain \( K_p \) and integral gain \( K_i \). The \( K_i \) parameter can be also interpreted as proportional gain divided by integral time \( T_i \), thus representing the controller in standard form, as shown in equation 4.3.

\[ u(t) = K_p \cdot e(t) + K_i \cdot \int_0^t e(\tau) \, d\tau = K_p \left( e(t) + \frac{1}{T_i} \cdot \int_0^t e(\tau) \, d\tau \right) \]  

(4.3)

\( u(t) \) is the input to the process, which in this case is the motor, or rather its FET pre-driver and voltage power inverter, while \( e(t) = SP - PV \) is the process error, which is equal to process value subtracted from the set point. The integral part of the sum is limited to the minimal and maximal value of the \( u(t) \) signal. Equation 4.3 represents a continuous time model, however the driver uses a discrete time model with a time base of 10 ms, which can be described using equation 4.4.

\[ u(k) = K_p \left( e(k) + \frac{T_B}{T_i} \sum_{i=0}^{k} e(k) \right) \]  

(4.4)

\( T_B \) is equal to the driver’s time base, 0.01 s in this case. In every control loop appropriately scaled \( u(k) \) is applied as the new PWM duty cycle.

At the end of a control loop, the gathered information on speed and current values is sent to a computer through USB interface with COM port emulation. The driver then reads new desired speed and current values (if any are given; otherwise they remain the same) and starts over by waiting for an interrupt from the PIT (periodic interrupt timer) module, after which it moves on with the next execution of the presented control loop.
4. Implementation

4.2.2 Speed synchronisation

Speed synchronisation of multiple motors is the purpose of this driver, therefore set points of speed may change in the middle of a control loop. This is independent of optional changes of the desired speed and current values done by the driver’s user. The reason for this is that if one of the motors draws current of a value exceeding the preset limiting value, the torque controller takes over in case of this motor, however the remaining motors may still operate under the supervision of speed controller. In such a case the set point speed values of remaining motors are calculated basing on the speed of the motor which is being controlled by the torque controller, taking into account the synchronisation variant. This allows the driver to maintain the restrictions imposed by the chosen synchronisation variant while simultaneously ensuring proper speed and torque controllers’ operation and overcurrent protection.
Figure 4.6 Flowchart representing the speed and current control loop which is a part of the driver’s control algorithm.
Chapter 5

Research of driver’s capabilities

The research was conducted using the assembled driver and two Tower Pro Outrunner Bm2410-09 BLDC motors, which feature seven pole-pairs per phase and can draw a maximal current of 8.4 A (effective value). They feature a motor velocity constant of $840 \text{ min}^{-1} \text{V}^{-1}$, effectively allowing them to reach a top speed of $9240 \text{ min}^{-1}$ [30] while their voltage power inverter is powered from a 11 V DC source. It should be noted that these motors ran without any load whatsoever, therefore their shafts’ moment of inertia was very low.

The only difference between aforementioned driver and the hardware used in the research was in changing the BLDC pre-drivers to JETI ECO 18 [25] in lieu of the MC33937, as unfortunately all attempts of communication with the intended pre-drivers failed. Moreover, they failed to provide correct control signals despite feeding proper PWM signals to their inputs. The oscillogram presented in figure 5.1 contains a record of the voltage applied to the gate of low side transistor belonging to power inverter for motor 1 in phase B. This part of the driver operates correctly as this waveform is in accordance with the PWM signal applied to the MC33937’s input. Then two identical PWM signals, complementary to the signal applied to the low side input of phase B, were applied to both high side and low side inputs of phase C. In order to force current flow between phases B and C, a 220 $\Omega$ resistor was connected to these outputs. The pre-driver’s logic inverts the signals applied to high side inputs, therefore the high side transistor of phase C should be switched on in the same moments as the low side transistor of phase B, and the low side transistor of phase C should be switched on when these two are switched off. This however was not the case as proven by the oscillogram presented in figure 5.2. The low side transistor of phase C was in fact never in conducting state as the gate voltage of high side transistor remains at a level of at least 12 V at all times. Additionally, this voltage was not boosted to the level that would allow entering a fully switched on state. The assumption that the low side transistor of phase C was not turned on was then confirmed by the fact that the voltage on its gate was equal to 0 V at all times. This leads to the conclusion that these pre-drivers must have been overheated or damaged in some other way.

