Magnebots: Actively Controlled Magnetic Robots

by

Steven L. Tamm

Submitted to the Department of Electrical Engineering and Computer Science
in Partial Fulfillment of the Requirements for the Degree of
Master of Engineering in Electrical Engineering and Computer Science
at the Massachusetts Institute of Technology

July 30, 1997

Copyright 1997 Steven L. Tamm. All Rights Reserved.

The author hereby grants to M.I.T. permission to reproduce and
distribute publicly paper and electronic copies of this thesis
and to grant others the right to do so.

Author

Department of Electrical Engineering and Computer Science

Certified by

Assistant Prof. of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by

Chairman, Department Committee on Graduate Theses
Magnebots: Actively Controlled Magnetic Robots
by
Steven L. Tamm

Submitted to the Department of Electrical Engineering and Computer Science
in Partial Fulfillment of the Requirements for the Degree of
Master of Engineering in Electrical Engineering and Computer Science
at the Massachusetts Institute of Technology
July 30, 1997

ABSTRACT

Magnebots are independently controlled robots that use electromagnetic repulsion and attraction for locomotion. Using large sinusoidal external magnetic fields, these robots power self-contained electromagnetic coils to move in two dimensions. The external field is provided by four coils of magnetic wire powered by an amplifier. The robots have remote control based on infrared transmissions similar to ones used to control major appliances. The untethered robots contain the magnetic field detectors and current amplifiers. A built-prototype performed according to theory, and by using more custom elements, the efficiency and size of the robots can be improved.

Thesis Supervisor: Gill A. Pratt
Title: Assistant Professor, Department of Electrical Engineering and Computer Science
For my parents,

Who taught me an education
was a terrible thing to waste
Table of Contents

1. INTRODUCTION .................................................................................................................... 4

2. THEORY ........................................................................................................................................ 5
   2.1 ELECTROMAGNETIC INDUCTION ............................................................................. 5
   2.2 PASSIVE SYSTEM........................................................................................................... 8
   2.3 ACTIVE ROBOT ............................................................................................................ 11

3. CIRCUITS ..................................................................................................................................... 13
   3.1 FUNCTION GENERATOR ........................................................................................... 13
   3.2 REMOTE CONTROL CIRCUITRY ........................................................................... 16
   3.3 MAGNETIC FIELD PROPULSION ............................................................................. 21

4. CONCLUSIONS... ERROR! BOOKMARK NOT DEFINED.......................................................... 25
   4.1 POSSIBLE ENHANCEMENTS .................................................................................... 25
   4.1.1 Function Generator ............................................................................................. 25
   4.1.2 Radio Control ..................................................................................................... 25
   4.1.3 Magnetic Field Propulsion ................................................................................... 27
   4.2 OVERALL RESULTS ERROR! BOOKMARK NOT DEFINED........................................ 29

5. APPENDICES.............................................................................................................................. 29
   5.1 APPENDIX A - REMOTE CONTROL ....................................................................... 29
   5.1.1 Analysis................................................................................................................ 29
   5.1.2 RC-5 Code ............................................................................................................ 30
   5.2 APPENDIX B - MEASUREMENTS ............................................................................. 31
   5.2.1 Induction Coil ....................................................................................................... 31
   5.2.2 Magnetic Field ...................................................................................................... 32
   5.3 APPENDIX C - LOGIC CODE .................................................................................... 34

6. ACKNOWLEDGEMENTS ............................................................................................................. 42

7. BIBLIOGRAPHY ......................................................................................................................... 43

