The University Of Sheffield.

Department Of Mechanical Engineering

MEng Mechanical Engineering

The Spherical Revolution:

Development and Promotion of an
Open-source & Arduino-based Spherical Drive System
in the Maker Community

Alex DUNNETT

May, 2020

Supervisor: David Polson

Thesis submitted to the University of Sheffield in partial fulfilment of the requirements for the degree of Master of Engineering

1 PROJECT SUMMARY

To promote consideration of omnidirectional actuators in a more diverse number of applications; an open-source omnidirectional actuator was conceptualized, designed, constructed and tested.

Upon selection of a spherical motor configuration for the actuator, further research was conducted to determine which technology would could be adapted to power it.

The research concluded with a selection of brushless DC motor technology as a means to actuate the spherical motor drive, with an emphasis on linear induction technology.

To promote the development of omnidirectional drive systems within the open-source community a set of generalised performance metrics was developed.

Initially work was undertaken to better understand the principles behind linear induction and permanent magnet motor technology. Maxwell's equations as well as insight from academic literature on the subject were then used to construct an analytical model in MATLAB. The model was validated through both numerical and experimental means.

Optimum design parameters, based on the recommendations of the design tool, were used to propose a design foundation from which a number of design concepts were generated. The designs were generated with consideration to the manufacturing capabilities of the IForge and the cost of materials.

In parallel to the physical design, and working within the constraints of accessibility and Arduino-based hardware, a three phase open-source motor controller was developed.

On account of Covid-19, the decision was taken to continue manufacture as accessibility was a design consideration. Third party suppliers were sourced in order to help build the components. In order to achieve this within an expanded, but still small, budget significant redesign was undertaken.

The motor was then switched on and and an attempt to characterise performance was made.

The findings were uploaded to the Project's hackaday page in addition to a video which explained how any viewer could go about constructing their own device, starting with the motor controller and power supply.

CONTENTS

NOMENCLATURE

The next list...

Physical Constants

ϵ_0	Permittivity of Free Space	$8.8542 \times 10^{-12}\, m^{-3} kg^{-1}$
μ_S	Relative Permeability of Mild Steel	1000
ρ_{Cu1}	Density of Copper	$8,940\, kg.m^{-3}$
ρ_{Cu2}	Resistivity of Copper	$1.71 \times 10^{-8}\, \Omega.m$
ρ_{Cu1}	Density of Steel	$7,750\, kg.m^{-3}$
ρ_{Cu2}	Resistivity of Steel	$1.18 \times 10^{-7}\, \Omega.m$
ρ_{Cu1}	Density of Aluminium	$2,700\, kg.m^{-3}$
ρ_{Cu2}	Resistivity of Aluminium	$2.65 \times 10^{-7}\, \Omega.m$
ρ_{Cu1}	Density of ABS	$1,040\, kg.m^{-3}$
ρ_{Cu2}	Resistivity of ABS	$1.50 \times 10^{16}\, \Omega.m$
ρ_{Cu1}	Density of Air	$1.225\, kg.m^{-3}$
ρ_{Cu2}	Resistivity of Air	$1.30 \times 10^{16}\, \Omega.m$

Design Parameters (Default Values)

l	Stator Width	$0.06\, m$
R_in	Stator Inner Radius	$0.05\, m$
ϕ	Phases	3
p	Number of Poles	4
δ_{airgap}	Airgap	$2 \times 10^{-3}\, m$
F_{bf}	Bearing Friction	$0\, N$
R	Rotor Diameter	$0.1\, m$
δ_{Shell}	Shell Thickness	$0.03\, m$
ω	Frequency	$1\, Hz$
I_{peak}	Current Limit	$4\, A$
V_{peak}	DC Source Voltage	$12\, V$
R_0	DC Internal Resistance	$1.5\, \Omega$
R_{load}	Additional Circuit Resistance	$0\, \Omega$
$Slot_w$	Slot Width	$0.005m$

$Slot_h$	Slot Height	$0.01m$
$N45$	Neodymium Magnet Rating	
X_{PM}	Permanent Magnet Cube Length	$5 \times 10^{-3}\, m$

Key Design Parameter

M_{Shell}	Rotor Shell Material	
M_{Core}	Rotor Core Material	
SF	Linear Size Scale Factor	
A_{Coil}	American Wire Gauge of Coil	AWG

Physical Quantities

E	Electric Flux Density	$C.m^{-2}$
B	Magnetic Flux Density	T
Φ	Magnetic Flux Linkage	T
G	Goodness Factor	
q_{enc}	Enclosed Charge	C
I_{enc}	Enclosed Current	A
S	Slip	
J	Current Density	Am^{-2}
X_L	Reactance	Ω
X_R	Resistance	Ω
Z	Impedance	Ω

Mathematical Notation

$\overrightarrow{B}$	Denotes Vector Field
$\times$	Denotes Cross Product
$\circ$	Denotes Dot Product
$\hat{n}$	Denotes Normal Vector
$\oint$	Denotes Surface Integral

ACKNOWLEDGEMENTS

I would like to take this opportunity to thank the following people:

Mr David Polson, my supervisor, for allowing me to self-initiate this project - probably against his better judgement! I could not have asked for a more supportive or enthusiastic supervisor, in either the good times or the bad.

Mr Will Sanders, my lockdown housemate, whose incredibly impressive supply of gadgets and gizmos, not to mentioned endless patience, allowed me to complete this project under the changing circumstances.

Electric Motor Suppliers Ltd, for giving me the AC induction motor that formed the basis of my experimental validation.

2 INTRODUCTION

2.1 Omni-Directional Drives

Most vehicles, robotics and machines are driven by single axis rotary motors or actuators. Manoeuvrability is achieved by way of steering mechanisms that drive a single DoF rotary actuator around a central axis. This is commonly achieved either with a secondary rotary motor or else a system of mechanical linkages.

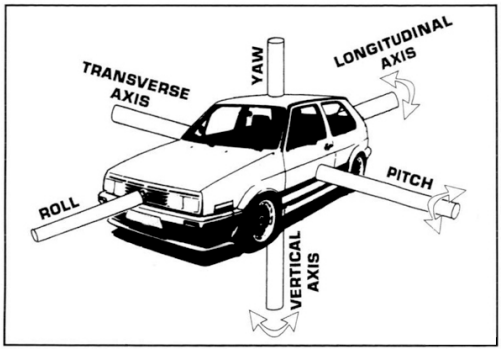


Figure 2.1 – Movement Axes Diagram

However, such steering mechanisms are incapable of producing translation along the transverse axis. Therefore they are incapable of providing instantaneous omnidirectional movement. Additionally, manoeuvrability provided by such steering mechanisms comes with the following limitations:

- Limited Steering Angle: Mechanical constraints typically only allow steering through an angle of $70°$ (such as in commercial automotive vehicles) [1], rising to about $120°$ in a research race vehicle reviewed in Race Car Dynamics. [2]
- Wheel Slip: If the momentum of the machine forces travel in a direction other than that in which the machine is currently steered for then a lateral force, relative to each tyre, is generated. [3]
- Supporting Machinery: All machines which utilize steering require a driving mechanism in order to do so. In most cases a series of mechanical linkages must be present throughout the machine . Adding to the mass, reducing the available volume and increasing deadband in the system[4]. This limitation is lessened by containment of a secondary motor in close proximity to the wheel, such as in EVs.[5]

Consequentially the ability of these machines to carry out their function efficiently is compromised. The cost of that inefficiency, in the context of traffic delays in large American cities, was quantified by Schrank et al. as $115 billion in 2009. [6] Furthermore Verhoef and Rouwendal state that traffic congestion also has a significant impact on vehicle speed, traffic safety and time cost.[7]

Li et. al conclude that the manoeuvrability of vehicles is a significant factor and that the cost resulting from these limitations could be reduced or alleviated through the adoption of omnidirectional machines that are capable of movement in 3 DoF. [8]

Omnidirectional motors can be divided into two types: holonomic and non-holonomic. A holonomic machine is so defined as being capable of movement in as many DoF as controllable DoF in the system. Therefore the position of the system is directly integrable from the inputs to the system. [9] Consequently, any machine that requires a steering mechanism is non-holonomic, even if it is capable of omni-directional movement (e.g. a swerve drive).[10]

The distinction is mostly semantics and the capabilities of either type of omnidirectional drive are virtually identical. The primary difference is found to be that the control system can be more accurate and efficient for a holonomic machine.[11]

Ultimately an omnidirectional vehicle requires an omnidirectional drive in order to operate within 3 DoF.[12] [13] [14] [15]

2.2 Limitations of Omnidirectional Drives

Despite the limitations of single DoF rotary machine drive mechanisms the simple design and inexpensive construction means that they permeate locomotion technology today.

Therefore any omnidirectional drive looking to usurp single DoF rotary machines as a major constituent of drive mechanisms must not only justify cost and complexity through performance, but actually perform better by a significant margin. Masters and Thiel quantify the significant margin to be an ”order of magnitude” in their work on the permeation of new technologies and successful startups.[16]

This is because the widespread adoption of wheels has created an inertia in the human psyche , as well as an inertia composed of physical machines which would have to be re-engineered.

Also, presently wheels are considered synonymous with mechanized motion and so engi-

neers and designers are likely to begin their thought process for solving problems of locomotion with wheels, instead of looking to alternative technologies. [17]

To promote adoption of the spherical drive system into the mainstream engineering thought process the various types of omnidirectional drives should become better understood, better characterized and more endemic within the engineering community.

It is the proposition of this report that the three objectives described above can be achieved simultaneously via promotion of omnidirectional drive systems within the open-source maker community.

2.3 The Maker Community

Open-source is a term that, historically, described software. The definition being *"having the source code freely available for possible modification and redistribution"*. [18] The same concepts can be applied to hardware however.[19]

An organization that specializes in enabling communities of people to engage with open-source hardware is Hackaday. Hackaday describe their purpose as to promote "Hacking", the term used to denote Hackaday's definition of the word *... is an art form that uses something in a way in which it was not originally intended. This highly creative activity can be highly technical, simply clever, or both*.

This, in addition to Hackaday's promotion of "the free and open exchange of ideas and information", would suggest that Hackaday could provide an effective germination for a wave of omnidirectional thinking.[20]

Furthermore, Fisher and Gould [21] state that open-source hardware provides a low-cost alternative for scientific instrumentation and research.

When these two observations are made in tandem, it becomes not unreasonable to see the open-source Maker community as a type of decentralized, non-profit technology accelerator, of sorts. Therefore, through the documented design and construction of an inexpensive & effective omnidirectional drive technology, and by the promotion of it in open-source communities, mainstream consideration of omnidirectional technologies could be expedited.

3 LITERATURE REVIEW

3.1 Methods of Omnidirectional Drive

A number of methods have been suggested by Cawood. [22]

3.1.1 Mecanum Wheels

An assembly composed of two Mecanum wheels (Figure 3.1) situated along the same central axis and displaying mirror symmetry would be able to achieve full motion along the transverse axis, in either direction, through actuation of the Mecanum wheels in opposing directions. Conventional longitudinal movement could be attained through actuation of both wheels in the same manner. [23] A system utilising Mecanum wheels is incapable of rotation around the vertical axis however.

3.1.2 Swerve Drive

The swerve drive (Figure 3.2) is an omnidirectional drive technology that can pivot a single DoF rotary actuator around the vertical axis with no restriction. Although not technically appropriate for holonomic vehicles, swerve drives are capable of providing omnidirectional drive to a machine that utilises them. In conjunction with omni-wheels, the burden placed upon the control system can also be reduced significant. [24]

3.1.3 Ball Drive

A ball or spherical drive system (Figure 3.3) operates by transmission of power to a spherical rotor instead of a cylindrical rotor or wheel. This means that 3 DoF actuation can occur instantaneously with a single action rather than as the output from multiple actions. [25]

A major disadvantage of the spherical drive system is that the ability to provide 3 DoF power transmission directly through a shaft or axle is lost. Shafts are particularly efficient methods of power transmission (virtually 100%, baring hysteresis losses and bearing friction). Therefore the effectiveness of a spherical drive system to transmit power in any number of DoF is dependant on the efficiency of the particular alternative power transmission method used.

Figure 3.1 – Commercial Mecanum Wheel

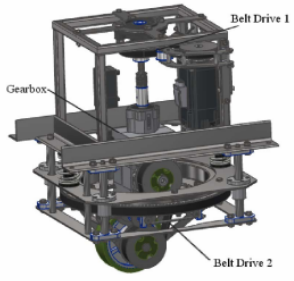


Figure 3.2 – Swerve Drive [10]

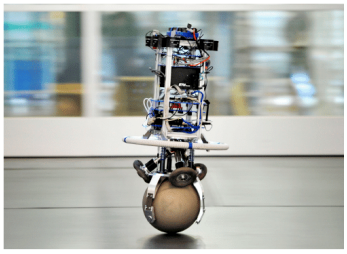


Figure 3.3 – Ballbot Rezero [26]

3.2 Omnidirectional Drive Selection

Of the three omnidirectional actuators discussed, the spherical ball drive is considered to harbour the most potential. This is because the spherical motor, unlike the other technologies discussed, can provide omnidirectional drive for a true holonomic vehicle.

An additional point, the relative simplicity of the spherical motor means that it is not just limited to vehicular applications; it could also be used to provide omnidirectional actuation to a joint or interface. For instance in the use of robotic arms, optics, gyroscopes and other rotating machinery. Such an actuator would have the advantage of compact size, high motion precision, fast response, a directly driven mechanism and a high efficiency as shown by the authors Howe, Oner and Chen. [27]; [28]; [29].

3.3 Overview of the Arduino Platform

Arduino is a brand of open-source micro controllers and computers. Arduino is the name of the company that first produced the designs and their name has become synonymous with their flagship line of products. However the open-source nature of their products means that there are multiple producers of technology with the arduino specification (such as Elegoo or Seeduino) that are functionally identical, but less expensive. For the purposes of this report all technologies that align with the specification of any Arduino device will be referred to as ”Arduino”.

Arduino are ubiquitous within the making community. They are inexpensive, relatively powerfully, modular in nature and can support programmers of all abilities. In fact the intended use of Arduino was as an educational tool.

The conclusion to be drawn is that if an omnidirectional technology could be produced with Arduino as the foundation then the accessibility of the project will be greatly increased, promoting widespread adoption.

3.4 Omnidirectional Drive Systems

Research has been ongoing on the topic of omnidirectional drive systems, in one form or another, since 1959 [30] and in the decades since a multitude of omnidirectional drives have been designed, built and patented. The presence and scope of such research warrants an extensive literature review.

Much like with conventional motors, the term "omnidirectional drive" is used as a catch-all term for any singular mechanism which actuates fully with 2 or more DoF. These systems can be classified further on the basis of their operating principles, as will be case in this review.

The inclusion criteria for further research is as follows:

1. No requirement for mechanical commutation.
2. Rotor must not require power supply to operate effectively.
3. Rotor geometry must have rotational symmetry, such that any 1 DoF technologies can adapted for 2 or 3 DoF application.
4. The technology must broadly align with a rotor and stator configuration. This rules out technologies such as the pendulum drive, which is used to operate the on-set model of BB-8 during the Star Wars films. [31].

The technologies researched during the literature review can be divided into the following umbrella categories.

- Ultrasonic
- Inverse Mouse Drive
- Brushless DC

3.4.1 Ultrasonic Motors

Ultrasonic Motors

The stator of an ultrasonic motor is comprised of an arrangement of piezoelectric transducers. By applying a voltage to these transducers oscillations are produced and, consequently, frictional forces on the rotor are produced. Through modification of the power supply voltage and transducer configuration control can be attained over the resultant speed and torque characteristics of the motor.[32] Alternative designs may achieve actuation through the manipulation of standing waves within the rotor.

Unlike other omnidirectional technologies, which mainly comprise adaptations of singular DoF rotary machines for spherical application, most ultrasonic technologies have been developed specifically for the purpose of actuating spherical rotors.

Consequently ultrasonic motor technologies show great potential as a means to produce ultrasonic motor systems that are compact in size, produce high torque at low speeds and generate high braking torque without the need to consume power. Furthermore they are capable of high resolution angular positioning as discussed by Hoshina et al.[33] in their work to develop a camera actuator for a pipe-cleaning robot.

Within this branch of omnidirectional drive development there are found to be the following schools of design:

3.4.2 Single-Stator Ultrasonic Motors (SSUM)

Technology Review

Utilizing only one transducer, SSUM cannot utilize frictional forces to create an omnidirectional motor. Instead SSUM induce different frequencies of vibration in the rotor where each mode of vibrations correlates to a particular mode of rotation.

Spherical Application

Examples of devices in literature include Rogers et al. [34] who presented a motor with a 350 micrometer diameter rotor that used multiple vibration modes of a vibrating cylinder (the transducer) to actuate the motor. [35]

In 2011, Zhang et al. also utilised the vibration modes of a cylinder to actuate their prosthetic finger joint, but they used the association of non-resonant and resonant vibration to achieve greater control over the actuator. Ultilisation of vibration modes synthesised from different transducer geometries form a common design variation within the literature. The design of a single-stator ultrasonic spherical motor based on the vibration modes of a disk like structure was presented by Lu et al. in 2010. [36]

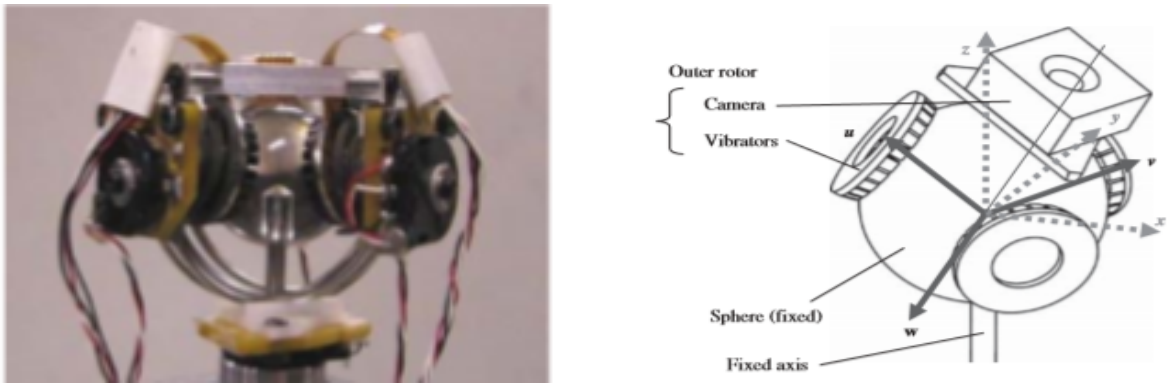
Further variations on this principle have also been proposed. For instance, a motor using surface acoustic waves on a single stator to actuate a sphere was proposed by Tjeung et al. in 2011. [37]

Appendages of all descriptions are a common application for SSUM, partially by virtue of the compact volumes possible. This quality was a key factor in the decision of Luo et al. to develop an ultrasonic motor powered bionic eye and thereby lay foundations for functioning prosthetic human eyes in the future. [38]

3.4.3 Multi-stator Ultrasonic Motors (MSUM)

Technology Review

Whereas the single stator spherical motor would use one actuator to induce vibration, the MSUM uses three or more piezoelectric vibrators in tandem. [32] In 2014, Leroy et al. considered that the use of independent parts limited the coupling of vibrations, which would otherwise have reduced the control resolution. Leroy et al. proceeded to offer numerous possibilities for stator design for MSUM.