The research was done in three main parts, which are:

- tuning of PI speed and current controllers,
- research of capabilities for speed synchronisation while keeping speeds at values proportional to each other with pre-defined ratios,
5. Research of driver’s capabilities

The parameters obtained during the controllers tuning were used in further parts of the research.

During the driver’s operation, speed and current data was collected through the USB interface with COM port emulation. It was then plotted using MATLAB.

5.1 PI controller tuning

The first part of the research entails tuning of the driver’s PI speed controller. The predetermined parameter sets include proportional gain $K_p \in \{0.1, 0.25, 0.5, 1\}$ and integral time $T_i \in \{0.1, 0.5, 1, 5, 10, 50\}$. The parameters were tested using a varying speed set point of 6000 min$^{-1}$ for the first 3.5 seconds, then 2500 min$^{-1}$ for the next 3.5 seconds and again 6000 min$^{-1}$ for the remaining 3 seconds of the 10 s test. The speed of the second motor was set to be 1.25 times greater than the speed of the first motor. Selected plots of motors’ speeds that are representative of the controller’s behaviour with various $K_p$ and $T_i$ values are presented in figures 5.3 to 5.17. No plot with $K_p = 1$ is presented due to the fact that high proportional gain values caused severe speed oscillations. Total error value was calculated for each of the parameter sets using integration as presented in equation 5.1.

$$e = \int_0^{10} |SP(t) - \omega(t)| \, dt$$  \hspace{1cm} (5.1)
5. Research of driver’s capabilities

$SP(t)$ is the speed set point value in time moment $t$. The approximate value of error was calculated using MATLAB’s trapezoidal numerical integration function `trapz`. Tables 5.1 and 5.2 contain total error values for all of the $K_p$ and $T_i$ values taken under consideration for the first and second motor respectively.

![Graph](image-url)

Figure 5.3 Speed waveforms acquired with a varying speed set point and $K_p = 0.1$, $T_i = 0.1$.

![Graph](image-url)

Figure 5.4 Speed waveforms acquired with a varying speed set point and $K_p = 0.1$, $T_i = 0.5$.
Figure 5.5  Speed waveforms acquired with a varying speed set point and $K_p = 0.1, T_i = 1$.

Figure 5.6  Speed waveforms acquired with a varying speed set point and $K_p = 0.1, T_i = 5$. 
Figure 5.7 Speed waveforms acquired with a varying speed set point and $K_p = 0.25$, $T_i = 0.1$.

Figure 5.8 Speed waveforms acquired with a varying speed set point and $K_p = 0.25$, $T_i = 0.5$. 
Figure 5.9  Speed waveforms acquired with a varying speed set point and $K_p = 0.25$, $T_i = 1$.

Figure 5.10  Speed waveforms acquired with a varying speed set point and $K_p = 0.25$, $T_i = 5$. 
Figure 5.11  Speed waveforms acquired with a varying speed set point and $K_p = 0.25$, $T_i = 10$.

Figure 5.12  Speed waveforms acquired with a varying speed set point and $K_p = 0.5$, $T_i = 0.1$. 
5. Research of driver’s capabilities

Figure 5.13  Speed waveforms acquired with a varying speed set point and $K_p = 0.5$, $T_i = 0.5$.

Figure 5.14  Speed waveforms acquired with a varying speed set point and $K_p = 0.5$, $T_i = 1$. 
5. Research of driver’s capabilities

Figure 5.15  Speed waveforms acquired with a varying speed set point and $K_p = 0.5$, $T_i = 5$.

Figure 5.16  Speed waveforms acquired with a varying speed set point and $K_p = 0.5$, $T_i = 10$. 
Figure 5.17  Speed waveforms acquired with a varying speed set point and $K_p = 0.5$, $T_i = 50$.

Table 5.1  Total error values for speed tracking for the first motor with various values of $K_p$ and $T_i$ parameters.