Table of Figures
Figure 1 - Layout of induction coils ............................................................................................... 7
Figure 2 - Theoretical Magnetic Field Strength ............................................................................ 8
Figure 3 - Passive System Vector Diagram .................................................................................. 9
Figure 4 - Torque on two coil Passive System ............................................................................ 11
Figure 5 - Measurement of solenoids. ....................................................................................... 12
Figure 6 - Circuit Diagram for Function Generator ..................................................................... 14
Figure 7 - Picture of Dual Function Generator Board ................................................................ 16
Figure 8 - Picture of Remote Control detector board ................................................................. 18
Figure 9 - Circuit Diagram for Remote Control Decoder ............................................................. 19
Figure 10 - FSM for RC Decoding .............................................................................................. 21
Figure 11 - Circuit Diagram for Magnetic Field Detector and Power Circuitry ......................... 23
Figure 12 - Picture of final Magnetic Field Detector and Power Circuitry .................................... 24
Figure 13 - Infrared Transmission ............................................................................................... 29
Figure 14 - Bit Encodings for RC-5 ........................................................................................... 30
Figure 15 - RC5 encoding for Volume + .................................................................................... 31
1. Introduction

Electromagnetism has been harnessed for locomotion since the beginning of its study. Magnetism has the property of causing force between two free objects without any friction, traction, or linkage. From electric motors to the new Superman roller coaster that uses mag-lev tracks to accelerate thrill-seekers to 100mph, the ability to cause motion without friction is exceptionally beneficial. A whole class of devices is built using magnets and electromagnetic induction for the sole purpose of reducing friction. Hall-Effect and proximity switches allow detection of key presses and impedances without contact. Transformers can step up and step down voltage for transmission across long distances with little energy loss. Voice coils create simple linear motors with high energy efficiency, driving speakers that last for 50 years. It is the lack of friction that contributes to their longevity and efficacy.

This quality is imperative in places where friction could cause harm. Advances in CAT and MRI allow doctors to peer inside the body without incisions, by exploiting magnetism. When a better picture is required, however, the doctors have to invade the body. Current techniques usually involve snaking wires and tubes through the pathways of the body. Anyone having to endure a catheter can tell you of his or her discomfort. Large amounts of time and effort are needed to insert and remove these devices from the body. The use of magnetic motion could greatly reduce the discomfort of patients, and allow new forms of invasive surgery never before imagined. Self contained actively controlled magnetic robots, or magnebots, will apply magnetic locomotion to these surgical procedures.

The eventual goal of the Magnebot project is to produce micro-technology surgeons. Imagine small spherical robots, weighing about 300g with a volume of about 10 cm\(^3\). Small surgical tools, like scalpels, clamps, and sutures are attached. A patient is placed in an operating room with huge electromagnets on the walls, floor and ceiling. A small incision is made in the leg of the patient. A few of these robots are inserted into a vein. The doctor then leaves the room and performs the surgery from the next room. Unlike X-rays, the surgeon can stay in the room if he wants. Some of the robots contain cameras others provide luminescence. Each of the robots are independently controlled using radio waves sent by a computer. The surgeon holds a joystick telling the computer which way to move the robots. After flowing through the heart, the robots head toward an aneurysm in the brain stem. By removing the blockage, the patient, untreatable by current surgery, could survive for years.

The coils surrounding the patient resonate with huge amounts of current. This current generates a large magnetic field inside the room, around 1T. The robots have coils of wire that they electrify to counteract the magnetic field. By opposing the magnetic field, a force is generated allowing them to move. The
force placed on the small robot is limited by the current it can generate. A downforce of around .5N is
casted by gravity. The robot would have to generate .5A-m, but the buoyancy of the blood should reduce
the force needed. This vision is currently beyond our capabilities of battery storage, let alone control
systems, but not that far off. There are intermediate steps that can prove the technology and still be
medically beneficial.

A more realistic goal is to replace some of the large bulky vision systems used for diagnostic treatment,
such as colonoscopy. Currently, a small camera and lamp is placed at the end of a long cable. This cable
is inserted into the rectum and directed through the intestine using the cable. The cable is about as wide
as your finger; needless to say, it is rather uncomfortable. In addition, the cable can get kinked or cause
tissue damage inside the lumen. A magnebot would contain a small CCD camera and a lamp, or maybe
use two magnebots. The doctor would direct the camera and lamp using a Nintendo 64 style joystick.
The small robots, encased in epoxy and as big as your thumb, would greatly simplify colonoscopy. The
robot, being larger than the previous vision but smaller than the current techniques, would prevent the
damage currently done. Most importantly, it would ameliorate the discomfort.