(a) Experimental MSUM (b) Illustration of MSUM

Figure 3.4 – Multistator Ultrasonic Motors

Spherical Application

Examples of MSUS motors include the multiple travelling wave motor proposed by Hoshina et al. in 2013 [33], a spherical motor based on four asymmetric standing wave actuators as described by Shen et al. in 2010 [39], and a motor combining multiple standing waves in four individual plates as presented by Otokawa et al. in 2005.[40]

3.4.4 Inverse Mouse Drives (IMD)

Technology Review

The IMD is so named for the visual similarities to the construction of a computer mouse ball. Only seen in a spherical context, the motor is driven by applying power to the independent omni-wheels that are arranged tangentially at equal intervals and are in contact with the spherical rotor. The nature of operation means that no special quality is required by the rotor, barring adequate structural integrity. The use of an inflated basketball as a rotor for "Basketball Rider" exemplifies this point. [41]

Spherical Application

IMDs are common amongst omnidirectional systems, as they are simple to design, inexpensive to produce and robust in operation. However, serious application of the inverse mouse drive is limited. This is because an IMD requires a single DoF, per wheel. Consequently, in order to rival the power of a single DoF motor the IMD must house between two and four separate motors of the same specification. A problem which is exacerbated with scale.

This limitation has not stopped all vehicular endeavours however. Students at San Jose University, California developed an SDS self-balancing motorcycle as part of a student led learning exercise. The vehicle utilised a tri-wheel IMD. Concept designs, technology test rigs and a prototype were produced. [42] However, the group disbanded soon after.

(a) Self-balancing SDS Motorcycle (b) CAD render of proposed motorcycle

Figure 3.5 – San Jose California, SDS Motorcycle

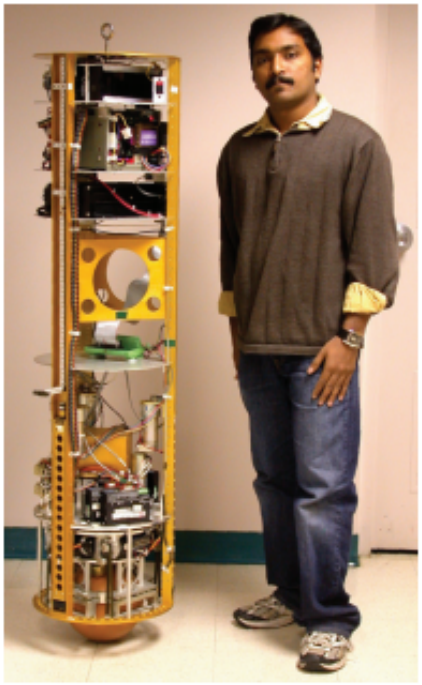


Figure 3.6 – Ballbot and creator

The "Ballbot" series of projects presents a more successful endeavour. Initially, it was proposed by Lauwers et al. in 2006 that the IMD technology showed enough potential for an entire generation of humanoid robots, based on spherical drive technology, to be spawned on the basis of the group's findings. [43]

Their work was built upon to produce the device pictured in the figure. [44] The bulk of their work was concerned with the control aspect however, specifically to allow the robot to self balance and act appropriately to a human-environment. Less detail was given in regards to the performance of the drive system, although their work does reveal that the IMD used was capable of driving Ballbot at velocities of at least 0.8 ms^{-1} and could meet the demands of the control system in order to resist the impulse received from a 310N kick imparted over 0.4s.

Brushless DC Motors

Brushless DC motors are so described because they lack the mechanical brush commutation that their brushed counterparts had required in order to generate a travelling magnetic field. The polarity switching necessary would instead arise from the motor controller and the software that drives it. The increasing processing speeds of micro controllers in recent decades [45] has lead to widespread adoption of brushless DC motors as motor controllers are more easily able to approximate AC current at high frequencies through implementation of PWM.

Despite sharing a common name and operating with different principles, many types of electric motor fall under the umbrella of the term brushless DC. United by the requirement for wave-forms approximated from a DC power supply via PWM.

3.4.5 Stepper Motor

Technology Review

Stepper motors fit within the umbrella category of brushless DC electric motors. The stepper motor differs from other brushless DC motor variants by the requirement for power

to be applied discretely to each coil in the form of pulses, as opposed to as a simulated AC waveform. This method of operation allows stepper motors to work effectively with open-loop control systems rather than being restricted to only closed-loop control systems, as is the case for other types of brushless DC motor.

The salient poles on the rotor and the stator are designed with an offset such that only one set of poles can be aligned at any one time. Therefore when a stator pole becomes energized there will always be a corresponding rotor pole which is closest in proximity and therefore is forced into alignment with the energized stator pole. The mechanism which gives rise to this force varies depending on the construction of the stepper motor.

This is possible because, in a stepper motor, the number and configuration of pulses sent by the controller will directly correspond to the position of the motor; on the condition that the motor does not lose synchronization (owing to excessive load torques or operating speeds). The ability to determine rotor position from only the input pulses and without the requirement for a specific output measure lends the stepper motor for use with open-loop control systems.

On account of this property stepper motors are used in industry for applications that require precision automation and control at a low cost. However stepper motors can also function similarly to brushless DC motors. This is achieved through supply of enough pulses at a rate such that the movement of the rotor becomes smooth and continuous.

Hughes tells us that the term used to describe stepper motor operation in this manner is referred to as "slewing".[46]

As mentioned previously, the construction of a stepper motor determines the manner in which force is generated. Here will be discussed the operating principles and relevant omnidirectional drive utilization of permanent magnet stepper motors and variable reluctance stepper motors within the literature.

Spherical Application - Variable Reluctance Stepper (VRSM)

In 1988 Lee and Kwan developed a spherical wrist actuator for use in prostheses which operated via the principles of variable reluctance. [47]

Later, in 1991, Lee et al also conducted an FEM analysis of a variable reluctance stepper motor, concluding that the technology held promise for further development as they were

able to exert significant control over the motor and torque output through manipulation of the spherical magnetic field. [48]

Spherical Application - Permanent Magnet Stepper (PMS)

The concept has proved readily adaptable to spherical motor development. PMS was the guiding principle behind Chirikjian et al., who in 1999, developed a general theory behind the kinematics and commutation of a spherical stepping motor and analyzed potential stator and rotor pole layouts for optimum performance. [49] To achieve this they attempted to use platonic geometry to achieve an isotropic distribution of poles over the sphere. This proved impossible for the number of poles they required and so they utilized a superposition of platonic geometries to produce a satisfactory, but not perfect design.

The work of Chirikjian et al. formed the basis for the work of Wang et al. who, in 2006 [50], used analytical and FEM techniques to propose improvements upon Chirikjian's design which were subsequently tested experimentally. Their report concluded with an optimistic outlook for the future of the technology.

3.4.6 Linear Induction Motor (LIM)

Technology Review

According Laithwaite's seminal work on the matter [51], an LIM can best be described as a (cylindrical) asynchronous AC induction motor rolled flat, and operated by a principle that was analogized by Laithwaite as a "Magnetic River" [52]. This term being a holistic description for the travelling wave of magnetic flux generated by the motor along the length of its stator coils. The wave of flux induces eddy currents in the rotor secondary that, in turn, generate flux which acts to oppose that flux that created it. This produces a force on the rotor that grows in magnitude, as the initial flux wave moves away from the languishing rotor, until the force upon the rotor accelerates the rotor to the velocity of the travelling magnetic wave. The lag in such system is quantified as slip.

The technology, originally utilized in textile mills around the turn of the 20th century, was pioneered by Laithwaite et al. to form the foundation of an advanced hovertrain (MagLev) infrastructure for Britain in the 1960's and 70's [53]. Despite this venture not being suc-

cessful, LIM technology has since been developed considerably and applied to many fields including both liquid metal pumping and mass-acceleration. [54]

High profile usage has also been seen through the efforts of countries like Japan and China, and private ventures such as Elon Musk's Hyperloop [55] to provide fast and efficient public transport infrastructure with MagLev technology.

Linear motors are well suited to this endeavour because, in addition to generating force co-linear to their length, they also generate a small but significant repelling force in the direction of the vertical axis. The repelling force may allow the secondary to rise off the primary stator and form an air-gap. Although the situation is dynamically unstable.

Laithwaite and his colleagues remedied this in 1971 by wiring a prototype LIM incorrectly and inadvertently constructing a transverse-induction Machine, which was capable of maintaining the secondary on a central plane during operation.[56]

When LIM technology is applied to spherical applications the repelling force will act to form an airgap and thereby reduce frictional forces incurred during operation. Depending on the stator geometry and operating parameters there is also a possibility of achieving full levitation of the rotor. This would eliminate the need for dynamic mechanical interfaces, such as bearings, completely during operation.

It is also feasible that the effect could be exploited in order to supply electromagnetic damping to a system containing the omnidirectional drive. [57]. This could prove valuable, particularly for vehicular application, as a means of augmenting or replacing conventional suspension.

Spherical Application

The seminal work on the application of *LIM* technology to spherical motors, was carried out by Laithwaite as part of Williams et al.[30]. A series of prototypes were tested and possible future applications proposed.

In 1987 an analytical treatment was given to the magnetic fields and torques that can be expected in a spherical induction motor by Davey et al. [58]. Recent application of the technology include Fernandes and Branco who, in 2016, developed what they termed a "shell-like" motor with a unique stator winding layout, inspired by the muscle system responsible for shoulder movement in humans, was intended for use in in low-speed application[59].

3.4.7 Permanent Magnet Synchronous Motor (PMSM)

Technology Review

The third variation of brushless DC motor covered in this review is the PMSM. Instead of inducing eddy currents within the rotor a PMSM relies on permanent magnets present within the rotor to generate flux. As the flux wave induced by the current in the windings travels along the length of the stator the permanent magnets become locked in place relative to the travelling magnetic wave. The rotor will therefore move in synchronization with the excitation current.

The construction of a permanent magnetic synchronous motor is not dissimilar to that of the permanent magnet stepper motor described earlier. In broad strokes they differ only by the nature of current supply and coil winding configurations. A stepper motor discretely energizes and de-energizes closed coil loops around separate stator poles, whereas a PMSM energizes coils which span the length of the stator and does so by emulation of an AC current waveform.

Early PMSM suffered from the de-magnetization of their permanent magnets during operation. Though modern Neodymium-Iron-Boron suffer no such fate and generate significantly more flux than their non-rare earth magnet counterparts. These qualities present the possibility that PMSM will become dominant motor technology in the future. In 2016 Lu argued that PMSM will eventually replace all other less efficient motor technologies as the era of efficient motors takes hold. [60]

Spherical Application

As a lynchpin of future motor technology it is perhaps unsurprising to find that PMSM have undergone adaptation to spherical applications in recent years. Not all as omnidirection actuators in a strict sense however. In 2013, Park et al. developed and tested a PMSM spherical motor that utilised 3 DoF to create a tilting shaft, in a manner not dissimilar to a powered ball joint.

Also, efforts from Li et al. in 2018 produced a number of papers detailing different aspects of design for an omni-directional PMSM device. Papers focused on the stator design, flux generation and control aspects of the PMSM. The same team later proposed a modified design which utilized a Halbach array, a particular configuration of permanent magnets which

amplifies flux density on one side of the array while nullifying it on the other side, in order to both increase efficiency and reduce the impact of the flux fringing presented by the spherical motor air gap. [61]

3.5 Technology Selection

After consideration of the technologies, linear induction technology was selected to form the basis of this project's eponymous open-source & Arduino-based spherical drive system.

3.6 Conclusion from Literature Review

From the literature review it has been understood that work has been undertaken to design and characterise spherical motors since the 1950's. Several projects on the topic conclude with optimistic outlooks on the future of the technology. Such projects describe at length the potential applications of omnidirectional technology such as:

Prostheses including bionic eyes, actuated finger and wrist joints. Locomotion for high-manoeuvrability vehicles ranging from wheelchairs and pipe-cleaning robots to industrial transport vehicles. Power-transfer in multiple DoF, ranging from helicopter rotors with tilt-control to wind turbines that can take full advantage of the turbulence within an oncoming flow of air.

However, while usage of the technology has occurred, use has only occurred in limited commercial circumstances and even then more often as a PR stunt than as a serious and considered solution. Goodyear's spherical tyre [62] and Audi's RSQ, as seen in 2004's I, Robot [63]

While it is clear that most of the potential applications for omnidirectional drives listed above can and have been met with single DoF rotary drives, there is still room for omnidirectional drives in the technoscape. An omnidirectional drive could be utilised in the design of a machine to remove the need for the compromises in performance that arise when adding additional DoF to single DoF rotary drives.

Therefore this project will endeavour to design, construct and test an open-source & Arduino-based spherical motor that utilises brushless DC motor principles, and do so in a manner as to provide a framework upon which future omnidirectional drive systems can be designed, constructed and tested. By widening the pool of stakeholders it is hoped not only that the

technological state of omnidirectional drives is improved and cost thereof reduced, but also that the concept comes more readily to the thought process of any engineer when faced with a problem to which such machines may be the best solution.

3.7 Aims & Objectives

1. Develop a parameterised model of both the LIM and PMSM spherical motors and use the model to optimise the parameters for prototype design.
   (a) Derive from classical physics and academic literature a set of equations that underpin the operation of linear induction technology.
   (b) Adapt equations for use in a parameterised model for a MATLAB-based design tool.
   (c) Validate model with experimental and numerical methods.
2. Develop a set of characteristics that can be used to characterise omnidirectional drive systems constructed in the open source community.
3. Design and construct a functioning spherical drive system.
   (a) Use the design tool to propose optimal key design parameters and associated dependant design parameters.
   (b) Create design concepts utilising the proposed design parameters.
   (c) Propose methods of construction that are easily accessible for makers and the open-source community.
   (d) Design, develop and test a motor controller for use with the spherical motor.
   (e) Construct prototype spherical drive system and test.
   (f) Characterise the spherical motor in the context of the parameters proposed for omnidirectional drive systems.
4. Promote omnidirectional and spherical drive systems within the open-source community.
   (a) Create a Hackaday blog and update regularly.
   (b) Create an education video about an aspect of the design process and post to YouTube.

4 THEORY

4.1 The Maxwell Equations

Both LIM and PM synchronous motors depend on the principles of Electromagnetism in order to function. A brief grounding on the relevant aspects of the Maxwell equations was considered appropriate. [64]

4.1.1 Faraday's Law

Faraday's Law describes the relationship between the *emf* induced in a circuit by a time-dependant magnetic flux enclosed by the circuit, and the time-dependant magnetic flux that gave rise to it.

$$\oint_C \overrightarrow{E} \circ \overrightarrow{dl} = -\frac{d}{dt} \int_S \overrightarrow{B} \circ \hat{n} da \qquad \text{Equation 4.1}$$

The left-hand side represents the line-integral of the vector electric field flux $\overrightarrow{E}$ that acts co-linearly over a infinitesimal segment of circuit. The right-hand side represents the time derivative of the surface integral of magnetic field flux $\overrightarrow{B}$ which acts through the surface bounded by the path of electric flux.

The implication is that a circulating electric field is induced from a changing magnetic flux linkage. However, Faraday's law in unaltered form could also be incorrectly applied to an electric particle moving through a magnetic field.

This is normally remedied with the caveat that $\overrightarrow{E}$ represents the electric field measured relative to a stationary segment the flux path or circuit, dl. An alternative formula can be given, as denoted below.

$$\oint_C \overrightarrow{E} \circ \overrightarrow{dl} = -\int_S \frac{\overrightarrow{B}}{dt} \circ \hat{n} da \qquad \text{Equation 4.2}$$

4.1.2 Lenz's Law

Lenz's law represents an observation on Faraday's law. Specifically it describes the phenomenon that arises by inclusion of the minus sign on the right hand side of Faraday's law.

The implication of the minus sign can be summarised as Lenz's law, that they are induced by a changing magnetic flux that flows as to oppose the change in magnetic flux that induced it.

4.1.3 Biot-Savart Law

The interconnected nature of electricity and magnetism is clearly defined within Maxwell's equations. Less clear immediately however is the direct contribution from any moving point of charge, to the surrounding magnet field field. This is described by the Biot-Savart law.

$$\overrightarrow{B} = \overrightarrow{H}\mu_s = \frac{\mu_0}{4\pi}\frac{Id\overrightarrow{l} \times \hat{r}}{r^2} \qquad \text{Equation 4.3}$$

The above equation is universal in application, although it has commonly been further manipulated in order to describe the magnetic field contribution from a given amount of charge flowing through a defined, geometrical path or circuit.

4.1.4 Lorentz Equation

Derived from Gauss' law for magnetic fields, the Lorentz equation describes the force that a charged particle experiences when moving within a magnetic field.

$$\overrightarrow{F}_B = q\overrightarrow{v} \times \overrightarrow{B} \qquad \text{Equation 4.4}$$

The force exerted on the charged particle is equal to the cross product between the product of the magnitude and velocity of the point charge, and the vector magnetic field strength in which it is moving.

4.2 Linear Induction Motor

A linear induction motor is at its core an unrolled induction motor. Ignoring some of the consequences of linear topology, such as end-effects, the linear induction motor operates using the same principles.

4.2.1 Operating Principles of the Induction Motor

All induction motors consist of a stator, wound with coil, and a rotor. For linear induction motors the terminology differs slightly and the stator is referred to as the primary and the rotor as the secondary.

A single phase of alternating current can be passed through a winding in the stator. The frequency of the AC signal ,f, RMS voltage ,V_{RMS}, and RMS current ,I_{RMS}, are key parameters.

In accordance with Faraday's law, the AC current flowing through the winding gives rise to a circulating magnetic field acting perpendicular to the direction of electric flux within the winding. Lenz's law can then be used to determine the direction of circulation of the magnetic field. The direction can also be determined empirically by application of Fleming's left hand rule.

In any frame of reference the individual circulating magnetic fields around the winding circuit can be taken collectively as a magnetic field. Much like the magnetic field generated from a permanent magnet, the magnetic field produced is a dipole system with the flux emergent from one pole and recirculating around the winding to complete the loop in accordance with Gauss' law.

The direction and strength of the magnetic vector field at any point in time is determined by Faraday's law. Specifically it is proportional to the negative of the rate of change of the winding current. As the AC waveform is sinusoidal, the transverse magnetic vector field direction and strength appear 90° ahead in phase relative to current.

The magnetic field emanating from the winding $\overrightarrow{H}$ magnetises the high-permeability, low-coercivity ferromagnetic material which comprises the stator. The result is the magnetic flux density vector $\overrightarrow{B}$. This can seen through the Biot-Savart equation in equation 4.3.

The path of the magnetic flux travels through the air-gap between the stator and rotor and penetrates the rotor. Again in accordance with Faraday's law the changing flux linkage penetrating the secondary induces an *emf*, this will cause an eddy current to flow through any conductive part of secondary, circulating around the penetrative flux. In accordance with Lenz's law the eddy currents will flow as to produce a magnetic flux which opposes the flux that created it.

The free electrons moving throughout the eddy current conduction path do so within the radial magnetic field that gave rise to them, therefore they experience a resultant Lorentz

force.

If, as described in the previous section, only a single winding is excited with an AC waveform then the rotor will not move. This is because the net eddy current, and therefore the net charge velocity, is zero.