<table>
<thead>
<tr>
<th>$K_p$</th>
<th>0.1</th>
<th>0.5</th>
<th>1</th>
<th>5</th>
<th>10</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>3824</td>
<td>4768</td>
<td>4632</td>
<td>4180</td>
<td>4590</td>
<td>5074</td>
</tr>
<tr>
<td>0.25</td>
<td>3397</td>
<td>3866</td>
<td>4671</td>
<td>4129</td>
<td>3664</td>
<td>3916</td>
</tr>
<tr>
<td>0.5</td>
<td>4678</td>
<td>4025</td>
<td>4830</td>
<td>3888</td>
<td>3519</td>
<td>3810</td>
</tr>
<tr>
<td>1</td>
<td>4431</td>
<td>4521</td>
<td>4464</td>
<td>4586</td>
<td>4459</td>
<td>4100</td>
</tr>
</tbody>
</table>

Table 5.2  Total error values for speed tracking for the second motor with various values of $K_p$ and $T_i$ parameters.

<table>
<thead>
<tr>
<th>$K_p$</th>
<th>0.1</th>
<th>0.5</th>
<th>1</th>
<th>5</th>
<th>10</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>5084</td>
<td>6262</td>
<td>5561</td>
<td>6632</td>
<td>6982</td>
<td>7563</td>
</tr>
<tr>
<td>0.25</td>
<td>4764</td>
<td>5208</td>
<td>5680</td>
<td>5278</td>
<td>4942</td>
<td>5490</td>
</tr>
<tr>
<td>0.5</td>
<td>5147</td>
<td>5233</td>
<td>5649</td>
<td>5886</td>
<td>4942</td>
<td>5129</td>
</tr>
<tr>
<td>1</td>
<td>5236</td>
<td>5317</td>
<td>5551</td>
<td>5615</td>
<td>5526</td>
<td>5605</td>
</tr>
</tbody>
</table>

It is apparent that increasing $K_p$ too much causes speed oscillations of an increasing amplitude. On the other hand, too low values of this parameter cause the speed value
not to reach the desired set point, especially at higher $T_i$ values. Low values of $T_i$ provide decent quality of control at low proportional gain values, however they also cause high overshoot. Increasing $T_i$ to high levels will also have negative impact on the time that the controller needs to achieve speed error values close to zero. For that reason a compromise between the amplitude of oscillations, overshoot and settling time must be achieved when choosing PI controller’s parameters.

After additional fine-tuning, the values determined as providing decent quality of control were $K_p = 0.45$ and $T_i = 10$. The reason behind that choice is total error values are low and the obtained waveforms appear to be the best with these parameters from all of the sets tested. The total error values were equal to 3369 and 4590 for the first and second motor respectively. Speed waveforms acquired using these parameters are presented in figure 5.18.

![Figure 5.18](image)

**Figure 5.18** Speed waveforms acquired with a varying speed set point and $K_p = 0.45$, $T_i = 10$.

### 5.2 Speed synchronisation

The speed synchronisation research was done in two aforementioned separate variants with $K_p = 0.45$ and $T_i = 10$. The first involved keeping two motors’ speed values proportional to each other. In the second, a constant speed difference was to be kept between the motors.

In the first part the ratio of speeds’ set points was set to 0.8, calculated as the first motor’s speed divided by the second motor’s speed, therefore the second motor’s rotor should rotate 1.25 times faster. The driver’s and motors’ behaviour with a variable speed
set point in form of a positive and negative ramp is shown in figures 5.19 and 5.21 respectively. The positive ramp starts at a set point of 2500 min\(^{-1}\) for the first motor and climbs up to 6500 min\(^{-1}\) with a change of 4 min\(^{-1}\) every 10 ms, while the negative ramp is the exact reverse. Figures 5.20 and 5.22 contain calculated ratio values and ratio error. The first 0.5 s of speed values were ignored in the ratio calculation as these values bear no significance due to accelerating the motors from a complete halt. Maximum and average absolute values of errors are presented in table 5.3.

Another kind of waveform brought into consideration was a stair-like waveform. The initial speed set point of 2500 min\(^{-1}\) for the first motor was increased by 1000 min\(^{-1}\) every 2 s. The achieved results and error values are presented in figures 5.23, 5.24 and in table 5.2 respectively.