To get to this kind of robot, the technology must be proved. The goal of my project was to build a
prototype robot. Its simple mission is to move with a moderate force inside a constant magnetic field.
Two large coils placed on either side of a glass cube provide the magnetic field. It is filled with water, so
we more aptly call it a fishtank. The robot will be controlled remotely, and move in two dimensions. The
circuitry floats on the water using a piece of foam. The magnetic field operates without regard to the
motion of the robot, so that many robots can be placed in the same surface. Ideally, multiple prototypes
should be built, and they should all operate independently.

Before we can get to this active powered stage, a passive system was built. In this system, the magnetic
field provided by the external coils, also known as induction coils, is varied with the direction desired.
The “dumb” robot, consisting solely of unpowered coils of wire, will move in response to a changing
magnetic field. Jianjuen “John” Hu and Prof. Gill Pratt did this work. The next section explains the
theory behind electromagnetic propulsion and the robots operation.

2. Theory

2.1 Electromagnetic Induction
The initial overriding concern is the ability to move in two dimensions. For simplicity's sake, the magnetic fields generated in each direction should be relatively independent. With a constant magnetic field, the distortion along the diagonals would impede linear motion. Our robot would have to detect the angle of the field direction and modulate power to its propulsion coils. The simplest way to remedy this problem is to use a changing magnetic field. It turns out that generating sinusoidal magnetic fields has pleasant properties. By running each of the dimensions at differing (and hopefully relatively prime) frequencies, the dimensions are decoupled. The direction of the field in the corners would still vary, but our robot would run at the frequency of just one of the directions. The average direction over time would be axial. The first step is to build two function generators with independent frequency control. This was my first task.

To generate the magnetic field, we used large coils powered by large amount of current. A function generator produces a sine wave that is placed into a current amplifier. This powers a large 17cm$^2$ square, ~100 turn coil of magnetic wire. This coil acts like both a resistor and an inductor. To tune out the reactance, a large capacitor is placed in series with the wire. When driven at the LRC circuit's breakpoint frequency, the coil should appear to the current amplifier as just a resistor. The measured impedance of the LRC circuit at 1kHz is about 10 ohms. Driven at 60V, it causes 6A to flow through the coil. The magnetic field generated is proportional to the amount of current driven through the coil.

Analysis of the resulting magnetic vector field requires a computer. Without simplifying assumptions, finite element analysis is required. My analysis of the magnetic field begins at basic electromagnetic theory. We assume that the coil is not a square coil, but a circular loop. Its radius would be around 10cm, but it is referred to as $a$. The construction of the fishbowl places one coil on each side of the cube, with the coil size approximating the face of the bowl. Since we are in a cube, the width of the fishbowl would then be one diameter of the coil, or two radii. To find the magnetic field as it varies in space, I'll start with the law of Biot and Savart.

$$dB = \frac{\mu_0 I}{4\pi} \frac{dA \times \vec{E}}{r^2}$$

To simplify matters, we will look only along the $x$-axis, which extends the width of the cube through the center of each coil.
As we travel along the x axis, as seen in Figure 1, our distance from the coil increases. The distance from the axis to the current is the root of the squared radius and the squared distance along x. The coil is symmetrical around the x-axis, so the y component of the field is 0. The magnetic field along the x-axis is

\[ dB_x = \frac{\mu_0 I}{4\pi} \frac{dl}{r} a. \]

To solve for the field, we integrate both sides and substitute for the radius.

\[ B_x = \frac{\mu_0 I}{4\pi} \frac{a}{r^3} \int dl = \frac{\mu_0 I}{4\pi} \frac{a}{r^3} 2\pi a = \frac{\mu_0 I}{2} \frac{a^2}{r^3} = \frac{\mu_0 I}{2} \frac{a^2}{(a^2 + x^2)^{3/2}}. \]

The effect is proportional to by the number of turns, simplifying to the final equation

\[ B_x = \frac{\mu_0 N I a^2}{2(a^2 + x^2)^{3/2}}. \]

This equation ignores the magnetic field off the center axis, but if the coils are far enough away in our final system, this is a valid simplification.