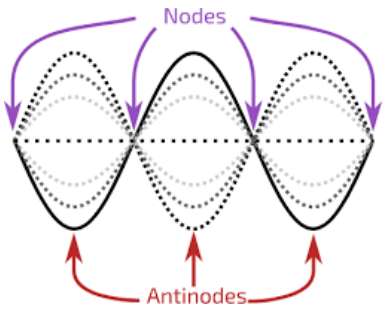


Figure 4.1 – Example of standing wave, as produced by a single phase AC current

This can be seen in figure 4.1. As the magnetic field emanating from the stator winding is in effect a standing wave, the nodes of the flux wave are stationary. By definition these nodes represent the areas on the secondary with the highest rate of change of flux linkage. At the locations of these nodes, according to Faraday's law, the largest *emf* in the system is generated and thus the greatest amount of circulating eddy current is induced.

This statement implies that the greatest number of charged particles are then moving incident to the greatest amount of flux linkage. The implication then is that the greatest amount of Lorentz force should be generated.

The net force is zero however. This is because, unlike in a travelling wave, at all points along a standing wave the rate of change of the relevant quantity is always of the same sign. Therefore as the standing flux wave waxes or wanes the eddy currents generated around the nodes flow as to oppose and cancel each other, completely.

However, movement can be achieved if there is relative motion between the standing flux wave, and the secondary.

This can be achieved either by moving the secondary within the standing flux wave that was generated by a single-phase AC winding, or else by using multiple phases of AC in multiple windings to create a travelling magnetic flux wave within the motor.

In such a scenario the sign of the rate of change of flux linkage relative will alternate between the distance of pole pitch along the wave. Therefore the eddy currents align to produce the maximum flow of charge and therefore the greatest Lorentz forces.

In such a scenario where the rotor is on the precipice of movement, the conditions for peak motor torque are present. This condition is also defined as one where the value of slip is maximum.

Slip refers to the relative difference between the velocity of the secondary and the velocity of the travelling magnetic field. The velocity of the magnetic field is also referred to as the synchronous speed.

As seen previously, maximum torque in an induction motor is produced when the value of slip is at its maximum, i.e. close to unity. Naturally the implication then is that when there is no relative velocity between the secondary and the synchronous speed there is also no torque.

Slip is then proportional to torque. Therefore if the motor is running with any amount of load the secondary will always have a velocity smaller than the synchronous speed. Although the greatest torque is produced when the motor is all but stalled, significant amounts of current are produced in this state and the operation of the motor ceases to be smooth. Therefore most commercial induction motors operate with a slip of between 1 and 5%.

4.2.2 LIM Topologies: Single-sided LIM

The Single-sided Linear Induction Motor (SLIM) is so named because there is only one primary stator located on a single side of the conductive secondary.

Unlike radial flux rotation motors and most other linear induction motors, the conductive secondary in an SLIM has no defined conduction paths. That is to say the secondary is a uniform sheet of conductive material.

The consequence of the lack of defined conduction paths in the secondary (such as laminations, cages or ladders) is that losses due to eddy currents can be significant. This is however the only form of secondary that can meet the requirements specified specified during the literature review, namely that rotor surface must be either homogeneous or patterned in an isotropic manner.

5 MODELLING

5.1 Analytical Design Tool

The analytical design tool was constructed in MATLAB, with the design objective of predicting the performance of a spherical linear induction motor through its independent design parameters. It was developed to help identify an optimum set of design parameters that could be used for the final prototype design.

The design tool could achieve this by prediction of both specific performance characteristics and the Goodness value of the motor, dependant on a given permutation of design parameters. The use of both output metrics would enable absolute and relative comparison between designs.

The tool would be validated through experimental and numerical means. The tool would then be used to identify optimum values for the key design parameters. Future work could be undertaken to optimise the motor design with respect to all of the design parameters by pairing the tool with an optimisation methodology or tool. The design parameters used can be found in the nomenclature.

5.1.1 Key Design Parameters

The use of a large array of design parameters meant a method of optimisation would need to be employed to identify the true optimum design. This was action was considered. However, a package of work to fully optimise the design was not undertaken because it was considered that a completely optimal spherical motor was not considered necessary to achieve the end goal.

Instead a compromise was made whereby a selection of independent design parameters, designated key design parameters, were varied while the other independent design parameters were kept constant. Values for the other independent design parameters depended on the task at hand. During validation the parameters were given values that were reflective of the experimental case, during design optimisation they were given values described in the literature for a similar motor.

Key parameters were selected on based on the following criteria:

- Significant use as an argument to design tool output functions.
- Independent of manufacturing process.

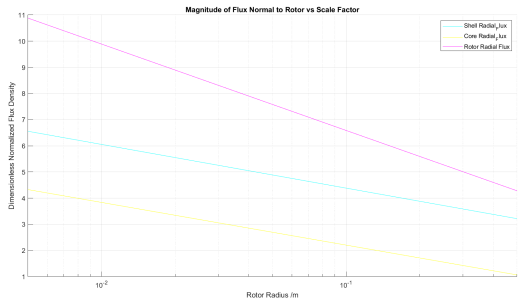
Denotes Key Parameter						
Scale Factor SF	h_{slot}	R	$\delta airgap$	$Slot_{arc}$	$\delta shell$	l
Rotor Radius R	92 Φ	m_{core}	m_{shell}	I_{core}	I_{shell}	
Core Material Core Material $Mcore$	ρ_{core}	ρ_{core}				
Core Resistivity ρ_{core}	E_{ind}					
Eddy Currents E_{ind}	F					
Shell Material $Mshell$	ρ_{shell}	ρ_{shell}				
Shell Resistivity ρ_{shell}	E_{ind}					
Eddy Currents E_{ind}	F					
Wire Diameter AWG	N					
Turns N	J	$\overrightarrow{B}$	X_L			

Table 5.1 – Parameters as arguments to other parameters

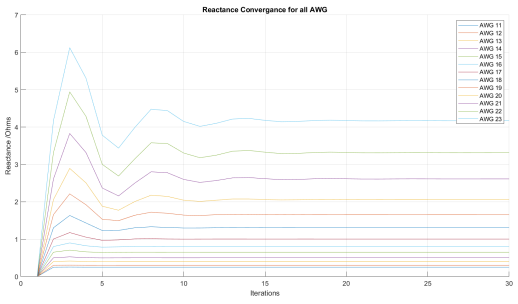
5.1.2 Analytical Methodology

The core of numerical analysis conducted by the design tool was based primarily upon the Novel method LIM thrust and lift calculation that was established by Ma et al. in 2012. [65] The operation of the design tool is described as follows. The code can be found in appendix B .

From the design and key design parameters described in the Nomenclature section all derivative static parameters were derived. Relevant parameters were passed through a series of functions that generated a value for the magnitude of flux density that was acting normal to the rotor, per coil.



(a) Flux normal to rotor against rotor radius



(b) Reactance convergance against Iterations

This process was iterated until convergence on a value of circuit reactance could be established.

The normally penetrating flux was then recalculated. The magnitude of eddy currents that would be induced in the shell and core sections of the rotor by the magnetic field were calculated. Subsequently, the Lorentz forces generated from the flow of eddy currents within the magnetic field were also calculated. The value of force was then passed to a series of functions to calculate the expected torque and acceleration of the rotor. A multi-tiered loop was established that would repeat the described numerical analysis for a predefined number of variations for each predetermined key design parameter.

- The current was considered to flow tangentially to the rotor at at all times, thereby enabling a 2D approximation.
- The losses incurred by eddy current were considered to be zero.
- The influence of harmonics present within the windings were considered to have no effect.
- The three-phase AC power wave form was considered to represent three perfect sinusoids, with exactly 120° between them, with a shared frequency matching exactly that of the defined frequency.
- The end-effects of the LIM motors used were considered to have zero effect.
- Thermal effects were assumed negligible
- The rotor is presumed maintain structural integrity
- The effect of flux fringing is presumed to be negligible

5.2 Validation

To validate the analytical model, it was proposed that both a numerical FEM simulation and experimental simulation be undertaken. An experimental design rig was designed and partially constructed.

However, the analytical design tool results predicted significantly lower values of radial flux and force production than had been anticipated, even with the assumptions made by the analytical design tool.

Additionally, it was realised that the cost of the complete construction of the experimental rig would use up a significant portion of the project budget. The possibility that there would

not be enough budget left over to complete construction of the final spherical drive system was considered.

An alternative method of validation was sought instead. In the same time period as the aforementioned quandary, a search for inexpensive sources of laminated electrical steel and copper windings was being undertaken. The author's search lead them to an electrical motor repair shop on Eldon street, Sheffield. Here the author was furnished with a three phase induction motor that would otherwise have been scrapped.

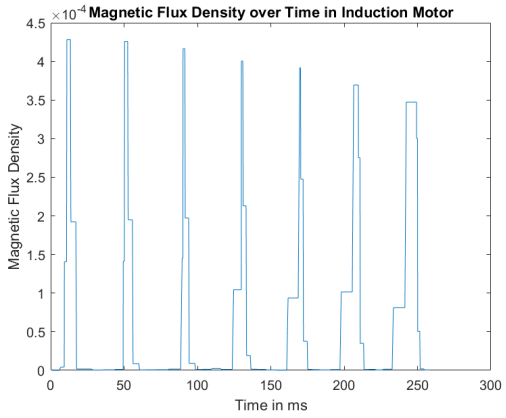
It was decided that validation of the analytical tool could instead be conducted through the use of a flux density measurement as opposed to force production, as force production is directly proportional to flux density equation 4.4.

Experimental validation of the experiment would therefore take place through the use of the salvaged induction motor, the numerical and analytical models would both be adapted to accurately reflect the design of the three phase induction motor. Validation would then be conducted using the radial flux density measurement.

5.2.1 Analytical

An instance of the design tool was run with the validation design parameters. The airgap radial flux density was found to be equal to $0.1988T$

5.2.2 Experimental



(a) Magnetic field strength in T (Single Winding)

(b) Three Phase AC Induction stator for test

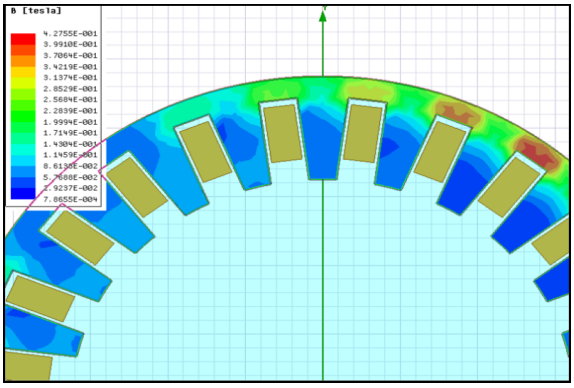
The method used to determine the magnitude of radial flux density is described in the Testing and Methodologies section.

The same operating parameters, including a frequency of $40Hz$ and peak voltage of $12V$, were applied as were used in the analytical and numerical models

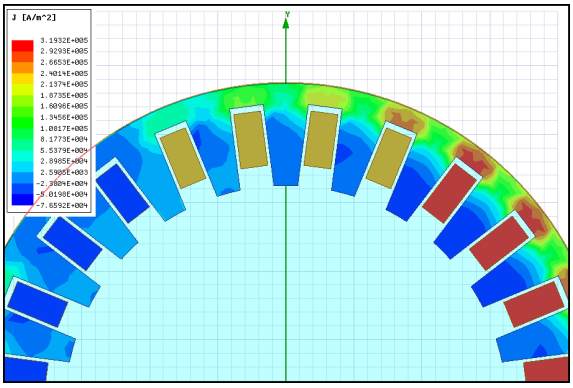
The peak magnetic flux density was found to be $4.50 \times 10^{-4}T$

5.2.3 Numerical

An FEM model with representative parameters to the induction motor was constructed in the ANSYS Maxwell FEM software. Both physical measurement and data sheet consultation were utilised to generate design parameters for the model. The design parameters inferred from the salvaged motor were combined with the output parameters from the Arduino motor controller to accurately reflect the experiment scenario.



(a) Magnetic Field Strength



(b) Current Density in windings

The peak magnetic field strength was found to be $0.427T$ and the residual of the numerical simulation found to be $4.58 \times 10^{-}7$.

5.2.4 Conclusion

During the validation process two of the quoted values for flux density agreed with each other, within an order of magnitude. This was considered acceptable due to small differences in design between the numerical and analytical. However, the numerical validation yielded a result which differed from the other two by multiple orders of magnitude.

The reasons for this are unclear. Though it was considered that the resolution required for a $40Hz$ signal was too low for the motor controller to reproduce the correct signal, or else that the power supply assumptions were incorrect.

The parameterised design tool was considered validated henceforth in the project timeline.

6 ARDUINO MOTOR CONTROLLER

6.1 Motor Controller Design Specification

Through consultation of the relevant literature and theory, and in accordance with the project scope, the following technical requirements were proposed:

- Variable frequency output
- AC output in three phases
- DC power supply
- Arduino Uno CPU

The motor controller and associated design criteria were initially considered as a single system. However, in order to conduct the design, the system was broken down into components associated with specific functions:

- Power Supply
- Inversion of DC signal
- Phase Shifting of AC waveform

Early on in the investigation it became clear that a significant factor in a successful design would be the relative ease of use and understanding. To this end a process was employed whereby promising components were purchased or borrowed until an insurmountable dead end was perceived to have been met, in an iterative manner. This process is detailed in Appendix A.

Ultimately, enough progress was made with the following components to warrant a transition to the motor controller development stage.

- Motor Controller: Arduino Uno
- Power Supply: 8 AA 1.5V Batteries in Series
- Inverter: Arduino Motor Shield Rev 3.

6.2 Code Development

Arduino code, written in the C programming language, was developed in order to turn the proposed configuration of hardware into a three-phase variable-frequency inverter and motor controller.

The primary objective in code development was to maximise the range of possible of frequencies while maintaining a high level of signal fidelity. It was considered that this objective could be achieved if the execution time of any looped code was reduced as much as possible.

The code used can be viewed in Appendix B.

6.2.1 SPWM Signal

A sinusoidal waveform was produced through modification of the duty cycle of the PWM signal outputted to a analog pin on the Arduino. Modification was achieved by looping a conditional instruction that would alter the PWM duty cycle after the duration of a predetermined PWM interval.

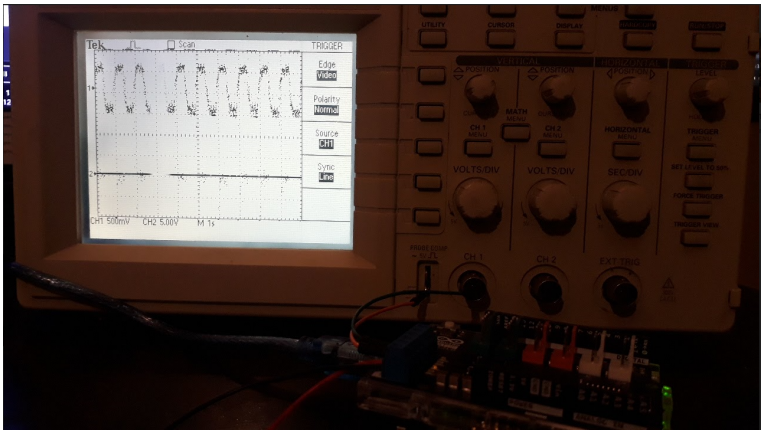


Figure 6.1 – 1Hz SPWM test with Arduino

To reduce execution time, the duty cycle value to be assigned was read from a look-up table, rather than being calculated afresh in each loop. The length of the PWM interval was determined by the resolution specified, as described in equation 6.1.

$$PWM_{int} = \frac{\lambda}{Resolution} \qquad \text{Equation 6.1}$$

The frequency of signal produced by the inverter was therefore proportional to the resolu-

tion specified and inversely proportional to the PWM interval. The limit to this relationship is described in Appendix A.

A value of resolution was proposed on the basis that the duration of PWM_{int} must be greater than the time to taken to execute the looped code t_{ex}.

t_{ex} was measured, via an inbuilt Arduino function, to be equal to 7 pre-scaled timer clocks, equal to $1.79ms$.

A table of frequencies and corresponding maximum resolutions was produced in order to inform code operating parameters.

$f - Hz$	1	5	10	20	30	40	50	100
$Res.$	558	111	55	28	19	14	11	6

In addition to maximising frequency, it was also considered important to maximise the range of frequencies producible in order to enable a more extensive performance characterisation process.

$$f_{range} = f_{up} - f_{lo} \qquad \text{Equation 6.2}$$

The lower limit of frequency exists as a result of the logic described in Appendix A. For same conditions in which the upper limits of frequency were derived, f_{lo} was found to approximately equal to 1Hz.

6.2.2 Polarity Switching

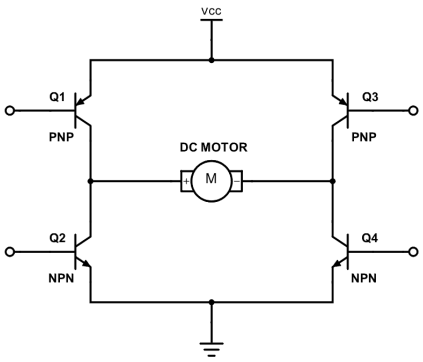
A limitation of PWM techniques is that they cannot produce the negative current required for an AC signal, only positive HIGH or LOW signals. However, negative current can be achieved through use of additional hardware to change the direction of the PWM signals through a given circuit.

This if often achieved in a motor controller setting through the use of a switching circuit, such as an H-Bridge, shown in figure 6.2.

In order to alter the switch states in the H-Bridge, and thereby alter the direction of current flow, timer interrupts were used. The mechanism behind the operation of the timer interrupts is described in Appendix A.

Code was written to alter the direction of current flow and attached to specified interrupt

Figure 6.2 – H-Bridge Circuit



functions in the form of an interrupt service routine (ISR). Two ISR were used per AC signal period. The first was attached to an interrupt that was offset in timing by half a period and would therefore switch the current direction at the central node of the sinusoid. The second was attached to an interrupt that would fire at the end and beginning of each period, switching the current direction at the end of the AC period and also clearing the timer counter.

To ensure that the execution time of the main looped code was not overly affected when the interrupts fired, the t_{ex} of the ISR contained only one or two actions and both were written in assembly code. The actions are described as follows:

- An XOR (bit-flip) operation that directly altered the state of the pins associated with the H-Bridge switches, featured on both ISRs.
- An operation to set the value of the TCNTn bit register to zero - featured only on one ISR.

6.2.3 Adaption for Three Phase Inversion

Ultimately, a variable-frequency single phase AC power was produced from an assembly of an Arduino Uno and an Arduino Motor Shield Rev3.

However, there was found to be a limiting factor in the hardware that was selected. This was because it was assumed that any hardware that could produce an SPWM signal with controllable frequency and phase could be scaled up in order to produce three phase power.

This assumption was found to be incorrect as it did not account for the collective need of the three SPWM signals to be synchronized.

Exploration of alternative methods to produce multiple phase-shifted and synchronized SPWM signals with the Arduino Uno was not fruitful. Fundamentally, this was because the Arduino Uno only contained one 16 bit timer that allowed only a limited amount of interrupts per timer.

It was considered that the CPU would require the same technology as the Arduino Uno, but more of it. Attention was then turned to another member of the Arduino family, the Arduino Mega.

The Mega runs on the ATMega2560 micro-processor chip. Unlike the ATMega328 chip, the ATMega2560 contains three independent 16-bit timers, each with their own dedicated interrupt functions. This solved the synchronization problem. However the Arduino Mega does not directly interface with the chosen Arduino motor shields.