In addition to researching the capabilities of keeping a constant ratio between speed values, the driver’s behaviour was also tested with a various desired ratio in form of a positive and negative ramp, starting at 0.5 and changing by 0.001 every 10 ms to reach 1.5 at 10 s and vice versa. The speed set point was defined as 5000 min\(^{-1}\) for the first motor. Plots of acquired waveforms and calculated ratio are presented in figures 5.25 and 5.27. The relative error values for this variant were calculated as maximum or average value over the whole 10 s from the absolute values of error values divided by desired ratio in a given moment, as presented in equations 5.2 and 5.3.

\[
e_{\text{max}} = \max \left\{ \frac{|e(k)|}{rt(k)}, \; k = 1, \ldots, n \right\}
\]

(5.2)

\[
e_{\text{avg}} = \frac{1}{n} \sum_{k=1}^{n} \frac{|e(k)|}{rt(k)}
\]

(5.3)

\(e(k)\) is the ratio error in control loop iteration \(k\), \(rt(k)\) is the desired ratio in iteration \(k\), \(n = \frac{10 \text{s}}{0.01 \text{s}} = 1000\) is the amount of data points.
5. Research of driver’s capabilities

Figure 5.19  Speed waveforms acquired with a varying speed set point (positive ramp) and a constant speed ratio of 0.8; $K_p = 0.45$, $T_i = 10$.

Figure 5.20  Calculated ratio of motors’ speeds along with ratio error during the positive speed ramp test; $K_p = 0.45$, $T_i = 10$. 
5. Research of driver’s capabilities

Figure 5.21 Speed waveforms acquired with a varying speed set point (negative ramp) and a constant speed ratio of 0.8; $K_p = 0.45$, $T_i = 10$.

Figure 5.22 Calculated ratio of motors’ speeds along with ratio error during the negative speed ramp test; $K_p = 0.45$, $T_i = 10$. 
Figure 5.23 Speed waveforms acquired with a varying speed set point (stair-like waveform) and a constant speed ratio of 0.8; $K_p = 0.45$, $T_i = 10$.

Figure 5.24 Calculated ratio of motors’ speeds along with ratio error during the stair-like waveform test; $K_p = 0.45$, $T_i = 10$. 
Figure 5.25  Speed waveforms acquired with a constant speed set point for the first motor and a varying speed ratio (positive ramp); $K_p = 0.45$, $T_i = 10$.

Figure 5.26  Calculated ratio of motors' speeds along with ratio error during the positive ratio ramp test; $K_p = 0.45$, $T_i = 10$. 
Figure 5.27 Speed waveforms acquired with a constant speed set point for the first motor and a varying speed ratio (negative ramp); $K_p = 0.45$, $T_i = 10$.

Figure 5.28 Calculated ratio of motors’ speeds along with ratio error during the negative ratio ramp test; $K_p = 0.45$, $T_i = 10$. 
5. Research of driver’s capabilities

Table 5.3 Ratio error values for all tested variants and $K_p = 0.45$, $T_i = 10$.

<table>
<thead>
<tr>
<th>waveform type (desired ratio: 0.8)</th>
<th>positive ramp</th>
<th>negative ramp</th>
<th>stairs-like</th>
<th>variable ratio: positive ramp</th>
<th>variable ratio: negative ramp</th>
</tr>
</thead>
<tbody>
<tr>
<td>maximum absolute value of error</td>
<td>0.120</td>
<td>0.092</td>
<td>0.158</td>
<td>0.180</td>
<td>0.251</td>
</tr>
<tr>
<td>average absolute value of error</td>
<td>0.021</td>
<td>0.018</td>
<td>0.024</td>
<td>0.034</td>
<td>0.044</td>
</tr>
<tr>
<td>maximum absolute value of error relative to desired ratio</td>
<td>15.03%</td>
<td>11.54%</td>
<td>19.81%</td>
<td>17.73%</td>
<td>17.16%</td>
</tr>
<tr>
<td>average absolute value of error relative to desired ratio</td>
<td>2.67%</td>
<td>2.24%</td>
<td>2.99%</td>
<td>3.69%</td>
<td>3.95%</td>
</tr>
</tbody>
</table>

All of the waveforms used to research the capabilities for synchronisation with a desired speed ratio were also used in testing the driver using the other synchronisation variant, which is keeping a constant difference between motors’ speeds. The chosen difference value was 1000 min$^{-1}$ added to the speed of the second motor in comparison to the first. Figures 5.29 to 5.34 contain obtained speed waveforms and error plots. The maximum and average absolute value of speed difference error is presented in Table 5.2.

![Figure 5.29](image.png)  
Figure 5.29 Speed waveforms acquired with a varying speed set point (positive ramp) and a constant speed difference of 1000 min$^{-1}$; $K_p = 0.45$, $T_i = 10$.  