Since we are driving two induction coils, the effect of the second coil on the field should be examined. Figure 2 contains equations and graphs showing this effect. The first curve is for just one coil. The second is for two coils powered in the same direction; this works like Helmholtz coils, causing a relatively flat magnetic field curve inside the fishbowl. The third curve is for two coils powered in the opposite direction; this has the effect of causing the field gradient to be nearly constant.
Magnetic Field for Different Coil Configurations

\[
\begin{align*}
\mathbf{x} & := 0, \ldots, 2.0 \\
\mathbf{y} & := 0, \ldots, 3.5 \\
\mathbf{x0} & := 0, \ldots, 2.0 \\
\mathbf{y10} & := 0, \ldots, 7.0 \\
a(i, t) & := \frac{\sin(t)}{3} \left( \frac{1 + i^2}{\left(1 + i^2\right)^2} \right) \\
b(i, t) & := \frac{\sin(t)}{3} + \frac{\sin(t)}{3} \left( \frac{1 + (2 - i)^2}{\left[1 + (2 - i)^2\right]^2} \right) \\
c(i, t) & := \frac{\sin(t)}{3} \left( 1 + i^2 \right) \left( \frac{1 + (2 - i)^2}{\left[1 + (2 - i)^2\right]^2} \right)
\end{align*}
\]

Figure 2 - Theoretical Magnetic Field Strength

Our passive system uses a single coil, but our active system can use either system of dual powering coils.

### 2.2 Passive System

The passive system works by having one of the induction coils powered at a time. The propulsion coils are passive; no external power is provided. To get this to work, the magnetic field must change. This is due to the phenomenon described by Faraday's Law of electromagnetic induction. The induced EMF in a circuit is the negative of the time rate of change of magnetic flux through the circuit. A constant magnetic field would provide no change in flux. With a sinusoidal input, the change in the flux and the voltage induced will also be a sinusoid, but differing by 90°.

For simplicity, instead of connecting any resistors or capacitors to the propulsion coil, we closed the circuit. The model used for this circuit is a voltage source connected to an inductor in series with a resistor. The resistance of the propulsion coils is about .4Ω and its inductance about .3mH.
Its resistive component can be ignored. By Lenz's Law, the current through the wire would oppose the change in the magnetic field. If the flux is a sine wave, the voltage is a cosine wave; 90° out of phase with the magnetic flux and magnetic field. If it is a resistor, the current going through the coil is proportional to the voltage; 90° degrees out of phase with the magnetic field. The opposing magnetic fields will be 90° out of phase; no net effect of propulsion will be seen. If it is an inductor, however, the current going through the coil will be 90° degrees out of phase with the voltage, and 180° out of phase with the magnetic field. It will oppose the magnetic field all the time. The propulsion coil with then always travel away from the powered coil. Figure 3 shows the interactions between the induced magnetic field and the current flow.

Figure 3 - Passive System Vector Diagram

This effect was seen, with one minor complication; torquing. Two loops of wire with current flowing in the same direction are attracted; in opposite directions repulsed. The propulsion coil moves away from the induction coil. It does this only if it is perfectly parallel to the induction coil. If it is off, by a certain degree, it torques in the direction of the perturbation to become perpendicular. This effect is exploited in AC motors, but is disastrous in our application.
How does this happen? Let’s say the magnetic field \( B_p \) is sine wave with amplitude \( B \). The current is flowing in the propulsion coil to oppose it. The torque on the propulsion coil, or on any current loop in an external magnetic field, is \( \tau = NIA \times B_p = \mu \times B_p \). The magnetic moment \( \mu \) points in the direction of the current using the right hand rule. Since the current always flows to oppose the magnetic field, the \( \mu \) and \( B \) vectors, in the parallel coils, point in the opposite x direction. At 180° the cross product and the torque is zero. With any angle in between, the cross product and the torque would exist.