At the cost of compactness, a solution was found whereby the relevant Mega pins were wired directly to the motor shields. This solution did not work as expected however, until each shield was mounted on an Arduino Uno. The three Uno-shield assemblies were then powered through the use of three Arduino plug-in power adapters.

6.2.4 Testing

Due to the lack of an oscilloscope after Covid-19, testing was performed at frequencies of $1Hz$, $5Hz$ and $10Hz$. At frequencies greater than these values it became difficult to distinguish a smooth sinusoidal curve of dimming and brightening with the author's eye.

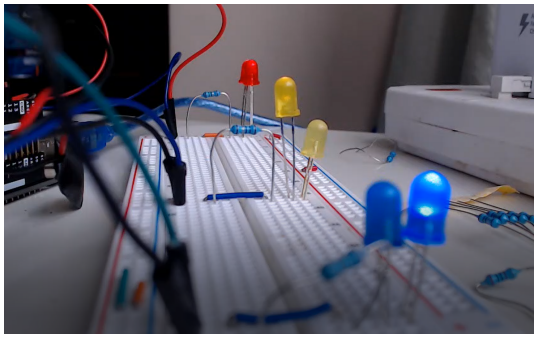


Figure 6.3 – Three-Phase Arduino Based Motor Controller LED test

7 SYSTEM ANALYSIS

To provide a framework upon which the omnidirectional motor could be characterized, two systems have been proposed. These are the specific metrics and the generic metrics.

- Specific Measurements are intended to characterize the motor qualities possessed that are inherent to all motors of that technology.
- Generic Measurements are intended to characterize the motor on the basis of qualities that are inherent to all omnidirectional motors.

7.1 Generic Metrics

The generic metrics were designed to quantify important characteristics of omnidirectional drive design, regardless of the underlying technology.

Five metrics were proposed:

- Inertia - Determinate of operating dynamics.
- Peak Torque - Determinate of maximum operating load.
- Efficiency - Determinate of operating efficiency.
- Driving Frequency - Determinate of rate of work done .
- Uniformity - Determinate of the extent to which performance can be replicated between two DoF.

The five generalised metrics have been defined as functions in terms of design parameters and measurable outputs. It was found to be the case that some metrics had multiple methods of possible derivation associated with them. Where this was found to be the case the derivation that was chosen would be the derivation that required the measurable outputs associated with the most accessible testing methodologies.

The derivation of the five generalised metrics are described in the sections below.

7.1.1 Inertia

The inertia metric was calculated through the use of the inertia formula for spheres of a homogeneous composition.

$$I = \frac{2}{5}MR^2 \qquad \text{Equation 7.1}$$

Separate calculations were performed for the shell and core of the spherical rotor and the results added together.

7.1.2 Torque

Newton's 2nd law, in respect to rotating bodies, is stated in equation 7.2.

$$T = I\alpha \qquad \text{Equation 7.2}$$

The torque metric would therefore be defined as the ratio between the inertia of the rotor and the angular acceleration measured upon start up, $\alpha 0$

7.1.3 Driving Frequency

The driving frequency represents the angular velocity of the driving mechanism. The measurement of driving frequency will vary based on the technology utilised by any omnidirectional design. For the case of the two motors discussed in this report it will be represented by the synchronous speed of the travelling magnetic field.

$$\omega = V_{syn} = \frac{2\pi f}{\phi} \qquad \text{Equation 7.3}$$

7.1.4 Efficiency

The expression for efficiency to be used is stated in equation 7.5.

$$\mu = \frac{\dot{W}_{output}}{\dot{W}_{input}} \qquad \text{Equation 7.4}$$

$\dot{W}_{output}$, or useful power output, can be calculated as the product of torque and driving frequency.

$$\dot{W}_{output} = \tau\omega \quad \text{Equation 7.5}$$

$\dot{W}_{input}$ can be calculated as the product between V_{RMS} and I_{RMS}.

$$\dot{W}_{input} = V_{RMS}I_{RMS} \quad \text{Equation 7.6}$$

7.1.5 Uniformity

Uniformity, as a ratio between two measurements of the same quality at different positions, could be calculated with any vectored performance characteristic. For this project angular acceleration was used for this purpose.

To calculate uniformity it was proposed that the magnitude acceleration be measured in two defined locations, one measurement being made on a principle plane and the other measurement being made on a plane that bisected the principle plane and an adjacent principle plane. Principle plane is used here to describe a plane in which torque is applied to the rotor by a stator or other similar construct.

7.2 Specific Metrics

Specific metrics will take the form of whatever parameters or metrics the maker believes will help them design or optimise most effectively. Comparison between designs based on similar technologies using specific metrics is plausible but not intended.

As both motor technologies operate on similar principles, the specific metrics proposed for use in this project were developed to be applicable for both the linear induction and permanent magnet synchronous variants of the eponymous spherical motor.

After consideration of theory & the academic literature it was decided that the quantities of magnetic flux linkage Φ and Goodness factor G would form the specific metrics for the design process.

7.2.1 Goodness Factor

The Goodness factor was originally proposed by Laithwaite in a well-received 1965 paper, "The Goodness of a Machine". G is a dimensionless expression of efficiency for an electric

motor, sometimes termed as the "magnetic Reynold's number". G was developed for application to induction motors, both rotary and linear. However it is applicable to any motor dependant on the principles of electromagnetism to actuate, representing a ratio between the angular velocity of the travelling magnetic wave and the product of the operating electrical resistance and magnetic reluctance.

$$G = \frac{\omega}{reluctance \times resistivity} \qquad \text{Equation 7.7}$$

The exact manifestation of the equation for Goodness depends on the topology of the motor.

However, in all cases G is dependant on 5 parameters: $\omega, \mu_0, p, \rho_r, g$, in the manner depicted in equation 7.8.

$$G \propto \frac{\omega \mu_0 p^2}{\rho_r g} \qquad \text{Equation 7.8}$$

For the case of a slow-moving linear induction motor, Boldea proposes that the motor is designed to achieve a Goodness factor that has a value reciprocal to that of the intended operating slip. [66]

$$SG = 1 \qquad \text{Equation 7.9}$$

The objective of design process described in 7.9 is to reduce the amplitude of end-effects that can significantly reduce the performance of linear induction machines. [66] However, because the spherical motor has a secondary with an effectively infinite length, the assumption can be made that end effects are significantly reduced and the design for Goodness can be conducted as for any rotary induction machine. That being bigger is better.

7.2.2 Magnetic Flux Linkage

Magnetic flux linkage is the magnitude of flux, described by the magnetic vector field $\overrightarrow{B}$, that is acting normal to a surface. The change in this quantity over time is the driving mechanism behind the principle of induction, as described in the theory section.

Therefore designing for maximum Φ is an effective method for producing an efficient electric motor.

8 OPTIMISED DESIGN

8.1 Independent Design Parameters

The analytical design tool was used to propose optimal values for the key design parameters. Values for the other design parameters were informed by the measurements described in "Shell-like spherical induction motor" project conducted by Fernandos et al. [59].

In order to maintain consistency with the key parameter of Scale Factor, all imported design parameters that denoted physical dimensions were scaled by a factor that represented the ratio between the radius of the rotor in their original designs and the baseline rotor radius in this project, defined as $50mm$.

The values of all the design parameters can be found in the nomenclature section.

8.2 Dependent Design Parameters

In addition to the output performance characteristics, the design tool also outputs parameters that describe the form of the motor as implied by the given set of independent design parameters. These dependant design variables are all functions of the independent variables, but will help to aid the actual design of any motor proposed by the use of the design tool.

8.3 Optimisation of Key Parameters

The selection of key parameters in this manner would change the functions of performance from equations of 16 variables.

$$\tau = f(l, R_{in}, \phi, p, \delta airgap, F_{bf}, \Phi, \delta_{shell}, \omega, I_{peak}, V_{peak}, R_0, R_{load}, Slot_w, Slot_h, R) \quad \text{Equation 8.1}$$

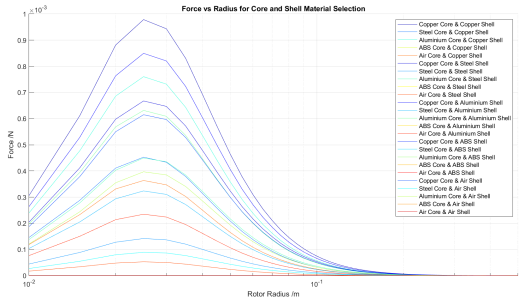
To a function optimised in terms of four variables.

$$\tau = f(SF, M_{core}, M_{shell}, AWG) \quad \text{Equation 8.2}$$

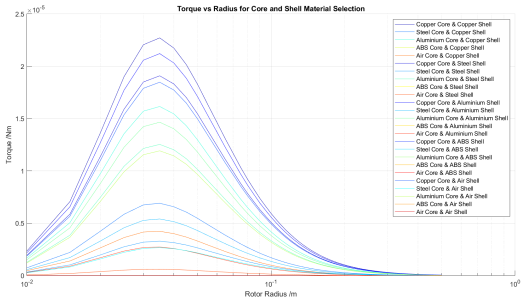
The design tool was used to find the combinations of key parameters that would maximise

the function of 8.2. This was done manually, by observation of the graphs produced by the design tool. Each graph presented 3 key parameters at any one time, the fourth key parameter would be held constant.

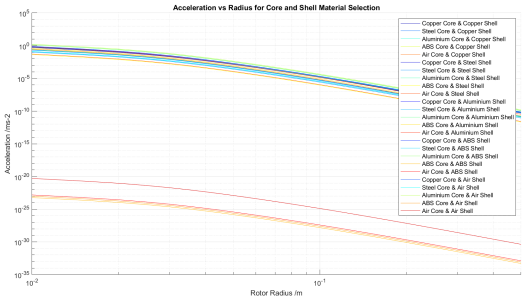
8.3.1 Scale Factor and Rotor Material



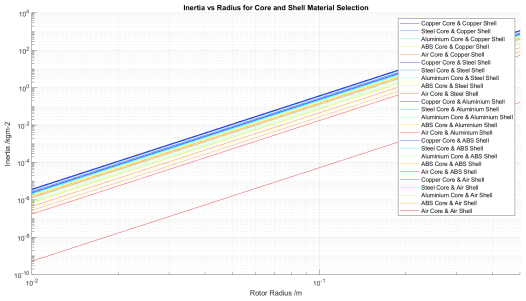
(a) Force vs Radius



(b) Torque vs Radius



(c) Acceleration vs Radius



(d) Inertia vs Radius

From figure 8.1a and figure 8.1b it can be seen that optimal results are achieved when the rotor radius is approximately $25mm$ in radius and $50mm$ in diameter. Figure 8.1c indicates that the acceleration characteristics will always favour a smaller radius, owing to the effect on rotor inertia.

$$SF = 0.5 \qquad \text{Equation 8.3}$$

Likewise for rotor material it can be seen that optimal force and torque are both achieved with a rotor of copper core and copper shell. Acceleration is more complex however, the graphs produced imply that many rotors will perform better with no shell material (air).

This is maybe true, but such readings actually present a variation in rotor radius as opposed to material and have been discounted.

Values of inertia were instead considered. Consulting figure 8.1d revealed that the motors with the lowest inertia were the aluminium shell, air core and copper shell, air core.

$$M_{core} = Air \quad \text{Equation 8.4}$$

$$M_{shell} = Aluminium/Copper \quad \text{Equation 8.5}$$

One note on the choice of M_{core} is that the assumption of negligible flux fringing is only considered valid in the presence of a ferromagnetic material.

8.3.2 AWG

The gauge of wire used will directly affect the number of turns that can used per coil winding, and so directly affect the current density. However, an increased number of coil turns will also increase the coil impedance, rendering the current density constant as seen in figure 8.5.

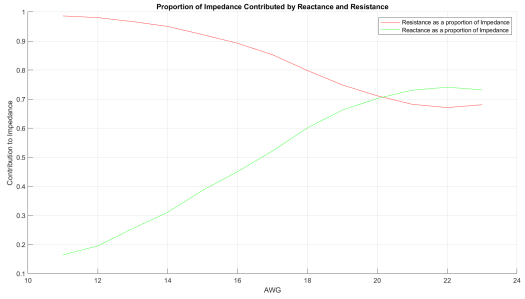


Figure 8.2 – Impedance, Reactance and Resistance Ratio

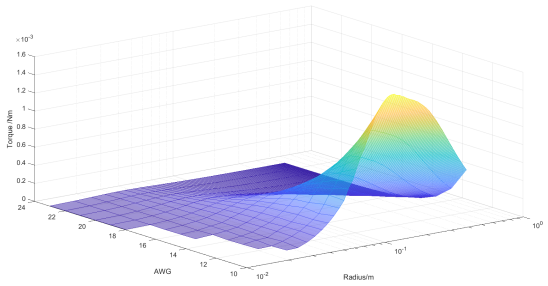


Figure 8.3 – Torque, AWG and Rotor Radius Surface

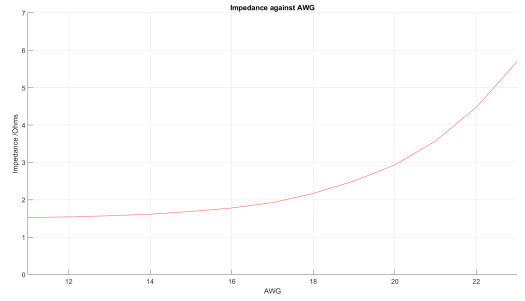


Figure 8.4 – Impedance

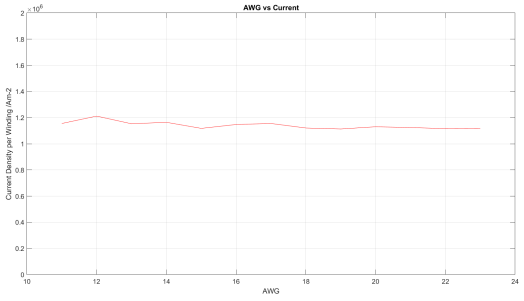


Figure 8.5 – Current Density

8.3.3 Permanent Magnet Grade

The design tool was used to predict the difference between the contribution of each coil on the stator to the magnetic field strength to the contribution from one N45 permanent magnet measuring $5mm^3$.

9 DESIGN FOR MANUFACTURE

9.1 Concept Generation and Design Selection

For the specified design parameters there are any number of physical manifestations. Four design concepts were generated, drawing from both linear induction and permanent magnet synchronous technologies.

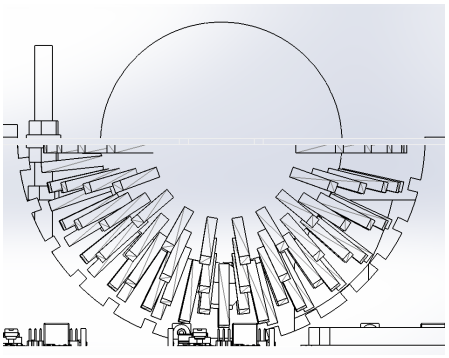


Figure 9.1 – Porcupine stator

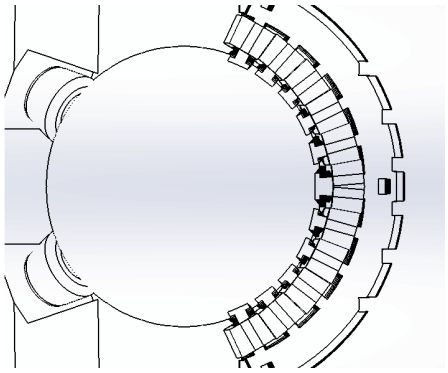


Figure 9.2 – Two 180 degrees stators

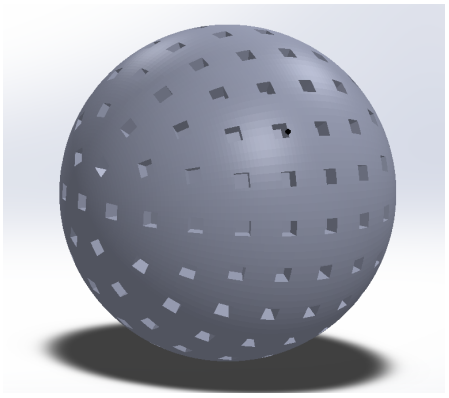


Figure 9.3 – Permanent magnet motor, for use with any stator

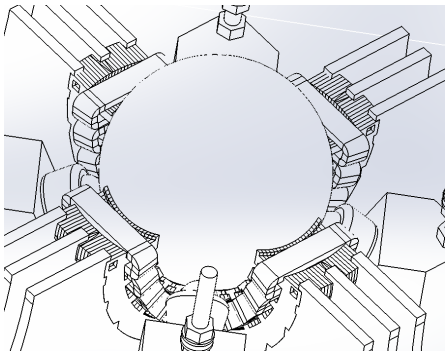


Figure 9.4 – Four 90 degree stators

Due to the difficult manufacturing circumstances however, all concepts that heavily required the use of 3D printing had to be removed from consideration.

In the selection process between the two $180°$ stators concept and the four $90°$ stators concepts, the four $90°$ stators concept was ultimately selected by the virtue that the packaging of the four $90°$ stators would have been simpler, as the stators could all be mounted on identical braces.

9.2 Covid 19

On the 18th March the university closed down the IForge, and subsequently closed its doors in line with the UK's social distancing policy.

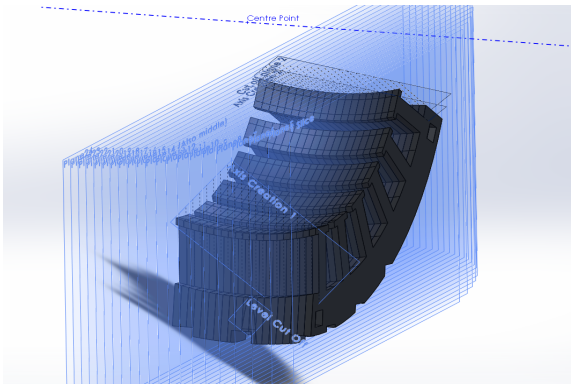
With few means of manufacture available it was decided to secure an increase in project budget and enlist third party companies to fulfill the roles that the IForge laser cutter and water jet cutter were supposed to have filled. The reminder of the manufacture was limited to the tools and supplies typically found in a 4th year engineering student house, in addition to local essential shops. The designs that were supposed to be produced can be found in Appendix D.

In short the project scope did not change. The objective of the prototype model was to provide a means to test the proposed omnidirectional performance characteristics.

9.3 Stator

The chosen design concept features four 90° stators. A primary focus of design was to ensure that the assumption that current would flow tangentially to the surface of the spherical rotor would remain valid for a physical prototype.

To maintain the assumption of tangential current it was proposed that the inner surface of the stator would always be tangential to the rotor also. To this end a 2D stator profile was positioned tangentially to a 2D section of the rotor and then revolved around the vertical axis of the rotor. To increase the positional tolerance of the stators within the final assembly, a Boolean operation was performed such that the walls of the stator remained parallel.