![Figure 5.30](image.png)
Figure 5.30 Calculated speed difference of motors along with speed difference error during the positive speed ramp test; $K_p = 0.45$, $T_i = 10$.

Figure 5.31 Speed waveforms acquired with a varying speed set point (negative ramp) and a constant speed difference of 1000 min$^{-1}$; $K_p = 0.45$, $T_i = 10$. 
Figure 5.32: Calculated speed difference of motors along with speed difference error during the negative speed ramp test $K_p = 0.45$, $T_i = 10$.

Table 5.4: Speed difference error values for all tested variants and $K_p = 0.45$, $T_i = 10$.

<table>
<thead>
<tr>
<th>Waveform type (desired speed difference: 1000 min$^{-1}$)</th>
<th>Positive ramp</th>
<th>Negative ramp</th>
<th>Stairs-like</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum absolute value of error [min$^{-1}$]</td>
<td>623</td>
<td>652</td>
<td>756</td>
</tr>
<tr>
<td>Average absolute value of error [min$^{-1}$]</td>
<td>137</td>
<td>128</td>
<td>149</td>
</tr>
<tr>
<td>Maximum absolute value of error relative to desired value</td>
<td>62.3%</td>
<td>65.2%</td>
<td>75.6%</td>
</tr>
<tr>
<td>Average absolute value of error relative to desired value</td>
<td>13.7%</td>
<td>12.8%</td>
<td>14.9%</td>
</tr>
</tbody>
</table>
Figure 5.33 Speed waveforms acquired with a varying speed set point (stair-like waveform) and a constant speed difference of 1000 min\(^{-1}\); \(K_p = 0.45\), \(T_i = 10\).

Figure 5.34 Calculated speed difference of motors along with speed difference error during the stair-like waveform test; \(K_p = 0.45\), \(T_i = 10\).
Chapter 6
Conclusions

- BLDC motors find an ever increasing usage in various applications due to their energy efficiency and reliability. Moreover, these motors provide the possibility of speed and (albeit limited) position control without additional wires and integrated circuits through sensorless operation. These advantages along with provided fire safety make them worth researching and developing further. However, they are not without disadvantages, as they are more difficult to control than, for example, other direct current motors and in a typical case require a voltage power inverter, which can be quite expensive.

- There are applications which demand the possibility of changing the motor’s speed not only through altering the average value of voltage applied to stator windings, but also through other means. An example of such an application is an electric train. BLDC motors with rotor consisting of a winding powered from a DC source instead of a permanent magnet would find their use in such cases [5]. Changing the voltage applied to such winding would alter the flux strength, thus changing the rotor’s angular velocity.

- As the research results indicate it is possible to create a solution for multi-motor drive synchronisation using relatively cheap components and uncomplicated control algorithms with decent results. However, tests of only one of two chosen synchronisation variants (keeping the ratio between motors’ speeds at a desired value) had satisfactory results. The average absolute value of error in relation to the set point did not exceed 4% for all tested waveforms in case of ratio-based synchronisation.

- There can be two possible causes of high relative error values obtained during tests of the other variant, which was keeping constant speed difference. First of all, the tests were conducted with a desired speed difference value which is relatively small in comparison to the motors’ maximal speed. Increasing the desired speed difference twice would effectively halve the calculated relative error values. Secondly, the motors were running without any load whatsoever, therefore their shafts’ moment of inertia was very low. This fact, along with imprecise PI controllers tuning, causes oscillations of a quite high amplitude to appear in the speed waveforms. In light of these issues the driver performed quite well. This situation can however be improved by further tuning of PI controllers or attaching load to the motors’ shafts. Another possible reason of control inaccuracies is using cheap motors of rather low quality and replacement pre-drivers/inverters. Even the differences between two motors of the same kind are noticeable as proven by low and high amplitude of speed
oscillations observed during PI controllers tuning for the first and second motor respectively.

- An example application of the assembled driver in tandem with another identical driver would be a complete control unit for a quadcopter. Using the ratio-based synchronisation variant would allow for precise roll, pitch and yaw control, thus assuring following proper flight direction. The necessary hardware modifications of the driver would entail adding connectors for sensors, such as an accelerometer, a gyroscope, a magnetometer and an absolute pressure sensor. These sensors are usually combined into 10 degrees of freedom IMUs (inertial measurement units). Distance measuring sensors would also need to be added.