Here’s another thought experiment. Assume the magnetic field was constant, and the induced current was a constant also. The torque on the coil would turn it 179° (with an initial 1° perturbation) to face the opposite direction. Our field is a sine wave, however, and the current will change when we go past perpendicular to flow in the “opposite direction.” In this case, the coil will try to torque back. In the steady state, we will find, with a quick enough oscillation, that the coil will be perpendicular to the oscillatory magnetic field. The current flowing through the propulsion coil reduces to zero. If the torque causes the coil to overshoot, any current would fluxuate with the magnetic field, causing a torque to return it to its equilibrium.

How do we reduce the torque to zero? by using two orthogonal coils. If the two propulsion coils are orthogonal, a little analysis gives us the total torque of the robot. Assume both coils have the same area and same number of turns. The whole apparatus is off by 10°. The magnetic flux going through the parallel coil (parallel to the induction coil) is \( BA \cos 10° \). The flux going through the perpendicular coil is \( BA \sin 10° \). The torque is related by cross product of the magnetic moment and the external field. The perpendicular coil has \( \tau = BN_\mu I \sin 80° \), the parallel has \( \tau = BN_\mu I \cos 80° \). By substituting the respective currents, we find that both torques are \( \tau = B^2 \mu_0 A \sin \theta \cos \theta = B^2 \mu_0 A \frac{\sin 2\theta}{2} \).
This shows that the torques in both coils are zero if either coil is perfectly parallel. When at an angle, the total torque is still zero.

Figure 4 - Torque on two coil Passive System

The current in the coils causes a magnetic field that points away from the external field. This causes the magnetic moments to point away from the field change. As seen in Figure 4, the induced fields in the coils cause a magnetic moment pointing away from the direction of the field. They look sort of like fletching on the arrow of the field. A torque is induced to reduce the angle between the propulsion coils; to bring them together. Since the coils are mechanically kept at 90°, the torques operate in opposite directions, so the overall force is zero. As a consequence, to reorient the robot, we just need to turn off one of the coils, and the robot will reorient itself.

2.3 Active Robot

As pointed out before, two parallel conductors with current traveling in the same direction are attracted to each other. In fact, the definition of the SI Unit Ampere is defined using such terms. An active robot must use this form of attraction. Our system has two powered coils on both sides of the robot. They are wired so that the magnetic fields generated in the same oppose. This causes a magnetic field gradient that is relatively constant. Our robot works by detecting this magnetic gradient and powering a propulsion
To detect the field gradient, I originally used two solenoids of magnetic wire. Placed so that the loops of wire are perpendicular to the gradient we are looking for, the voltage induced in the coils is proportional to the change in magnetic field. The flux should vary inside the coils according to the overall gradient of the magnetic field. By comparing the induced voltages, we can obtain the direction of the field gradient. Using the remote control system to find the intended direction, we can find which way to power the coils by taking the exclusive-or of the gradient and the desired direction. This becomes the input to an current amplifier or an analog switch.

This digital signal, however, is not what we need to power the coil. This gives the information needed to oppose the derivative in the magnetic field, not the magnetic field itself. When the magnetic field change opposes the direction of the field itself, we get the correct signal, as in Figure 5. But when they face the same direction, we get the opposite answer. We have a signal that is again delayed by 90°. Using an RC filter, we can delay the circuit, but only if we can guarantee a good input signal. Our signals had a lot of noise due to the power circuitry, internal magnetic fields, and variations of the coils.