(a) CAD Stator divided into sections

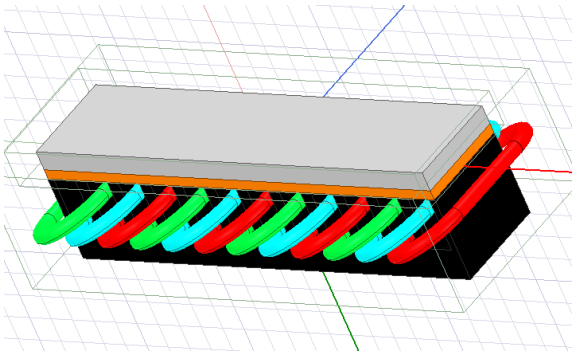
(b) Covid-19 Manufacture

For manufacture, the stator design was divided into 20 sections, each of 2mm thickness, such that profiles could be manufactured using the waterjet cutter in the IForge. The tech-

nique used to bond the 20 steel sections together would be selected on the criteria of adhesion and resistivity. A resistivity criterium was specified to allow the bonding method to serve the dual-purpose of adhesion and lamination. This was considered to reduce the amount of eddy current induction and associated losses, justifying the model assumptions.

9.3.1 Windings

16 AWG copper enamelled wire was used to construct the windings. The simple winding pattern employed by Laithwaite in 1965 was employed. A deviation was made from this scheme



(a) Previous ANSYS Maxwell model detailing winding arrangement

(b) Winding process during Covid 19

in order to connect the four stators into orthagonal pairs. This allowed the windings to be continuous across pairs, at the cost of an irregular magnet field. During the winding process it was found that only 12 turns of wire could be wound on to the stator, in contrast to the specified 20. Future work might consider flexural rigidity as a wire parameter.

9.4 Rotor

The design and manufacture of the rotor would be split into two sections, the shell and the core. For the rotor developed by Fernandes et al. in 2017 [67], both the core and shell sections of the rotor were constructed through the use of subtractive milling to construct two approximately hollow hemispheres per rotor section. Each hemisphere was then grounded to a fine tolerance using a spherical ball mill. The rotor shell was constructed from copper and the rotor core constructed from a Silicon-Iron composite designed to reduce eddy currents in all directions, at the cost of relative μ.

For construction of the rotor for this project it was considered that any spherical machining operation that involved metal would not be feasible for use in the project. However many model assumptions and principles depended on the rotor being spherical in form.

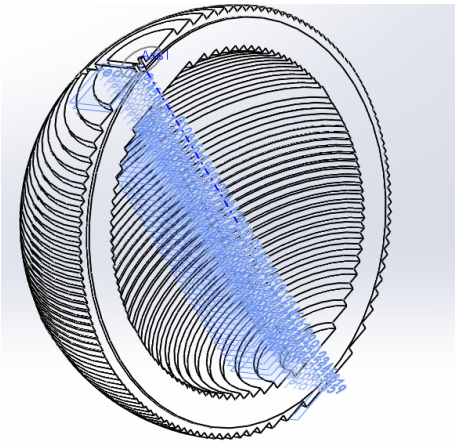
Requests were made to specialist companies such as MetalSpheres.com, but the quotes were too expensive. It was decided that the rotor shell would need to be constructed from a mass-produced commercial item.

The only item to meet this specification was a 4.5"$(\ 115mm)$ copper float valve listed by BES plumbing, retailed at £7.96.

An electric sander in the IForge was used to remove the seam of the float valve and thereby create two copper hemispheres. A micrometer was used to take measurements of the thickness of the wall-thickness, which was found to be approximately $1.5mm$. A second float valve was purchased in order to double the shell thickness of the rotor.

9.4.1 Rotor Core

Several rotor core substitutes were also considered, though none were considered feasible. After discussions with Jamie Booth, a senior technician, a design was proposed whereby 57 cross sections of the hollow sphere would be cut out of 2mm mild steel using the waterjet cutter in the IForge. Notches would be included in the design profiles to produce steel sections with a defined weld channel. This would allow the technicians to weld seams around the steel sections and thereby create an approximate hollow sphere out of steel.



(a) 56 slice $10mm$ thick rotor core

(b) 136 unlabelled profiles

9.4.2 Support Structure

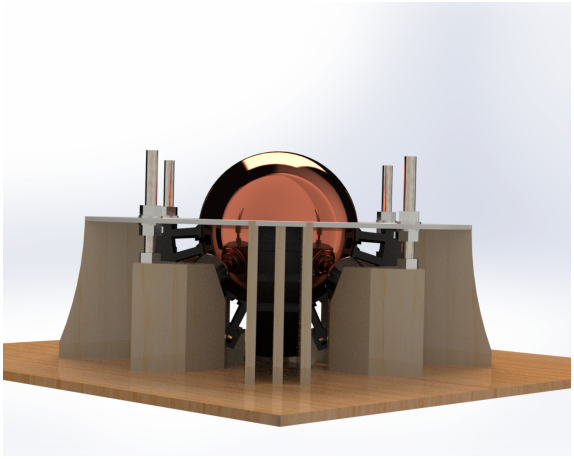
The support structure was designed to provide control over design parameters such as airgap and rolling friction.

To reduce cost, the components were designed for production via an IForge laser cutter.

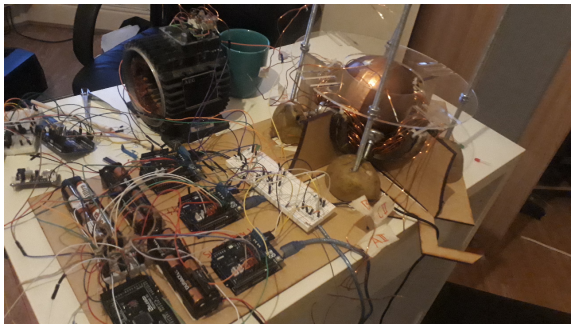
The use of interference slots and mechanical fastening for assembly was justified by a low predicted mechanical load. Specific bearing support blocks were also designed for additive manufacture with the IForge 3D-printers. The support blocks were designed such that the rotor could be suspended, at the airgap distance, above the stators with minimum friction on five force transmission bearings.

9.4.3 Prototype

A prototype was produced. The pictures indicates a comparison a render and the finished prototype.



(a) Render of proposed design



(b) Prototype produced under lockdown

9.5 Permanent Magnet Synchronous Motor

The scope of the project was originally expanded to include the construction of a permanent magnet synchronous variant only on the condition that few additional resources would be required. Therefore the spherical PMSM motor components were designed to be almost entirely common with components of the spherical LIM, with the exception of the rotor.

9.5.1 Permanent Magnet Synchronous Rotor

The spherical PMSM design was adapted from the design of a rotary fractional-slot PMSM. A teeth:pole ratio of $12:14$ was adopted from an open-source design for a permanent magnet DC motor for use in powered scooters.[68] The ratio was raised to the second power to adapt it for use in a spherical motor.

In the literature review it was discussed that Chirikjian et al. had been forced to overlay platonic geometries over rotor sphere in order to approximate a uniform distribution of points [49]. For this project it was decided to adopt an approximate numerical solution.

This was achieved through the use of Blender, a polygon based 3D-modelling software, and a Python add-on for blender written by a StackExchange.com user with the alias of "Lemon". The python add-on, entitled "Homogenous_Sphere.py", would produce a sphere of a defined radius and distribute points across it that would maintain an approximately isotropic distance from each other.

The number of points added and the spacing between them was defined by the number of points to be placed on the equator. This was considered analogous to the number of poles on a stator. 187 points were specified for a sphere of 50mm radius and 28 teeth.

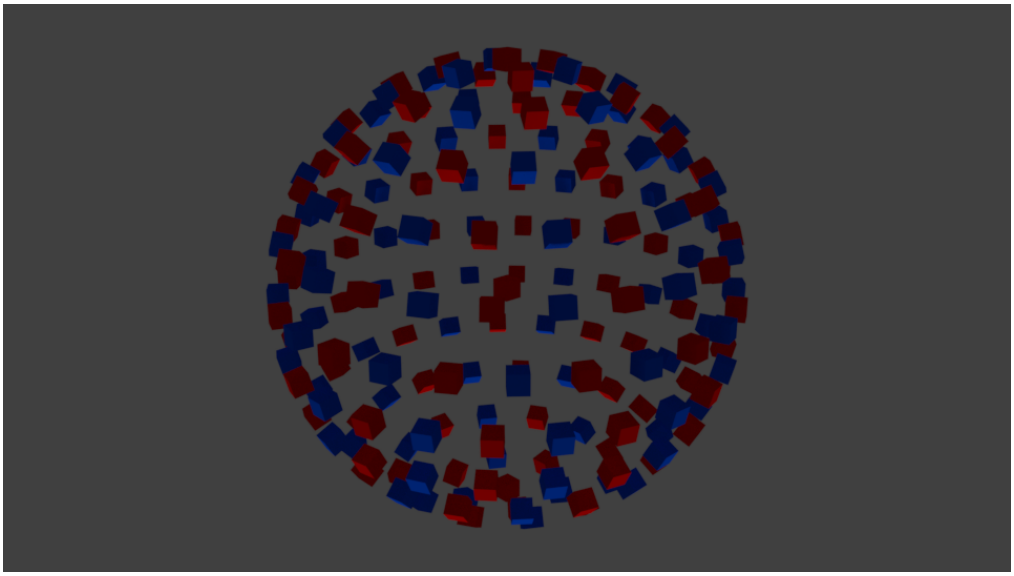


Figure 9.9 – Magnet layout (N/S) for PMSM

It was decided to use magnets of $5mm$ in diameter, to match the stator wall thickness. $5mm$ Neodymium N45 cube magnets were purchased. A representative magnet geometry was constructed in Blender, which incorporated a $0.2mm$ interference fit, and the geometry was duplicated across all points specified by Lemon's script. The model was adjusted such that each magnet was embedded in the rotor, a face of each magnet was tangential to the surface of the sphere and that a the outward faces formed a lattice of opposing polarities.

The file was converted to .STL file format, the intention being to manufacture additively with the IForge printers. This limited material selection to ABS plastic.

10 TESTING METHODOLOGY

In order to generate the metrics required for validation of the analytical tool and also characterisation of the prototype, a number of required physical measurements were defined. The methodologies used to obtain the required metrics, and selection thereof, are discussed in this section.

10.1 Sense Rig

Many of the measurement methodologies described in this section depend on the use of sensors. For such measurements the sense rig was used to facilitate the measurements. The sense rig, figure 10.1, consisted of a single Arduino Uno, connected to any relevant sensor, with data output via a serial port to a data logging program in MATLAB and an LCD Screen. The screen was included as a means for the experimenter to check readings in real time for efficiency.

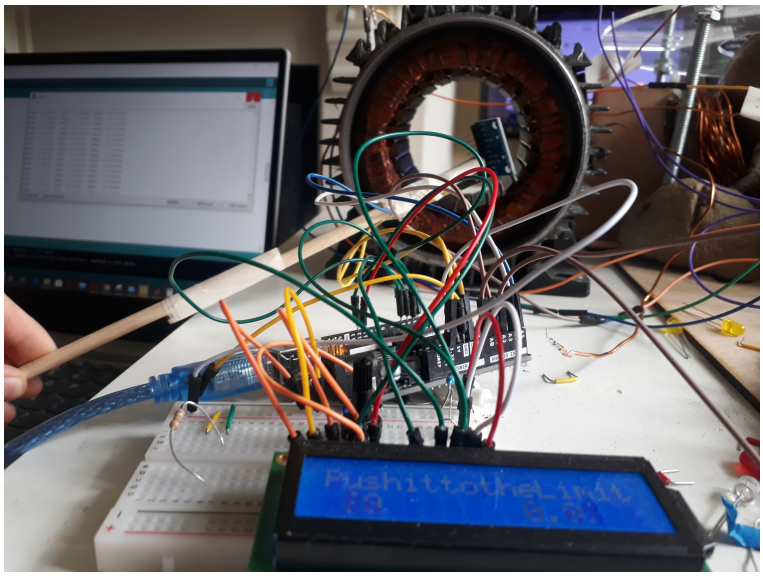


Figure 10.1 – Sense Rig

10.1.1 Peak Current

The Arduino motor shields each posses a current sensing capability that outputs to an external pin.

10.1.2 Peak Voltage

The assumption was made that peak AC voltage is equal to the rated voltage of the DC power supply.

10.1.3 Flux Density

The measurement of flux density was achieved by locating a magnetometer in place on the stator and the exciting the windings appropriately.

Two magnetometers were considered:

- SODIAL hall effect sensor.
- Paradisetronic MPU-9250 Drone IMU.

The hall effect sensor was selected. Calibration of the device proved difficult however, there was significant drift in the readings given, the extent of which varied with time.

As a result the MPU- 6265 Drone IMU chip was chosen. The IMU contained a purpose built magnetometer with the ability to sense a magnetic flux density of up to $\pm 4800 \mu T$ in accordance with the datasheet. [69]

10.1.4 Angular Acceleration

A measurement of angular acceleration was obtained by measurement of the magnitude of acceleration experienced by an element of the rotor at a defined radius R. The measurement was achieved through the use of a mechanical fastening to attach an accelerometer to the defined element.

The MPU-9250 chip also contained an accelerometer of suitable specification for the purpose of motor characterisation. On those grounds it was decided that the GY-9250 accelerometer would be used for the purpose of measuring acceleration.

For calculation of uniformity, the measurement of α would be obtained in the same manner but with more defined placement. For $\alpha_{principle}$ the accelerometer would be attached to the rotor directly above the central lamination of the stator. For $\alpha_{bisector}$ the distance between the centre lines of the stators would then be measured. The accelerometer would then be fastened to the rotor at a point directly in line with the halfway measurement.

11 RESULTS

In order to calculate the generic and specific performance characteristics of the prototype, a series of tests were proposed in order to obtain the measurements specified in the System Analysis section. The methodologies proposed for use are discussed in the Testing section.

Upon completion of the prototype and initial testing, it was discovered that the prototype did not actuate, thereby preventing effective characterisation.

11.1 Characterisation of a Dysfunctional Omnidirectional Drive System

As a result of the dysfunctional motor, of the five generalised characteristics (Inertia, Peak Torque, Efficiency, Driving Frequency & Uniformity), only Inertia and Driving Frequency could be calculated.

Of the specific metrics, only flux linkage could be measured.

Consequently, it was decided to use the measurement of magnetic field strength calculated for the prototype rotor in order to calculate the magnitude of force and the torque that would have been generated if the prototype's physical dimensions had not been altered from the design parameter specifications, by way of the manufacturing process.

This was performed by, firstly, calculation of the magnetic field strength, B, that would give rise to the value of flux density that was measured at at the distance of the air gap from the inner radius of the stator.

Secondly, the value for B would be substituted back into the design tool, along with the prototype design parameters. This was performed in order to predict the output characteristics that could have been expected if the physical dimensions of the prototype had exactly matched the intended design parameters.

Values for the generalised characteristics would then be proposed on that basis.

The largest value of flux density, found at a frequency of $5Hz$ and a $\delta airgap$ of $0.002m$, was $2.8 \times 10^{-3}T$.

The value for B was substituted into a purpose-built section of the design tool

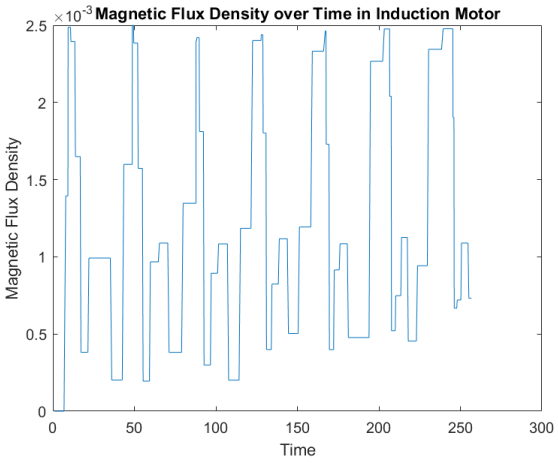


Figure 11.1 – Prototype Flux Density at 1Hz 12V AC

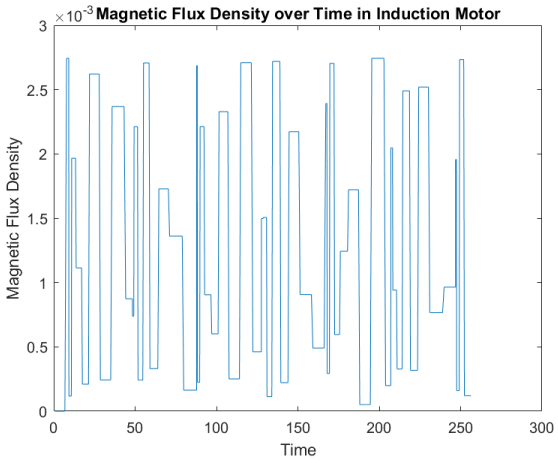


Figure 11.2 – Prototype Flux Density at 5Hz 12V AC

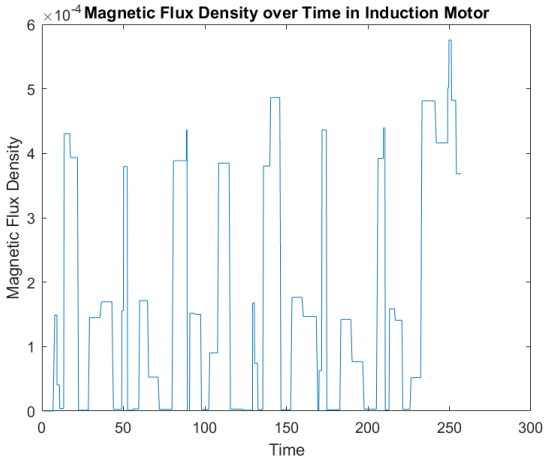


Figure 11.3 – Prototype Flux Density at 10Hz 12V AC

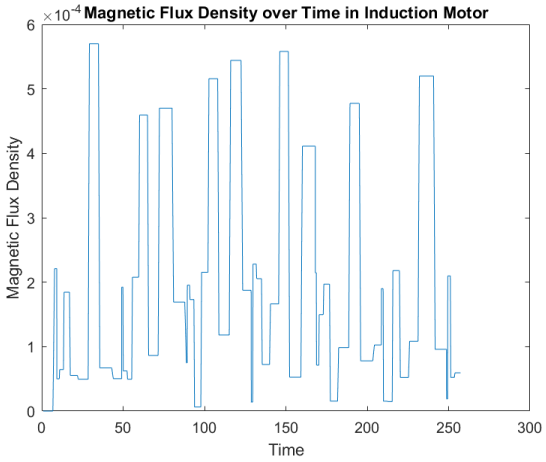


Figure 11.4 – Prototype Flux Density at 30Hz 12V AC

$$B(\delta airgap) = 2.8 \times 10^{-3}T \qquad \text{Equation 11.1}$$

Prediction of Prototype Performance Characteristics					
Key Design Parameters		Notable Design Parameters		Generalised Characteristics	
$Mcore$	Air	δ_{airgap}	$0.002m$	I	$0.0094kgm^2$
$Mshell$	Copper	δ_{shell}	$0.003m$	τ	$2.93 \times 10^{-8}Nm$
SF	1.15	$F_{bearing}$	$0N$	V_{sync}	$0.7854rads^{-1}$
AWG	16	N	12	α	$3.11 \times 10^{-6}ms^{-2}$
$B(r)$	$5.6 \times 10^{-6}T$			U	1
f	$5Hz$				

Table 11.1 – Predicted performance characteristics of a spherical drive system producing the same airgap flux density as the prototype

12 ENGAGEMENT WITH OPEN-SOURCE COMMUNITY

12.1 Hackaday

A Hackaday blog was started on the 9th January 2020 as means to provide both a convenient method of project documentation and also to raise awareness of omnidirectional drive systems in the open source community, in line with an objective outlines in the project scope.