- The driver’s capabilities can be further researched and most probably improved by implementing control algorithms other than PI for speed and torque controllers. This would also allow for comparing different control algorithms’ behaviour in motor control applications. Among other control algorithms often used in applications involving motor control the most prominent examples are PID controllers, fuzzy logic controllers and hybrid PID-fuzzy controllers [9].
Appendix A

Abstract in Polish

A.1 Problem synchronizacji napędu


W celu zapewnienia synchronizacji kilku silników w napędzie, można zastosować jedno z dwóch podejść. Pierwszą opcją jest użycie systemu przekładni oraz wałów. Jest to jednak rozwiązanie wysoce nieefektywne z wielu względów [8]. Znacznie lepszym wyborem jest użycie elektronicznego systemu zarządzającego pracą poszczególnych silników w napędzie.

Celem niniejszej pracy jest utworzenie modułu sterującego dla napędu wielosilnikowego oraz przebadanie jego możliwości. Na rozważany napęd składają się dwa bezszczotkowe silniki prądu stałego (BLDC) z enkoderami. Został on stworzony z zamysłem umieszczenia w quadcopterze (wraz z drugim identycznym systemem utworzona zostanie kompletna jednostka napędowa quadcoptera). Synchronizacja prędkości obrotowej przy uwzględnieniu ograniczenia maksymalnej wartości momentu obrotowego może przebiegać na kilka sposobów. Tworzony sterownik będzie w zamyśle mógł posłużyć do synchronizacji prędkości obrotowej z utrzymywaniem stałego stosunku prędkości bądź stałej różnicy prędkości poszczególnych silników względem pozostałych.

A.2 Teoria sterowania silników BLDC

Silniki bezszczotkowe prądu stałego należą do grupy maszyn elektrycznych trójfazowych synchronicznych. W celu uruchomienia i sterowania silnikiem BLDC niezbędny jest falownik napięcia, składający się z trzech par zaworów, który poda odpowiednie przebiegi napięciowe na uzwojenia stojącego silnika. Falownik zasilany jest źródłem napięcia stałego. Schemat poglądowy kompletnego systemu sterowania silnika został zamieszczony na rysunku 2.1.

Silniki bezszczotkowe należą również do grupy silników prądu stałego i wymagają przeprowadzania komutacji w ich obwodach w celu wytworzenia wirującego pola magnetycznego generującego moment obrotowy [23]. W silnikach szczotkowych jest to dokonywane przy
pomocy szczotek, które jednak mają swoje wady, na przykład iskrzenie. W silnikach BLDC dokonywana jest komutacja elektroniczna poprzez zmienianie aktywnych par elektronicznych kluczy falownika napięcia zasilającego silnik. Przebiegi napięć fazowych podczas jednego pełnego obrotu elektrycznego przedstawiono na rysunku 2.2.

Komutacja musi być przeprowadzana w ściśle określonych położeniach kątowych wirnika. Największy moment obrotowy jest wytwarzany, kiedy wektory strumienia wirnika i stojana są położone względem siebie pod kątem 90°, jednak w silnikach BLDC kąt ten w stanie normalnej pracy zmienia się w zakresie 60° do 120°, przez co powstają pulsacje momentu obrotowego [14]. Oznacza to równocześnie, że komutacja musi być przeprowadzona co każde 60° elektrycznych. Implikuje to konieczność posiadania informacji o położeniu wirnika w celu zapewnienia poprawnej pracy silnika. Informację tę można pozyskać poprzez wykorzystanie czujników Halla, enkoderów albo pomiar wartości napięcia indukowanego w niezasilanej (w danym momencie) fazie silnika. Napięcie to jest związane z położeniem wirnika, gdyż jest ono praktycznie równe połowie napięcia zasilania w momencie, gdy w danym kroku sterowania osiągany jest kąt 90° między wektorami strumieni wirnika i stojana, przez co przy pracy beczcznikowej po wykryciu przejścia napięcia przez ten poziom wystarczy odczekać długość czasu, która upłynęła od początku danego kroku, po czym przejść do następnego kroku. Dokonywanie komutacji przy pomocy sensorów ma wiele zalet nad pracą beczcznikową, wadą tego rozwiązania jest jednak konieczność doprowadzenia przewodów do sensorów.