![Figure 5 - Measurement of solenoids.](image)

To get the correct signal, using Hall Effect transducers is much better. The Hall Effect explains the phenomenon of voltage being induced across a current carrying wire in the presence of an orthogonal
magnetic field. The voltage induced is proportional to the magnetic field. These sensors are very small and are used for detecting gear teeth and building proximity sensors and contactless switches. These sensors use embedded magnets and produce digital outputs. Our system would need to use the linear transducers commonly used in position sensing equipment, others are analog current sensors. By placing a current of wire through these sensors, the magnetic field given off is measured and returned as a voltage. By placing two Hall Effect sensors on the robot, we can determine exactly the magnitude and direction of the magnetic field and the magnetic field gradient. This would obviate the need to delay the incoming signal. If we can tune a single sensor to measure the external magnetic field, while ignoring field of the propulsion coil, a lot more flexibility in the external field can be maintained. As of now, this requires the field to be in the less efficient opposing system.

3. Circuits

My thesis deals with the building of the initial prototype of a magnebot. This section contains details of how the circuits used were designed and built.

3.1 Function Generator

The first circuit needed was a function generator for the magnetic coils. The requirements were extremely low distortion, variable frequency and amplitude, outputs with variable phase, and high current output. Common VCO/function generator chips do not provide low enough distortion in the sine wave outputs. All our distortions turn into wasteful heat inside the power circuitry. To overcome this, I chose to use a common function generator chip. The ICL8038 manufactured by Harris Semiconductor includes square-, triangle, and sine-wave outputs. The linearity of the '8038 chip’s triwave output is about .1%, whereas the distortion of the sine wave is 1%.

In addition, building a quadrature oscillator\(^1\) is easier from a triangle wave than a sine wave. To generate a cosine wave from a sine wave, a simple op-amp integrator does a less than adequate job. The amplitude and distortion increase dramatically and minor fluctuations in the sine wave lead to enormous changes in output. To generate the sine and cosine outputs concurrently, trigonometric function generating chips make it much simpler. The AD639 Universal Trigonometric Function Generator made by Analog Devices is flexible enough to do this. It can generate nearly all of the trigonometric functions, including the more obscure ones like arcsecant and versine.

---

\(^1\) A quadrature oscillator produces a sine and cosine wave at the same frequency and amplitude. Actually, any generator of two waves that are exactly 90\(^\circ\) out of phase, be they triangle, square, or sine, is a quadrature oscillator.
The AD639 is a very complicated chip. The nominal output is a ratio of sine waves; \( \text{sin}(X1-X2)/\text{sin}(Y1-Y2) \). The output is buffered using a built-in op-amp. It uses \( 50^\circ /V \) conversion on angle inputs.

Figure 6 - Circuit Diagram for Function Generator
To generate a sine wave from a triwave, the output must swing from -1.8V to +1.8V. The triwave-output of the ‘8038 was scaled down using a high precision potentiometer. The simple circuit for the sine generator is contained in Figure 6. The cosine wave is generated using the same output, but has the square wave output of the ‘8038 attached to X1 and the triwave to X2. If the square wave is 3.6Vpp and switches quickly at the zero points of the triwave, the output will be a cosine wave.

The square wave output of the ‘8038 is open collector, and swings from -15 to +15. Scaling the wave down to 3.6V turned out to be a major problem. Using a voltage divider doesn’t work well since the pull-up resistor causes the output to be asymmetric. Since the output is rail-to-rail we can’t use a standard op-amp to buffer the circuit. The first solution was to wire the square output to a +1.8 voltage source and use a voltage clamp to reduce the negative output to -1.8V. The AD639 conveniently provides a 1.8 volt output, which is used to keep the denominator of the sine ratio at unity. This solution worked well, but using the simple diode clamp caused the rise time of the square wave to be very large (2μs). The wave’s distortion at 1kHz was so high as to make proper filtering a Herculean task.