To this end the project blog received 898 views and 3 likes. Although these statistic are not the largest on Hackaday they may be indicative of 898 potential engineers, designers or makers who may now go on to consider the use of omnidirectional drive systems just a little bit more seriously than they might otherwise have done.

The project page can be found at the web address:
https://hackaday.io/project/169355-open-source-spherical-motor

12.2 YouTube

To promote the Hackaday blog and enable more makers in the open-source community to utilise the technologies that were researched and developed in this project, a YouTube video was produced.

Entitled ”DIY Spherical Motor: Arduino Motor Controller and SPWM”, the video was intended to educate any viewer, specifically, about the principles of SPWM and how to construct the motor controller.

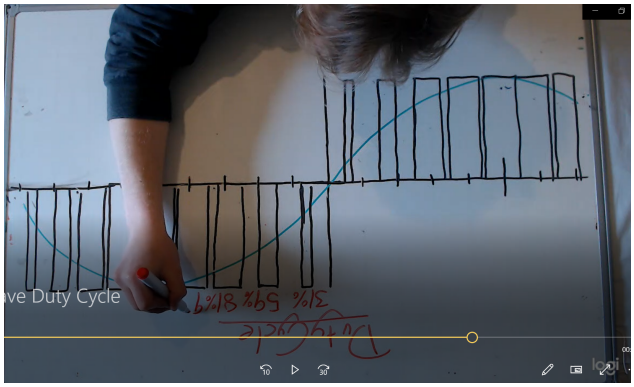


Figure 12.1 – Frame from Video

13 DISCUSSION

In summary,

1. A purpose for open-source, arduino-based omnidirectional drive systems was established.
2. A configuration of omnidirectional motor, and appropriate technology, was selected.
3. Design was conducted with open-source and arduino-based replication in mind.
4. A prototype was manufactured.
5. The process was shared in the open-source community.
6. An attempt was made to characterise the prototype according to the generalised performance characteristics that had been proposed.

.

It can be seen that although most of the aims and objectives laid out at the start of the project were met, the production of an operational spherical motor was not achieved.

Additionally, it can be seen from the results in the previous section that, even if a perfect manufacture had taken place, the produced motor would still likely not have worked.

Certain parameters necessitated by manufacture had some role to play, including the reduced shell thickness and limited number of coil turns, but fundamentally it is thought that the power source and winding coil selected were not adequate for the purpose.

13.1 Future Work

The design tool proposed, as validated during the course of the project, is considered to be an asset that was not fully utilised during the course of the project. Therefore, like the rest of the project, it will be made available to the open-source community.

It is recommended that the design tool be combined with an optimising software such that all design parameters can be altered, not just designated key parameters.

The first step is often the hardest and it is hoped that any future work can benefit from the paths that have been journeyed upon and described throughout this project.

14 APPENDIX A

14.1 Motor Controller Theory

The Arduino motor shield and Arduino Mega selected for the motor controller are both micro-controllers. Reliance on electronic switches (transistors) dictates that all outputs from the Arduino components are binary in form. The term used to describe a motor controller that operates through the use of electronic switches and timer interrupts is chopper drive.

A chopper drive can approximate a required waveform through the use of PWM. Over any one PWM time period the duty cycle during that time period will correlate exactly to the percentage of V_{max} present in the output signal. Therefore the change in duty cycle with respect to time will be reflected in the change in output voltage with respect to time. However, only an approximate waveform can be generated because the duty cycle is a discrete quantity i.e. the duty cycle is discontinuous between PWM periods.

Manipulation of the duty cycle such that that the change in duty cycle with respect to time resembles a sinusoidal function results in an approximate output AC power signal of a frequency that matches the frequency of an equivilent sinusoidal function. This provided the basis for generation of an AC output.

The achievement of three phase AC power production initially required production of three synchronous, but otherwise independent, SPWM signals. Once this had been achieved, the $\pm 120^{\circ}$ phase shift was achieved by adjustment of the relative start times of two of the signals.

14.1.1 SPWM Production

An Arduino, capable of outputting either a HIGH (1) or LOW (0) signal is not capable of generating an SPWM signal because it cannot enable production of the negative signal that would be required for half of the waveform. The sign of the signal is dependant on the direction of the flow of current, but in an Arduino all HIGH signals flow to GND.

The relative direction of current flow can be altered through the use of electronic switches in order to alter the position of HIGH and GND relative to the circuit. The most common

configuration of transistors that is capable of performing this function is referred to as an H-Bridge. So named due to the visual similarities the circuit shares with the letter $'H'$.

The L298 Motor controller is widely used integrated circuit that features an H-Bridge that utilises MOSFET transistors. The Arduino Motor Shield Rev 3 is itself an L298 Motor controller that has been integrated into a Arduino shield for use of use with the Arduino platform.

15 MICROCONTROLLER THEORY

The Atmel ATMEGA 328 is the microprocessor chip that forms the core of the Arduino Uno. Within the microprocessor memory are contained the bit registers, each an 8-bit libraries storing the 8 pieces of binary information. Bit register are used for storing variables, an 8-bit register is capable of storing variables of size up to 2^8 ,256. Bit registers can be combined to allow storage of variables up to 2^8n, where n is the number of 8-bit registers involved.

Beyond the storage of variables, the bit registers are also used to change the behaviour of the output signals. Specific bit registers are associated with a number of grouped functions. Each function associated with an individual bit within the bit register.

The bit registers used were PORTX, DDRX, TCCRnA, TCCR1nB, OCRnA, OCRnB, TIMSK and TCNTn.

15.0.1 Lower Frequency Limit

The PWM signal relied on the the Arduino inbuilt 16-bit timers to operate. However, in default operation of the ATMEGA328, the timer counts with each each complete clock cycle of the 16MHz crystal oscillator within the Arduino. The 16 bit Timer1 can store variables of up to 65,536. It was clear that in this setting the timer would roll over before one AC time period had elapsed. This was resolved by assigning a value of 1 to the CS12 bit and 0 to the CS11 & CS10 bits contained within the the TCCR1B register. This applied a prescaler of 256 to the internal clock and caused Timer1 to count after 256 systems clocks had elapsed. This meant that just over 1 second would elapse before Timer1 rolled over. This would allow for a minimum frequency of 1Hz.

15.0.2 Upper Frequency Limit

The upper limit of frequency for the approximated AC power signal is determined by the time taken to access and assign the new duty cycle to the PWM output pin and the time taken for PWM interval to occur.

Resolution of the PWM signal can be increased, and consequently the PWM interval decreased, without significant loss of signal accuracy, until the point where the time taken to execute the loop exceeds the PWM interval. The output signal will no longer approximate a sinusoidal function at this point. The optimum resolution per frequency can be calculated by determining the resolution that corresponds to a PWM interval exceeds 1 by the smallest margin.

15.0.3 Interrupts

Interrupts enable the microprocessor to interrupt the actions executed by the main body of code and execute their associated ISR actions in place of the looped actions.

The ATMEGA 328 microprocessor supports three types of timer interrupts. These are External Flag, Overflow and Input/Output Compare Match interrupts. Each interrupt is triggered by a different stimuli. External Flags interrupts are triggered by the detection of a signal on a specific pin, Overflow interrupts are triggered in the event of a specific timer exceed its count limit and rolling over to zero. Input/Output Compare Match interrupts continually compare the value of the timer value, stored in TCNTn, with the value stored in the the Output Compare register, stored in OCRnA.

A design matrix was created to select the type of interrupt that would be used to initiate the change in current direction. The Input/Output Match Compare interrupt was chosen.

15.0.4 Component Selection

Controller	Power Supply	Inverter	Image	Supply Characteristics	Frequency Control	Risk		Arduino Compatibility	Cost	Tested?	Single Phase Gen Success?	Three Phase Gen Success?	Final Design
Arduino Uno	9V Battery	L298N Motor Controller		9V DC | 500 mAh	SPWM	Low		Yes	~£20	Yes	Yes	No	No
Arduino Uno	Bangood SGS200	L298N Motor Controller		12V DC | 20800 mAh	SPWM	Low		Yes	~£30	Yes	Yes	No	No
Arduino Uno	L298N Motor Controller	Arduino Motor Shield Rev 3		12V DC | 20800 mAh	SPWM	Low		Yes	~£40	Yes	Yes	No	No
None	Mains Power Supple	Tektronix Waveform Generator		12V DC/AC | ∞	In-built Variable Frequency Drive	Low		None	~£80	No	N/A	N/A	No
None	Mains Power Supply	None		240V AC | ∞	None	V. High		None	~£0	No	N/A	N/A	No
Bangood SGS200 SPWM Board	9V Battery	Arduino Motor Shield Rev 3		12V DC | 20800 mAh	SPWM	Medium	Undocumented Hardware, despite low voltages	Possible	High	Yes	No	No	No
None	8 1.5V AA Batteries in series	Op-Amp Phase Oscillator		12V DC | 20800 mAh	Oscillator Circuit	Medium	Exposed Circuitry	None	High	Yes	No	No	No
Arduino Mega	8 1.5V AA Batteries in series	Car Battery		12V DC | 45000mAh	£60.00	Low	Medium Voltage , documented hardware	Yes	Low	Yes	Yes	Yes	Yes

Figure 15.1 – Component assessment for purchase and preliminary testing

16 APPENDIX B

```
1
2  #define Divisions (200)// Sub divisions of sisusoidal wave.
3  static unsigned int lookUp_A[Divisions];
4  static unsigned int lookUp_B[Divisions];
5  static unsigned int lookUp_C[Divisions];
6  //Preparing each look up table as an array with "Divisions" number of
       entries.
7
8  //Defining values that will be used later
9
10 static int MHz = 16;                                    // Micro clock
       frequency
11 static int freq = 10;                                    //
       Sinusoidal frequency
12 uint32_t DutyCycle;                                     // The
       length of each PWM Duty Cycle
13 uint32_t period;                                        // Period of
        each sinosoid waveform
14 uint32_t halfperiod;                                    // Half of
       the period of each sinosoid waveform
15 uint32_t Third;                                         //Also known
        as 1/3 AKA 0.333333333333333333333
16 uint32_t TwoThirds;                                     //Also known
        as 2/3 AKA 0.666666666666666666667
17 uint32_t X;                                             //Used as a
       counting variable in the loop for Phase A
18 uint32_t Y;                                             //Used as a
       counting variable in the loop for Phase B
19 uint32_t Z;                                             //Used as a
       counting variable in the loop for Phase C
20 float ClocksperSecond = 62500;                          // Including
        pre-scaler, how many system clocks will occur each second
21 float ClocksperPeriod = ClocksperSecond / freq;         // How many
       clocks will happen per period of sinosoid waveforn
22 float ClocksperCycle = 65536;                           //System
       clocks per system cycle. Will only change with the pre-scaler.
23 float PeriodClockFraction = ClocksperPeriod / ClocksperCycle; //What is
       the fraction of the duty cycle against length of the system cycle?
       Measured in system clocks.
24 const double currentFactor = 2 / 3.3;                   // 3.3V per
       2A which is 0.909
25
26 // Uno Pins
27 //If you want to run single-phase AC on an Arduino Uno (and Arduino
       Motor Shield Rev 3)
28
29 //[Physical Pin Number] & Pin as denoted on Arduino IDE
30 //const int MotorPinA = PB4;                           //Pin [18]
       12
31 //const int MotorSpeedPinA = PD3;                      //Pin [5] 3
```

```
32 //const int MotorBrakePinA = PB1;                                    //Pin [15] 9
33 //const int CurrentSensePinA = A0;                                     //Current
      Sensing
34
35 //Mega pins
36 //If you want to run three-phase AC on an Arduino Mega (and three
      Arduino Unos and three Arduino Motor Shields Rev 3's )
37
38 //[Physical Pin Number] & Pin as denoted on Arduino IDE
39 const int MotorPinA = PH1;                                           //Pin [13] &
       16 MotorPinA for MotorShield switches the direction of the current
      and generates the negative component of the waveform
40 const int MotorSpeedPinA = PH6;                                     //Pin [18] &
       9 MotorSpeedPinA for MotorShield provides a PWM signal, and will
      allow modulation of the duty cycle
41 const int MotorBrakePinA = PH5;                                    //Pin [17] &
       8 MotorBrakePinA for MotorShield provides will cut the signal if
      necessary (i.e. Circuit breaker with CurrentSensing)
42 const int CurrentSensePinA = A0;                                   //Current
      Sensing
43
44 const int MotorPinB = PA5;                                          //Pin [73] &
       27
45 const int MotorSpeedPinB = PB7;                                     //Pin [26] &
       13
46 const int MotorBrakePinB = PA0;                                     //Pin [78] &
       22
47 const int CurrentSensePinB = A1;                                    //Current
      Sensing
48
49 const int MotorPinC      = PL7;                                     //Pin [42] &
       42
50 const int MotorSpeedPinC = PB6;                                     //Pin [25] &
       12
51 const int MotorBrakePinC = PL6;                                     //Pin [41] &
       43
52 const int CurrentSensePinC = A2;                                    //Current
      Sensing
53
54
55 uint32_t compA       =  PeriodClockFraction * ClocksperCycle / 2;
                        //Interrupt compare value A, (halfway through cycle
      )
56 uint32_t compB       =  PeriodClockFraction * ClocksperCycle;
                        //Interrupt compare value B, at the end of the
      cycle
57 uint32_t TimerBStart =  PeriodClockFraction * 0 * ClocksperCycle;
                //Offset for Phase A
58 uint32_t TimerCStart =  PeriodClockFraction * ClocksperCycle * 1 / 3;
                //Offset for Phase B
59 uint32_t TimerAStart =  PeriodClockFraction * ClocksperCycle * 2 / 3;
                //Offset for Phase C
60
```

```
61
62  // Counter and Compare Values
63
64  //For Mega
65  uint32_t t3_load = TimerAStart;
66  uint32_t t3_compA = compA;
67  uint32_t t3_compB = compB;
68
69  uint32_t t4_load = TimerBStart;
70  uint32_t t4_compA = compA;
71  uint32_t t4_compB = compB;
72
73  uint32_t t5_load = TimerCStart;
74  uint32_t t5_compA = compA;
75  uint32_t t5_compB = compB;            //seems you can only really trust a
        uint32_t, dunno why int const in and float didnt work :/
76
77  void setup() {
78    cli();                              //stop interrupts while we set up
       the timer
79    Serial.begin(9600);                 //seial monitor initialized, use
       Serial.print to see what is happening to variables within the sketch.
80
81    double temp;                        //Double varible for <math.h>
       functions.
82    period = ClocksperPeriod;
83    halfperiod = period / 2;
84    DutyCycle = period / Divisions;     //Period in Microseconds, remember
       it has a resolution of 4.
85
86
87    //Lookup Table for Phase 1/A
88    for (int i = 0; i < Divisions; i++) {                    // Filling
       each value in the array as its corresponding index. i.e. [1] = 1 [2]
       = 2 [3] = 3 etc.
89      temp = abs(sin((i * 2 * M_PI / (Divisions)))) * 255;  //Values in
       array now correspond to a sine wave where the peak values is now 255.
        255 later corresponding to 100% duty cycle
90      lookUp_A[i] = (int)(temp + 0.5);                      // Round up to
       integer.
91      lookUp_B[i] = (int)(temp + 0.5);
92      lookUp_C[i] = (int)(temp + 0.5);
93      Serial.print(i);
94      Serial.print("  :  ");
95          Serial.println(lookUp_A[i]);
96    }
97
98    //DDRX is a bitmap oriented register that controls the Data Direction
       of each bit on Port B. i.e. 0 for Read and 1 for Write
99
100   DDRH |= (1 << MotorPinA);
101   DDRH |= (1 << MotorSpeedPinA);
102   DDRH |= (1 << MotorBrakePinA);
```

```
103
104     DDRA |= (1 << MotorPinB);
105     DDRB |= (1 << MotorSpeedPinB);
106     DDRA |= (1 << MotorBrakePinB);
107
108     DDRL |= (1 << MotorPinC);
109     DDRB |= (1 << MotorSpeedPinC);
110     DDRL |= (1 << MotorBrakePinC);
111
112     //PORTB is the register the code uses to set the port pins of Port B
          if writing, i.e. 0 for LOW and 1 for HIGH
113
114     //Mega
115     PORTH |= (1 << MotorPinA);
116     PORTH |= (1 << MotorSpeedPinA); //This acts like analogWrite(255) i.e.
          100% duty cycle
117     PORTH |= (0 << MotorBrakePinA);
118
119     PORTA |= (1 << MotorPinB);
120     PORTB |= (1 << MotorSpeedPinB);
121     PORTA |= (0 << MotorBrakePinB);
122
123     PORTL |= (1 << MotorPinC);
124     PORTB |= (1 << MotorSpeedPinC);
125     PORTL |= (0 << MotorBrakePinC);
126
127     //Mega
128     TIFR3 |= (1 << OCF3B);
129     TIFR3 |= (1 << OCF3A);
130     //Enabling Timer Flags for Outut compare A and B. i.e. the type of
          interrupt we will be using.
131
132     TCCR3A = 0;
133
134     // Clearing TCCR3A
135     //Bits in TCCR3A can be set to do a bunch of things, but we don't need
          any of them for this.
136     //
137     TIFR4 |= (1 << OCF4B);
138     TIFR4 |= (1 << OCF4A);
139
140     // Clearing TCCR4A
141     TCCR4A = 0;
142
143     TIFR5 |= (1 << OCF5B);
144     TIFR5 |= (1 << OCF5A);
145
146     // Clearing TCCR5A
147     TCCR5A = 0;
148     //   Reset Timer3 Control Reg B
149     TCCR3B = 0;
150     //  Reset Timer4 Control Reg B
151     TCCR4B = 0;
```

```
152   // Reset Timer5 Control Reg B
153   TCCR5B = 0;
154   //Just making sure this zero before we set any bits
155
156   //set pre-scalar to 256
157   //Arduino clock operates at 16,000,000 Hz. 16 bit timers roll over
       after 65536 clocks (2^16). In order to use frequencies like 1Hz or 50
       Hz a prescaler is applied.
158   //This means that 256 system clocks will pass everytime a single clock
        is counted by any of the timers
159   //Timer3B
160   TCCR3B |= (1 << CS32);
161   TCCR3B |= (0 << CS31);
162   TCCR3B |= (0 << CS30);
163   //    //Timer4B
164   TCCR4B |= (1 << CS42);
165   TCCR4B |= (0 << CS41);
166   TCCR4B |= (0 << CS40);
167   //    //Timer5B
168   TCCR5B |= (1 << CS52);
169   TCCR5B |= (0 << CS51);
170   TCCR5B |= (0 << CS50);
171   //Setting these bits correspond to pre-scaler of 256
172
173   //  //TCNT3 is Timer3
174   TCNT3 = t3_load;
175   OCR3A = t3_compA; //Comp A
176   OCR3B = t3_compB; //Comp B
177   //
178   ////    //TCNT4 is Timer4
179   TCNT4 = t4_load;
180   OCR4A = t4_compA; //Comp A
181   OCR4B = t4_compB; //Comp B
182   ////
183   ////    //TCNT5 is Timer5
184   TCNT5 = t5_load;
185   OCR5A = t5_compA; //Comp A
186   OCR5B = t5_compB; //Comp B
187
188   sei(); //enable global interrupts
189   //  //Timer3
190   TIMSK3 = 0;
191   TIMSK3 |= (1 << OCIE3A);
192   TIMSK3 |= (1 << OCIE3B);
193   ////Specifically allows Timer 3 to intitate interrupts
194   //// //Timer4
195   TIMSK4 = 0;
196   TIMSK4 |= (1 << OCIE4A);
197   TIMSK4 |= (1 << OCIE4B);
198   ////Specifically allows Timer 4 to intitate interrupts
199   ////Timer5
200   TIMSK5 = 0;
201   TIMSK5 |= (1 << OCIE5A);
```

```
202   TIMSK5 |= (1 << OCIE5B);  //Disable or enable sets of timers using
       this timsk stuff
203   //Specifically allows Timer 4 to intitate interrupts
204 }
205
206 void loop() {
207   double currentSenseA = map(analogRead(CurrentSensePinA), 0, 1024, 0,
       5000);
208   double currentSenseB = map(analogRead(CurrentSensePinB), 0, 1024, 0,
       5000);
209   double currentSenseC = map(analogRead(CurrentSensePinC), 0, 1024, 0,
       5000);
210   double CurrentA = currentSenseA * currentFactor;
                        // get channel A current from voltage and current
       factor (from data sheet)
211   double CurrentB = currentSenseB * currentFactor;
                        // get channel A current from voltage and current
       factor (from data sheet)
212   double CurrentC = currentSenseC * currentFactor;
213   // get channel A current from voltage and current factor (from data
       sheet)
214   if (CurrentA > 2000
215       || CurrentB > 2000
216       || CurrentC > 2000
217 //      )
218   X = int(((TCNT3) / DutyCycle));                              //X
       represents the number of DutyCycles that have passed on the current
       Timer 3 clock cycle
219   analogWrite(9, lookUp_A[X]);                                  //
       Assigns the corresponding sine value to analogwrite, changing the
       duty cycle as appropriate
220   Y = int(((TCNT4)/ DutyCycle));
221   analogWrite(13, lookUp_B[Y]);
222   Z = int(((TCNT5) / DutyCycle));
223   analogWrite(12, lookUp_C[Z]);
224 } //end loop
225 //Interrupts - These are called when the interrupt conditions establish
       previously are met
226
227 //Mega
228 ISR(TIMER3_COMPA_vect) {
229   PORTH ^= (1 << MotorPinA); //Current direction changed
230   //  PORTH ^= (1 << MotorSpeedPinA);
231   //  PORTH ^= (1 << MotorBrakePinA);
232 }
233 ISR(TIMER3_COMPB_vect) { //Current direction changed
234   PORTH ^= (1 << MotorPinA);
235 //    PORTH ^= (1 << MotorSpeedPinA);
236 //    PORTH ^= (1 << MotorBrakePinA);
237   TCNT3 = 0; //There are more for timer 4 and 5 but appendix page limit
```