Sterowanie PWM ma kilka wariantów, z których każdy ma swoje dobre i złe strony. W zrealizowanym projekcie sterownika skupiono się na sterowaniu unipolarnym komplementarnym, które pozwala na pracę silnika w dwóch kierunkach w obydwu czterech kwadrantach (praca silnikowa lub prądnicowa). Drugą zaletą tego wariantu jest liniowość (względem współczynnika wypełnienia sygnałów PWM) zmiana średniej wartości napięcia przyłożonego do uzwojeń silnika [12].

W silnikach prądu stałego moment obrotowy jest wprost proporcjonalny do prądu przepływającego przez twornik, w przypadku silnika BLDC są to uzwojenia stojana [23]. Maksymalny moment obrotowy silnika można ograniczyć przez ograniczenie maksymalnej wartości prądu stojana. Wystarczający w tym celu jest pomiar prądu płynącego ze źródła napięcia.
A.3 Założenia projektowe dla sterownika

Przy doborze elementów składowych sterownika postawiono kilka kluczowych warunków. Mikrokontroler musi posiadać wystarczającą ilość peryferiów, to jest dwa dekodery kwadraturowe, dwa moduły typu timer z sześcioma kanałami PWM każdy, dwa interfejsy SPI, interfejs USB oraz liczbę wejść/wyjść ogólnego przeznaczenia wystarczającą do obsługi pozostałych sygnałów.

Na ostateczny wybór elementów składają się:

- mikrokontroler Freescale Kinetis K60, z zapasem spełniający postawione jednostce centralnej wymagania;
- dwa sterowniki tranzystorów N-MOSFET MC33937, stworzone z myślą o sterowaniu silnikami i maszynami trójfazowymi;
- tranzystory N-MOSFET AP1RA03GMT, cechujące się niską rezystancją dren-źródło w stanie przewodzenia;
- sensory prądu wykorzystujące efekt Halla ACS712ELCTR-20A-T;
- enkodery magnetyczne AS5040, mające rozdzielczość 1024 impulsów na pełen obrót.

Schematy elektryczne sterownika znajdują się na rysunkach 3.2 do 3.7, natomiast lista wykorzystywanych peryferiów mikrokontrolera została zamieszczona w podrozdziale 3.3.

W sterowniku struktura regulatorów prędkości obrotowej oraz momentu obrotowego jest typowa dla sterowników silników i stanowi kaskadę ze sterownikiem momentu jako jednostką nadrzędną. W takiej sytuacji z pewnością nie dojdzie do większych przekroczeń zadanej maksymalnej wartości prądu. Schemat blokowy sterownika znajduje się na rysunku 3.8.

A.4 Realizacja praktyczna

Wgląd gotowej płytki z obwodami przedstawiono na rysunkach 4.3 do 4.2. Schematy przepływu sterowania w oprogramowaniu znajdują się na rysunkach 4.5 oraz 4.6. Sterownik po uruchomieniu podejmuje czynności inicjalizacyjne, takie jak mierzenie napięcia zasilania, ustawienie sterowników tranzystorów poprzez SPI oraz wstępne ustawienie wirników silników i zresetowanie dekoderów kwadraturowych. Po uruchomieniu silników sterownik wchodzi do pętli sterowania prędkością obrotową i momentem obrotowym, której każda iteracja jest uruchamiana co 10 ms. Przechodzenie do kolejnych kroków sterowania (dokonywanie komutacji elektronicznej) jest przeprowadzane w przerwaniu generowanym przy przepełnieniu liczników dekoderów kwadraturowych, które zostały ustawione w taki sposób, aby przerwania te generowane były co 60° elektrycznych. Sterownik we wspomnianej pętli odczytuje wyniki pomiarów konwertera analogowo-cyfrowego, oblicza prędkość obrotową, po czym zadaje nowe wartości współczynnika wypełnienia dla sygnałów PWM (przy uwzględnieniu faktu, że regulator momentu jest nadrzędnym), posługując się algorytmem regulatora PI. Jako że celem sterownika jest synchronizacja prędkości obrotowych silników, w przypadku wykrycia przekroczenia ograniczenia pradowego przez którykolwiek z silników, zadane wartości prędkości obrotowej są korygowane dla pozostałych silników. Punktem odniesienia dla korekty jest bieżąca prędkość...
silnika, który osiągnął ograniczenie prądowe. Pod koniec każdego wywołania pętli informacje o bieżącej prędkości obrotowej i wyniki pomiarów konwertera analogowo-cyfrowego są przesyłane do komputera poprzez interfejs USB (z emulacją portu COM).