The second attempt was to use dual tracking voltage regulators to power the 8038 at ±12V. The open-collector output can be scaled down after buffering it through an op-amp, and then scaled using a voltage divider. This solution was the final one. First, it further isolates the AD639 from the rest of the circuit, and reduces the rise time to only 1μs. The output of the cosine wave is relatively distortion free, except at the zero points when the square wave passes through zero. Spikes in the output lasted as long as the rise and fall time of the square wave. By adding a capacitor to the output potentiometer, the spikes are filtered out. The cosine wave’s distortion was around 1%, while the distortion of the sine wave was .1%. In our application, this was deemed within limits.

The outputs of the both AD639 chips are scaled using potentiometers and buffered using voltage followers. To generate arbitrary phase output, you just need to use a potentiometer and connect the two ends to the sine and cosine wave. The tap will be a phase offset between 0° and 90°. The other angles can be found by negating either the sine or cosine wave. The amplitude of the output can dip by up to 3dB, but since all the outputs are scaled by potentiometers anyway, this can be remedied using yet another potentiometer. Since the output of all three nominal outputs (sine, cosine, variable) use a op-amp voltage follower, the current output is as good as the op-amp. I used a TL084 quad op-amp.

I needed to generate two sets of these function generators, one for each axis. I placed them on the same protoboard, but the loading effects on the power supply increased the distortion. By adding in 1μF ceramic capacitors at the power supplies of each chip, the effects were reduced. The interference between the two chips wasn’t noticeable, except when the generators were placed too close to the current.
amplifiers, induction coils, or the nearby gigantic motor generator. The final circuit fit onto a small board.

Figure 7 - Picture of Dual Function Generator Board

3.2 Remote Control Circuitry

The robots are required to be untethered, so some form of remote control is needed. There are many remote control systems from which to chose. The largest, and most binding, requirement is that the number of chips required to decode be small. The electromagnets, batteries and magnetic coils should weight down our robots, not our control circuitry.

RC Car control systems were the first examined; but they have analog outputs and the decode circuitry required at least three chips to decode the signal and an antenna to detect the signal. The antennae placement and magnetic interference were also negative aspects of that solution. My previous work with autonomous robots (the MIT 6.270 competition) lead me to gravitate toward IR control. Modulated infrared signals are digital, relatively noise free, and interchangeable.
To simplify our lives, we decided to use the TV remote control standard RC5. Our work was cut in half since we didn’t have to generate the output signals; we used an off-the-shelf universal remote. The universal remote allows us to pick which flavor of the code is the easiest to decode. Appendix A contains a description of the RC-5 code and alternative IR formats.

Since the output of the IR decoder is a simple digital wave at around 1kHz, my initial instinct was to use an FSM to decode the signal. Having just taken 6.111, and not wanting to build a complicated microcontrolled system, I attempted to build the decoder so it fit simply into a single PAL. I chose the largest cheap PAL, the 22v10. It has 10 output pins, but no internal nodes. It is large enough, however, to build very complicated state machines. As the next month bore out, it is much easier said than done.

The number of bits in the signal is 14. Since the last bit sent needs to be checked, the minimum state machine would be a counter. This requires at least 4 bits of state. The complexity of the encoding system requires us to have at least 16 states. Using another bit of state gives us a lot more leg room to deal with the inevitable clocking problems.

The FSM has to be clocked at the rate of the incoming bits. Originally, to clock the PAL, two of the outputs were reserved for building a digital oscillator. Using a resistor, potentiometer, capacitor, and two digital inverters, a stable clock circuit is easy to build.² This uses two more of the PAL outputs. With seven of the outputs now used, only three outputs are available. This is enough to transmit the four directions and a valid bit. The simple FSM should be able to reject invalid signals and decode the correct ones. The initial PAL code, contained in Appendix C, contains an FSM that would work theoretically. In fact, it worked correctly 30% of the time. We’re building circuits, not playing baseball, however.