17 APPENDIX C

```
for k = 1:SF_num
   %Variation of Scale Factor, SF denotes array.
    Scale_Factor = Scale(1,k);
    b_SF{k} = b * Scale_Factor;
    Slot_Distance_SF{k} = Slot_Distance * Scale_Factor;
    R_SF{k} = R * Scale_Factor;
    Shell_Thickness_SF{k} = Shell_Thickness * Scale_Factor;
    StatorWidth_SF{k} = 0.06 * Scale_Factor;
    Airgap_SF{k} = Airgap * Scale_Factor;
    StraightWire_Length_SF{k} = StatorWidth_SF{k};
    Coil_Circumference_SF{k} = Coil_Circumference * Scale_Factor;
    Rotor_Diameter_SF{k} = R_SF{k};
    MaximumCoilCrossSection_Area_SF{k} = MaximumCoilCrossSection_Area *
   (Scale_Factor^2);


    for AWG = 1:AWG_num
            N_SF{k}{AWG} = round(((MaximumCoilCrossSection_Area_SF{k}/
   WireCrossSectional_Area{AWG})*MaximumCirclePackingDensity)-0.5); %
   Number of Turns per Coil
    %Number of coil turns that will fit into stator cross sectional area
    for K = 1:Kcoil
%Slots contribute to the magnetic field by virtue of position as well as
%time.

%Current, Voltage and Resistance of Coils
%Current as a function of position and time (Itk)
            Coil_Resistance_SF{k}{AWG} = (N_SF{k}{AWG} *
   Coil_Circumference_SF{k} * rho_c) / WireCrossSectional_Area{AWG}; %
   should probably include Inductance/ Reactance tbh
            XR{k}{AWG} = Coil_Resistance_SF{k}{AWG} +
   DC_Internal_Resistance + Additional_Resistance;
            Coil_Current_SF{k}{AWG}{K} = V_Load / XR{k}{AWG};
            I_RMS_SF{k}{AWG}{K} = Coil_Current_SF{k}{AWG}{K}/(2^0.5);
            Itk_SF{k}{AWG}{K}=(2^0.5)*N_SF{k}{AWG}*I_RMS_SF{k}{AWG}{K}*
   sin((-w * t +ThetaA0)+PhaseA+(((2*K)+1)*((bn*pi)/(P_pair*Tau))));%
   Current in terms of t and k, currently the same everywhere, should be
    changed per phase


    %Approximate indication of Magnetic Field produced by stator
   windings.
    B_Coil{k}{AWG}{K} = (u0 * Itk_SF{k}{AWG}{K} * us) / (2 * pi);

    %Inductance,Reactance and Impedance
    %L{k}{AWG}{K} = (N_SF{k}{AWG} * B_Coil{k}{AWG}{K})/Itk_SF{k}{AWG}{K
   };
    L{k}{AWG}{K} = (N_SF{k}{AWG}^2 * WireCrossSectional_Area{AWG} * u0 *
    us)/StatorWidth_SF{k};
    XL{k}{AWG}{K} = 2 * pi * f * L{k}{AWG}{K};
```

```
39      Z{k}{AWG}{K} = complex(XR{k}{AWG}, XL{k}{AWG}{K});
40      %Preparing variables for iteration. Used to converge on impedance.
41      Coil_Current_SF_t{k}{AWG}{K}{1} = Coil_Current_SF{k}{AWG}{K};
42      I_RMS_SF_t{k}{AWG}{1} =  I_RMS_SF{k}{AWG};
43      Itk_SF_t{k}{AWG}{K}{1} = Itk_SF{k}{AWG}{K};
44      B_Coil_t{k}{AWG}{K}{1} = B_Coil{k}{AWG}{K};
45      L_t{k}{AWG}{K}{1} = L{k}{AWG}{K} ;
46      XL_t{k}{AWG}{K}{1} = XL{k}{AWG}{K};
47      Z_t{k}{AWG}{K}{1} = Z{k}{AWG}{K};
48
49  %         Self - contained iterative process to find impedance
50              for iter = 2:30
51                  Coil_Current_SF_t{k}{AWG}{K}{iter} = V_Load / abs(Z_t{k
    }{AWG}{K}{iter-1});
52                  I_RMS_SF_t{k}{AWG}{K}{iter} = Coil_Current_SF_t{k}{AWG}{
    K}{iter}/(2^0.5);
53                  Itk_SF_t{k}{AWG}{K}{iter}=(2^0.5)*N_SF{k}{AWG}*
    I_RMS_SF_t{k}{AWG}{K}{iter}*sin((-w*t+ThetaA0)+PhaseA+(((2*K)+1)*((bn
    *pi)/(P_pair*Tau))));
54                  B_Coil_t{k}{AWG}{K}{iter} = (u0 * Itk_SF_t{k}{AWG}{K}{
    iter} * us) / (2 * pi);
55                  L_t{k}{AWG}{K}{iter} = (N_SF{k}{AWG} * B_Coil_t{k}{AWG}{
    K}{iter-1})/Itk_SF_t{k}{AWG}{K}{iter};
56                  XL_t{k}{AWG}{K}{iter} = 2 * pi * f * L_t{k}{AWG}{K}{iter
    };
57                  Z_t{k}{AWG}{K}{iter} = complex(XR{k}{AWG}, XL_t{k}{AWG}{
    K}{iter});
58                  Zabs_t{k}{AWG}{K}{iter} = abs(Z_t{k}{AWG}{K}{iter});
59              end
60
61          Coil_Current_SF{k}{AWG}{K} = Coil_Current_SF_t{k}{AWG}{K}{iter};
62          I_RMS_SF{k}{AWG}{K} = I_RMS_SF_t{k}{AWG}{K}{iter};
63          Itk_SF{k}{AWG}{K} = Itk_SF_t{k}{AWG}{K}{iter};
64          B_Coil{k}{AWG}{K} = B_Coil_t{k}{AWG}{K}{iter};
65          L{k}{AWG}{K} = L_t{k}{AWG}{K}{iter};
66          XL{k}{AWG}{K} = XL_t{k}{AWG}{K}{iter};
67          Z{k}{AWG}{K} = Z_t{k}{AWG}{K}{iter};
68          Zabs{k}{AWG}{K} = abs(Z_t{k}{AWG}{K}{iter});
69
70          %There are two current limits. One dictated by the motor
    controller
71          %(4A) and one dictated by the breakdown voltage of the enamelled
72          %wire being used. The value of I_load reflects this.
73          Required_Resistance{k}{AWG}{K} = V_Load / I_Load;
74          Resistor{k}{AWG}{K} = Required_Resistance{k}{AWG}{K} - Zabs{k}{
    AWG}{K};
75          I_load_final{k}{AWG}{K} = V_Load./(Zabs{k}{AWG}{K} + Resistor{k
    }{AWG}{K});
76          Itk_SF{k}{AWG}{K} = I_load_final{k}{AWG}{K}; %current limited to
     4A
77      end
78
79  %All current contribution to flux summed across one stator. Will later
```

```
    be
%needed for summing across all stators
     B_Stator{k}{AWG} = 0;
     Itk_Stator_SF{k}{AWG} = 0;

         for K = 1:Kcoil
             B_Stator{k}{AWG} = B_Stator{k}{AWG} + B_Coil{k}{AWG}{K};
             Itk_Stator_SF{k}{AWG} = Itk_Stator_SF{k}{AWG} + Itk_SF{k}{
    AWG}{K};
         end
     end
     for i = 1 : Material_Number
         Core_Volume_Sphere{k}{i} = (4*pi*(R_SF{k}^3) / 3);
         Shell_Volume_Sphere{k}{i} = (4*pi*((R_SF{k}+Shell_Thickness_SF{k
    })^3) / 3) - (4*pi*(R_SF{k}^3) / 3);
         Core_Mass_Sphere{k}{i} = Core_Volume_Sphere{k}{i} * Density{i};
         Shell_Mass_Sphere{k}{i} = Shell_Volume_Sphere{k}{i} * Density{i
    };
         Core_Inertia_Sphere{k}{i} = (2/5) * Core_Mass_Sphere{k}{i} *
    R_SF{k}^2;
         Shell_Inertia_Sphere{k}{i} = (2/5) * ((Core_Mass_Sphere{k}{i} *
    (R_SF{k} + Shell_Thickness_SF{k})^2) - (Core_Mass_Sphere{k}{i} * R_SF
    {k}^2));
         Core_Volume_Cylinder{k}{i} = pi * (R_SF{k}^2) * 2 * (
    StatorWidth_SF{k}/2) ;
         Shell_Volume_Cylinder{k}{i} = ((pi*(R_SF{k} + Shell_Thickness_SF
    {k})^2) * 2 * (StatorWidth_SF{k}/2)) - ((pi*(R_SF{k})^2) * 2 * (
    StatorWidth_SF{k}/2));
         Core_Mass_Cylinder{k}{i} = Core_Volume_Cylinder{k}{i}*Density{i
    };
         Shell_Mass_Cylinder{k}{i} = Shell_Volume_Cylinder{k}{i}*Density{
    i};
         Core_Inertia_Cylinder{k}{i} = (1/2) * Core_Mass_Cylinder{k}{i} *
     R_SF{k}^2;
         Shell_Inertia_Cylinder{k}{i} = ((1/2) * Core_Mass_Cylinder{k}{i}
     * (R_SF{k}+Shell_Thickness_SF{k})^2) - ((1/2) * Core_Mass_Cylinder{k
    }{i} * R_SF{k}^2);

             for AWG = 1:AWG_num
                 %EMF generated by changing magnetic field
                 E_ind{k}{AWG}{i} = B_Stator{k}{AWG} * cos(2 * Theta_Z) *
     Relative_V * 2 * Stator_Width;
                 %Eddy currents assumed to flow along the contours of the
     electric
                 %field flux. No laminations in rotor so there would be
    significant
                 %eddy current losses, but that is assumed not to be the
    case here.
                 dI_ind{k}{AWG}{i} = E_ind{k}{AWG}{i} / ((N_SF{k}{AWG} *
    Coil_Circumference_SF{k} * Resistivity{i}) / (WireCrossSectional_Area
    {AWG}));


```

```
112             %differential form of force acting over an element of
    the rotor
113             Lorentz_Forces{k}{AWG}{i} = B_Stator{k}{AWG} * dl_ind{k
    }{AWG}{i} * Stator_Width;
114
115             Flux_Normalized_Function_Radial_Core{k} = @(
    theta_variable) (log( b_SF{k} ) - log( b_SF{k}.^2 - 2.*b_SF{k}.*(R_SF
    {k} - Shell_Thickness_SF{k}).*cos(theta_variable) + (R_SF{k} -
    Shell_Thickness_SF{k})^2)) .* cos(1 .* theta_variable) ;
116             Flux_Normalized_Function_Tangential_Core{k} = @(
    theta_variable) (log( b_SF{k} ) - log( b_SF{k}.^2 - 2.*b_SF{k}.*(R_SF
    {k} - Shell_Thickness_SF{k}).*cos(theta_variable) + (R_SF{k} -
    Shell_Thickness_SF{k})^2)) .* sin(1 .* theta_variable) ;
117             Flux_Normalized_Function_Radial_Shell{k} = @(
    theta_variable) (log( b_SF{k} ) - log( b_SF{k}.^2 - 2.*b_SF{k}.*R_SF{
    k}.*cos(theta_variable) + R_SF{k}^2)) .* cos(1 .* theta_variable);
118             Flux_Normalized_Function_Tangential_Shell{k} = @(
    theta_variable) (log( b_SF{k} ) - log( b_SF{k}.^2 - 2.*b_SF{k}.*R_SF{
    k}.*cos(theta_variable) + R_SF{k}^2)) .* sin(1 .* theta_variable);
119             %Simple numerical integration
120             Flux_Integral_Radial_Core{k}          =integral(
    Flux_Normalized_Function_Radial_Core{k},-thetalimit_core,
    thetalimit_core);
121             Flux_Integral_Tangential_Core{k}      =integral(
    Flux_Normalized_Function_Tangential_Core{k},-thetalimit_core,
    thetalimit_core);
122             Flux_Integral_Radial_Shell{k}         =integral(
    Flux_Normalized_Function_Radial_Shell{k},-thetalimit_shell,
    thetalimit_shell) - Flux_Integral_Radial_Core{k};
123             Flux_Integral_Tangential_Shell{k}     =integral(
    Flux_Normalized_Function_Tangential_Shell{k},-thetalimit_shell,
    thetalimit_shell) - Flux_Integral_Tangential_Core{k};
124             %Total force acting on Shell and Core of Rotor.
    Unsurprisingly
125             %Tangential contribution is negligible.
126             Force_Shell_TANG{k}{AWG}{i} = Lorentz_Forces{k}{AWG}{i}
    * Flux_Integral_Radial_Shell{k};
127             Force_Shell_RADI{k}{AWG}{i} = Lorentz_Forces{k}{AWG}{i}
    * Flux_Integral_Tangential_Shell{k};
128             Force_Core_TANG{k}{AWG}{i} = Lorentz_Forces{k}{AWG}{i} *
     Flux_Integral_Radial_Core{k};
129             Force_Core_RADI{k}{AWG}{i} = Lorentz_Forces{k}{AWG}{i} *
     Flux_Integral_Tangential_Core{k};
130
131             %Contribution from both tangential and radial is summed.
132             Total_Force_Shell_RADI{k}{AWG}{i} = Force_Shell_RADI{k}{
    AWG}{i}*2*StraightWire_Length_SF{k};
133             Total_Force_Shell_TANG{k}{AWG}{i} = Force_Shell_TANG{k}{
    AWG}{i}*2*StraightWire_Length_SF{k};
134             Total_Force_Core_RADI{k}{AWG}{i} = Force_Core_RADI{k}{
    AWG}{i}*2*StraightWire_Length_SF{k};
135             Total_Force_Core_TANG{k}{AWG}{i} = Force_Core_TANG{k}{
    AWG}{i}*2*StraightWire_Length_SF{k};
```

```
136                 %Performance Characteristics are calculated
137                 Torque_Shell{k}{AWG}{i} = R_SF{k} *
    Total_Force_Shell_TANG{k}{AWG}{i};
138                 Torque_Core{k}{AWG}{i} = (R_SF{k} - Shell_Thickness_SF{k
    }) * Total_Force_Core_TANG{k}{AWG}{i};
139                 %Forces is multiplied by number of stators or total
    stator
140                 Force_Total{k}{AWG}{i} = ((2*pi)/Stator_Pitch) * (
    Force_Core_TANG{k}{AWG}{i} + Force_Shell_TANG{k}{AWG}{i});
141                 Torque_Total{k}{AWG}{i} =((2*pi)/Stator_Pitch) * (
    Torque_Shell{k}{AWG}{i} + Torque_Core{k}{AWG}{i});
142                 Inertia_Total_Sphere{k}{i} = Core_Inertia_Sphere{k}{i} +
     Shell_Inertia_Sphere{k}{i};
143                 Acceleration_Sphere{k}{AWG}{i} = Torque_Total{k}{AWG}{i
    }/Inertia_Total_Sphere{k}{i};
144             end
145         end
146
147 end
```