A.5 Wyniki badań

Ze względu na problem z układami MC33937, które prawdopodobnie uległy przegrzaniu podczas lutowania, tranzystory dolnej strony w jednej z faz nie były załączone, co w efekcie uniemożliwiło realizację badań z wykorzystaniem tych układów. Zamiast nich zastosowano sterowniki modelarskie JETI ECO 18.

Pierwszym etapem badań był dobór nastaw dla regulatorów PI. Przebadano zachowanie się zestawu z przebiegiem prędkości zadanej w postaci ciągu \((6000 \text{ min}^{-1}, 2500 \text{ min}^{-1}, 6000 \text{ min}^{-1})\) ze zmianą co 3,5 s (całkowity czas trwania pojedynczej serii to 10 s). Uzyskane przebiegi prędkości zamieszczono w podrozdziale 5.1. Dobrane według kryterium całkowitej wartości nastaw wyniosły \(K_p = 0,45\) oraz \(T_i = 10\), takie też wartości przyjęto dla dalszych rozważań.

Następnie sprawdzono zdolność sterownika do synchronizacji prędkości obrotowej silników. W przypadku wariantu, w którym prędkości obrotowe miały być do siebie proporcjonalne, średnia wartość modułu błędu względnego wyniosła w najgorszym przypadku niecałe 4%. Maksymalna wartość również nie była bardzo wysoka. W przypadku drugiego wariantu, w którym między prędkościami silników miała być utrzymywana stała różnica, błąd różnicy prędkości osiągał znaczne wartości. Wszystkie rezultaty badań włącznie z uzyskанныmi wartościami błędów regulacji zamieszczono w podrozdziale 5.2.

A.6 Konkluzje

• Silniki BLDC są coraz chętniej wykorzystywane w rozmaitych robotycznych i mechanicznych systemach dzięki swoim liczным zaletom. Do ich wad zaliczyć należy skomplikowaną (w porównaniu do typowych silników prądu stałego) procedurę sterowania oraz konieczność użycia fałownika napędu, który może być drogi przy dużych wartościach mocy.

• Silniki BLDC z uzwojeniem wirnika zamiast stałego magnesu można z powodzeniem wykorzystać w zastosowaniach, gdzie bardziej pożądane jest sterować prędkością przez zmianę natężenia strumienia \([29]\), na przykład w pociągach elektrycznych.

• Zgodnie z wynikami badań, stworzenie efektywnego systemu służącego do synchronizacji prędkości obrotowej do wartości proporcjonalnych względem siebie nie musi wymagać drogich komponentów ani skomplikowanych algorytmów sterowania.

• Uzyskane przy drugim wariancie synchronizacji wartości błędów są znaczne z dwóch powodów: wybrana pożądana różnica prędkości była stosunkowo niewielka w porównaniu do prędkości osiąganych przez silnik oraz silniki pracowały praktycznie bez obciążenia (co oznacza mały moment bezwładności na wale silników oraz pojawianie się oscylacji prędkości). Dodatkową przyczyną mógł być fakt, że wykorzystane silniki są tanie i kiepskiej jakości (ewidentna była różnica między pracą silników przy dobieraniu parametrów regulatora prędkości). Dokładniejszy dobór nastaw bądź
użycie lepszej jakości silników powinno umożliwić lepsze zrealizowanie tego rodzaju synchronizacji.

- Przykładowym zastosowaniem stworzonego sterownika w tandemie z drugą identyczną jednostką jest układ kontrolny quadcoptera. Zastosowanie wariantu synchronizacji opartego na proporcjonalności prędkości względem siebie pozwoliłoby na precyzyjne sterowanie kierunkiem lotu.

- Możliwości sterownika mogą najprawdopodobniej zostać ulepszone przez wprowadzenie innych algorytmów sterowania dla regulatorów prędkości i momentu, jak na przykład popularne w sterowaniu silnikami regulatory PID, fuzzy oraz hybrydowe PID-fuzzy [9].
Bibliography