17.1 Fritzing Diagram of 3 Phase Motor Controller

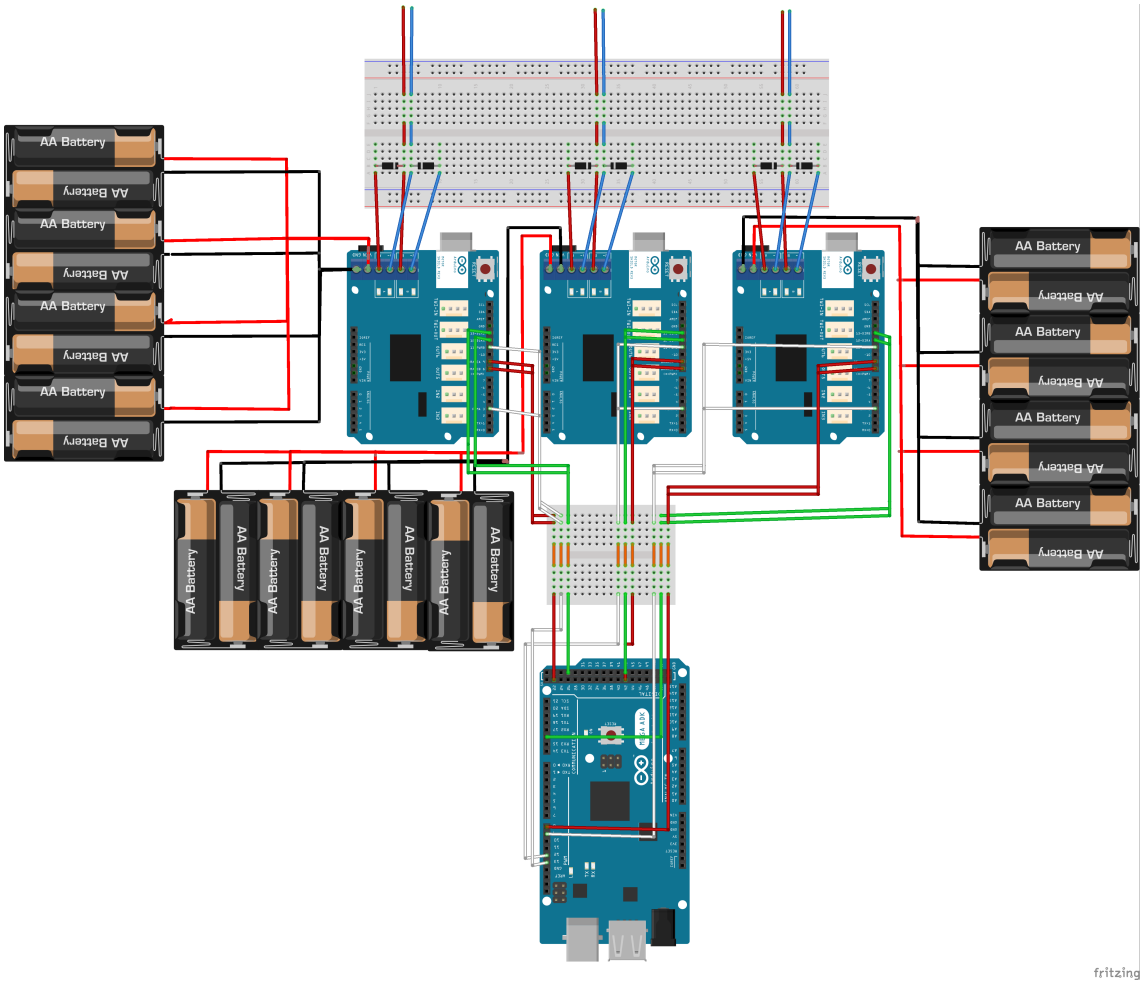
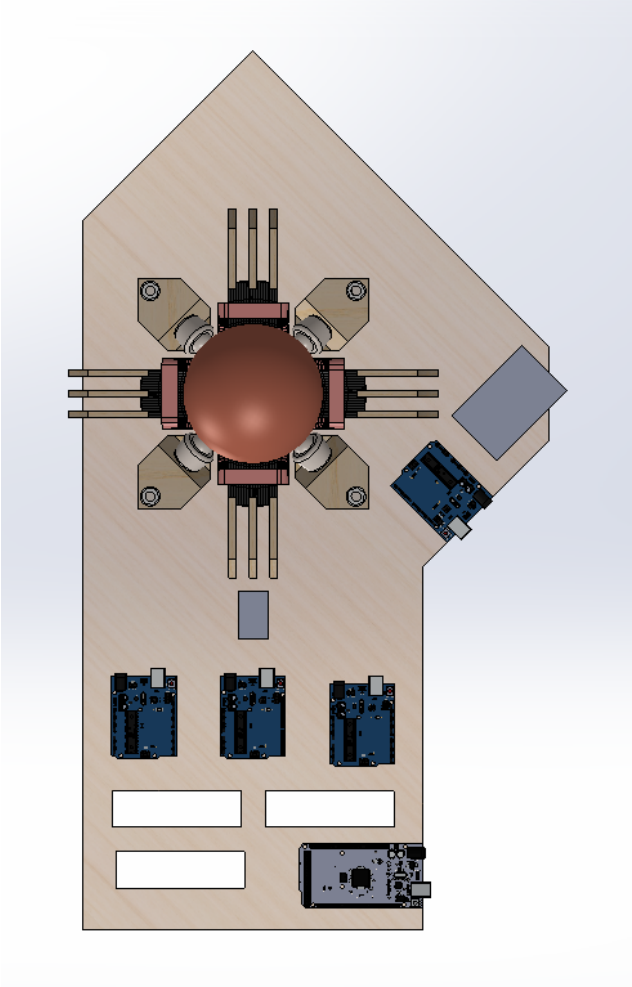
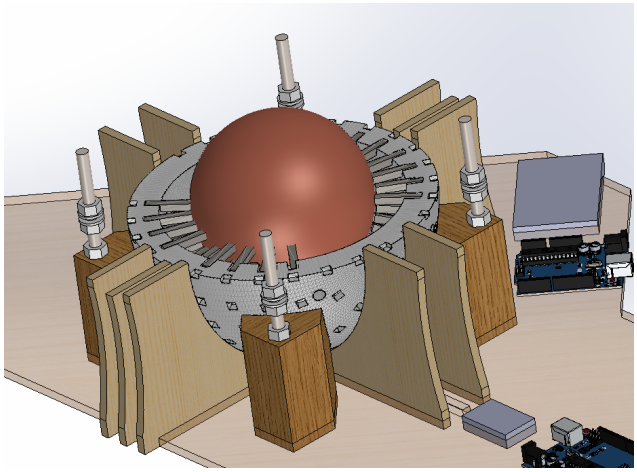


Figure 17.1 – Three-Phase Arduino Based Motor Controller

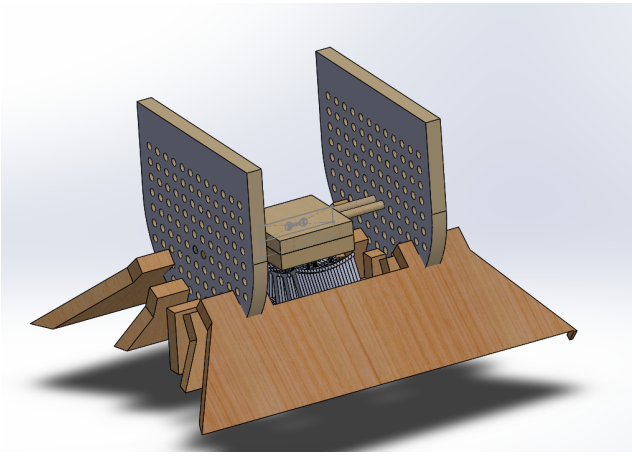
18 APPENDIX D



(a) Full Induction CAD that was ultimately produced.



(b) Alternative PMSM design



(c) Flux Measurement

References

[1] J. C. Dixon, Suspension geometry and computation. John Wiley & Sons, 2009.

[2] W. F. Milliken, D. L. Milliken et al., Race car vehicle dynamics. Society of Automotive Engineers Warrendale, PA, 1995, vol. 400.

[3] M. Beato, V. Ciaravola, M. Russo, and A. Volpe, "Lateral tyre force by a milliken test on a flat track roadway simulator," Vehicle System Dynamics, vol. 34, no. 2, pp. 117–129, 2000.

[4] G. G. Gómez, M. Nybacka, E. Bakker, and L. Drugge, "Objective metrics for vehicle handling and steering and their correlations with subjective assessments," International Journal of Automotive Technology, vol. 17, no. 5, pp. 777–794, 2016.

[5] J.-n. Wang, Q.-n. Wang, L.-q. Jin, and C.-x. Song, "Independent wheel torque control of 4wd electric vehicle for differential drive assisted steering," Mechatronics, vol. 21, no. 1, pp. 63–76, 2011.

[6] D. Schrank, B. Eisele, and T. Lomax, "Tti's 2012 urban mobility report," Texas A&M Transportation Institute. The Texas A&M University System, vol. 4, 2012.

[7] E. T. Verhoef and J. Rouwendal, "A behavioural model of traffic congestion: endogenizing speed choice, traffic safety and time losses," Journal of Urban Economics, vol. 56, no. 3, pp. 408–434, 2004.

[8] B. Li, H. Du, W. Li, and Y. Zhang, "Side-slip angle estimation based lateral dynamics control for omni-directional vehicles with optimal steering angle and traction/brake torque distribution," Mechatronics, vol. 30, pp. 348–362, 2015.

[9] A. M. T. West and H. Asada, "Design of a holonomic omnidirectional vehicle," Master's thesis, Massachusetts Institute of Technology, Dept. of Mechanical Engineering, 1992.

[10] A. Macfarlane and T. van Niekerk, "Design tool for development and performance analysis of omni-directional agvs," in 2017 Pattern Recognition Association of South Africa and Robotics and Mechatronics (PRASA-RobMech). IEEE, pp. 50–55.

[11] G. Szayer, B. Kovács, F. Tajti, and P. Korondi, "Feasible utilization of the inherent characteristics of holonomic mobile robots," Robotics and Autonomous Systems, vol. 94, pp. 12–24, 2017.

[12] K. Tadakuma, R. Tadakuma, and J. Berengeres, "Development of holonomic omnidirectional vehicle with "omni-ball": Spherical wheels," in 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2007, pp. 33–39.

[13] R. Holmberg and O. Khatib, "Development and control of a holonomic mobile robot for mobile manipulation tasks," The International Journal of Robotics Research, vol. 19, no. 11, pp. 1066–1074, 2000.

[14] S. Zhang and W. Huang, "Application of a propeller-based air propulsion system to the land-based holonomic vehicle," Applied Sciences, vol. 9, no. 21, p. 4657, 2019.

[15] K. L. Moore, M. Davidson, V. Bahl, S. Rich, and S. Jirgal, "Modelling and control of a six-wheeled autonomous robot," in Proceedings of the 2000 American Control Conference. ACC (IEEE Cat. No. 00CH36334), vol. 3. IEEE, 2000, pp. 1483–1490.

[16] B. Masters and P. Thiel, Zero to one: notes on start ups, or how to build the future. Random House, 2014.

[17] A. H. Hon, M. Bloom, and J. M. Crant, "Overcoming resistance to change and enhancing creative performance," Journal of Management, vol. 40, no. 3, pp. 919–941, 2014.

[18] Meriam-Webster, "Meriam-webster online dictionary; meaning of open-source in english," https://www.merriam-webster.com/dictionary/open-source, 2020.

[19] S. Davidson, "Open-source hardware," IEEE Design & Test of Computers, vol. 21, no. 5, pp. 456–456, 2004.

[20] S. et al., "About hackaday," Hackaday.com.

[21] F. Daniel K and G. Peter J, "Open-source hardware is a low-cost alternative for scientific instrumentation and research," Modern instrumentation, vol. 2012, 2012.

[22] G. Cawood, "Summary of omni-directional drive choices," Mechantronics, 2012.

[23] V. Alakshendra and S. S. Chiddarwar, "Adaptive robust control of mecanum-wheeled mobile robot with uncertainties," Nonlinear Dynamics, vol. 87, no. 4, pp. 2147–2169, 2017.

[24] J.-B. Song and K.-S. Byun, "Steering control algorithm for efficient drive of a mobile robot with steerable omni-directional wheels," Journal of mechanical science and technology, vol. 23, no. 10, p. 2747, 2009.

[25] B. Lu and Y. Xu, "Development and analysis of a novel spherical 2-degree-of-freedom (2-dof) hybrid stepping motor," Energies, vol. 11, no. 1, p. 41, 2018.

[26] L. Hertig, D. Schindler, M. Bloesch, C. Remy, and R. Siegwart, "Unified state estimation for a ballbot," 05 2013, pp. 2471–2476.

[27] D. Howe, "Magnetic actuators," Sensors and Actuators A: Physical, vol. 81, no. 1-3, pp. 268–274, 2000.

[28] Y. Öner, "A permanent magnet spherical rotor design and three dimensional static magnetic analysis," Sensors and Actuators A: Physical, vol. 137, no. 2, pp. 200–208, 2007.

[29] W. Chen, L. Zhang, L. Yan, and J. Liu, "Design and control of a three degree-of-freedom permanent magnet spherical actuator," Sensors and Actuators A: Physical, vol. 180, pp. 75–86, 2012.

[30] F. Williams, E. Laithwaite, and J. Eastham, "Development and design of spherical induction motors," Proceedings of the IEE-Part A: Power Engineering, vol. 106, no. 30, pp. 471–484, 1959.

[31] S. Dufresne, "https://hackaday.com/2016/06/24/driving-bb-8-more-than-one-way-to-move-this-bot/," 2016.

[32] E. Leroy, J. Lozada, and M. Hafez, "A curved ultrasonic actuator optimized for spherical motors: Design and experiments," Ultrasonics, vol. 54, no. 6, pp. 1610–1619, 2014.

[33] M. Hoshina, T. Mashimo, N. Fukaya, O. Matsubara, and S. Toyama, "Spherical ultrasonic motor drive system for camera orientation in pipe inspection," Advanced Robotics, vol. 27, no. 3, pp. 199–209, 2013.

[34] G. Rogers, "Three degree-of-freedom piezoelectric ultrasonic micro-motor with a major diameter of 350 μm," Journal of Micromechanics and Microengineering, vol. 20, no. 12, p. 125002, 2010.

[35] X. Zhang, G. Zhang, K. Nakamura, and S. Ueha, "A robot finger joint driven by hybrid multi-dof piezoelectric ultrasonic motor," Sensors and actuators A: Physical, vol. 169, no. 1, pp. 206–210, 2011.

[36] B. Lu, M. Aoyagi, T. Takano, and H. Tamura, "Examination of sandwich-type multidegree-of-freedom spherical ultrasonic motor," Japanese Journal of Applied Physics, vol. 49, no. 7S, p. 07HE24, 2010.

[37] R. T. Tjeung, M. S. Hughes, L. Y. Yeo, and J. R. Friend, "Surface acoustic wave micromotor with arbitrary axis rotational capability," Applied Physics Letters, vol. 99, no. 21, p. 214101, 2011.

[38] J. Luo, C. Huang, H. Li, S. Xie, and Y. M. Zhang, "Structural design and analysis of 3-dof bionic eye based on spherical ultrasonic motor," in International Conference on Intelligent Robotics and Applications. Springer, 2012, pp. 348–356.

[39] S.-C. Shen and J.-C. Huang, "Design and fabrication of a high-power eyeball-like microactuator using a symmetric piezoelectric pusher element," Journal of microelectromechanical systems, vol. 19, no. 6, pp. 1470–1476, 2010.

[40] K. Otokawa, K. Takemura, and T. Maeno, "A multi-degree-of-freedom ultrasonic motor using single-phase-driven vibrators," in 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2005, pp. 499–504.

[41] T. Endo and Y. Nakamura, "An omnidirectional vehicle on a basketball," in ICAR'05. Proceedings., 12th International Conference on Advanced Robotics, 2005. IEEE, 2005, pp. 573–578.

[42] P. Li and Ratner, "Sds electric motorcycle," SphericalDriveSystem.com, 2016.

[43] T. B. Lauwers, G. A. Kantor, and R. L. Hollis, "A dynamically stable single-wheeled mobile robot with inverse mouse-ball drive," in Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006. IEEE, 2006, pp. 2884–2889.

[44] U. Nagarajan, G. Kantor, and R. Hollis, "The ballbot: An omnidirectional balancing mobile robot," The International Journal of Robotics Research, vol. 33, no. 6, pp. 917–930, 2014.

[45] C. Mack, "The multiple lives of moore's law," IEEE Spectrum, vol. 52, no. 4, pp. 31–31, 2015.

[46] A. Hughes and B. Drury, Electric motors and drives: fundamentals, types and applications. Newnes, 2019.

[47] K.-M. Lee, G. Vachtsevanos, and C. Kwan, "Development of a spherical stepper wrist motor," Journal of Intelligent and Robotic Systems, vol. 1, no. 3, pp. 225–242, 1988.

[48] K.-M. Lee and C.-K. Kwan, "Design concept development of a spherical stepper for robotic applications," IEEE Transactions on Robotics and Automation, vol. 7, no. 1, pp. 175–181, 1991.

[49] G. S. Chirikjian and D. Stein, "Kinematic design and commutation of a spherical stepper motor," IEEE/ASME Transactions on mechatronics, vol. 4, no. 4, pp. 342–353, 1999.

[50] Q. Wang, Z. Li, Y. Ni, and W. Jiang, "Magnetic field computation of a pm spherical stepper motor using integral equation method," IEEE Transactions on magnetics, vol. 42, no. 4, pp. 731–734, 2006.

[51] E. Laithwaite, "Linear induction motors," Proceedings of the IEE-Part A: Power Engineering, vol. 104, no. 18, pp. 461–470, 1957.

[52] J. Eastham and E. Laithwaite, "Linear induction motors as 'electromagnetic rivers'," in Proceedings of the Institution of Electrical Engineers, vol. 121, no. 10. IET, 1974, pp. 1099–1108.

[53] E. R. Laithwaite, A history of linear electric motors. Macmillan International Higher Education, 1987.

[54] E. Laithwaite, "Adapting a linear induction motor for the acceleration of large masses to high velocities," IEE Proceedings-Electric Power Applications, vol. 142, no. 4, pp. 262–268, 1995.

[55] E. Musk, "Hyperloop alpha," SpaceX: Hawthorne, CA, USA, 2013.

[56] E. Laithwaite, J. Eastham, H. Bolton, and T. Fellows, "Linear motors with transverse flux," in Proceedings of the Institution of Electrical Engineers, vol. 118, no. 12. IET, 1971, pp. 1761–1767.

[57] K. Higashi, S. Ohashi, H. Ohsaki, and E. Masada, "Magnetic damping of the electrodynamic suspension-type superconducting levitation system," Electrical Engineering in Japan, vol. 127, no. 2, pp. 49–60, 1999.

[58] K. Davey, G. Vachtsevanos, and R. Powers, "The analysis of fields and torques in spherical induction motors," IEEE Transactions on Magnetics, vol. 23, no. 1, pp. 273–282, 1987.

[59] J. F. Fernandes and P. C. Branco, "The shell-like spherical induction motor for low-speed traction: electromagnetic design, analysis, and experimental tests," IEEE Transactions on Industrial Electronics, vol. 63, no. 7, pp. 4325–4335, 2016.

[60] S.-M. Lu, "A review of high-efficiency motors: Specification, policy, and technology," Renewable and Sustainable Energy Reviews, vol. 59, pp. 1–12, 2016.

[61] C. Xia, H. Li, and T. Shi, "3-d magnetic field and torque analysis of a novel halbach array permanent-magnet spherical motor," IEEE Transactions on Magnetics, vol. 44, no. 8, pp. 2016–2020, 2008.

[62] A. Moseman, "Goodyear made spherical maglev tires that are totally nuts," Popular Mechanics, 2016.

[63] K. Neer, "How product placement works," HowStuffWork. com (2013), 2003.

[64] D. Fleisch, A student's guide to Maxwell's equations. Cambridge University Press, 2008.

[65] J. Ma, Y. Zhu, and J. Jiang, "A novel method to calculate the thrust of linear induction motor based on instantaneous current value," Journal of Modern Transportation, vol. 20, no. 2, pp. 76–81, 2012.

[66] I. Boldea, Linear electric machines, drives, and MAGLEVs handbook. CRC press, 2017.

[67] J. F. Fernandes, S. M. Vieira, and P. C. Branco, "Multiobjective optimization of a shell-like induction spherical motor for a power-assisted wheelchair," IEEE Transactions on Energy Conversion, vol. 33, no. 2, pp. 660–669, 2017.

[68] Instructables, "Make your own miniature electric hub motor."

[69] Paradisetronic, "Mpu 9250 data sheet."