The timing of the clock had to be just right; the clock circuit was very sensitive to temperature. It would hold for about 20 minutes before starting to float too much. The remote control signal generator, the clicker, did not produce the code according to spec. The high state of each transition was much longer than the low state. Since the transitions occur at different times depending on the bit, we can’t guarantee being in the right portion of the transition. If the clock ends up ticking in the middle of the transitions, we may always receive high. Furthermore, the incoming signal from the decoder is not synchronized with the clock of the PAL. Without any more output bits to use to buffer the input, the logic was forced to use the bare input. Metastability problems were creeping into the system. Between the two problems, the chance of our FSM recognizing the signal is at most 75%.
After many attempts to circumvent metastability and chance, I caved in and spread the decoder over two PALs. The first clocking pal would be a synchronizer and a crude PLL supplying the clock signal for the second decoding PAL. The clocking pal would operate at a known multiple of the signal frequency and lock onto the two initial 1 bits. After recognizing the signal, it would send a clock signal to the decoder PAL so that it would pick up the value of that bit (i.e. the second half of the cycle). The decoder PAL is given the 12 bits after the initial bits and using a 4-bit FSM, can recognize the signal and provide all the outputs necessary. After it receives the last bit, the decoder PAL sends a reset signal to the clocking PAL, starting the cycle over again.

The first PAL initially used the same two-inverter clock oscillator as the old PAL, but I used a crystal instead of a capacitor to stabilize the clock output. I chose a frequency multiple of 12, or 6.8kHz. The initial bit detector ensures that it finds the output going from 0101, and then switches to a clock-providing mode. Every 12 cycles after detecting the valid signal, it sends out a clock. It will do so until it receives the reset signal. When it is reset, it sends out one more clock to reset the decoder pal.

Figure 8 - Picture of Remote Control detector board

---

2 Horowitz and Hill, 285
The decoder PAL is a simple FSM. It counts through 13 states, sends the reset signal, and returns to the start state, where it lies dormant until another signal is received. The output pins independently check the state and synchronized IR input to determine their correct state. At the second to last state, the whole signal is declared valid and the output is fixed. This PAL also has the capability of clearing the valid bit when the mute button is pressed, turning off all motion. This worked about as well as the previous system. The problem with all the circuits, I found, was a bad clock signal. Using the two inverter oscillator made a very stable circuit, but the clock edge would ring three or four times. I was running the circuit at far too low a frequency. Without a definite edge, I was getting multiple random clocking causing a simple test counter I built using the oscillator to fail miserably. By adding a separate oscillator, the circuit was working within a few hours. I used the 16-pin 74S124 VCO; a tad large, but available in large quantities.

The stability of the new oscillator was still the major fault with the circuit. Without any phase locking, any variance in the frequency would lead to garbage output. It took minutes for the oscillator to settle into a steady state. After tuning the potentiometer to 6.28kHz at the steady state, the decoding circuit would work with 100% accuracy. It took 75 seconds after cold start to start recognizing valid inputs, however.

To increase the reliability of the circuit, I changed the clock PAL to detect the transitions of the bits. This allows for correction of clock fluttering, which is a large problem with our poor oscillator. The PAL would operate with a wide of input frequencies, but produce the same output as the previous PAL. The new PAL uses an FSM and a 3 bit counter. At the start, we wait until we get a high input. After receiving two consecutive highs, we clock the detector since we are past the first transition. We then use the 3 bit counter to advance past the bit transition so that we are at the beginning of the next bit. I chose 7 cycles because that was closest to the 12x-frequency rate previously used. Depending whether we are currently high or low, wait until we reach another transition. After that, clock the counter again and restart. Figure 10 contains the FSM diagram.

The FSM continues like this until it receives the reset from the decoding PAL. Since the two initial start bits are sent to the decoding PAL, I had to reprogram it to ignore the first two bits. After these changes, the circuit would operate with a clock frequency from 6.25 kHz to 8.1 kHz before rejecting valid input. The final circuit uses the IR decoder, two PALs, a '124 oscillator, a potentiometer, a capacitor, and three LEDs. The final PAL code appears in Appendix C.
3.3 Magnetic Field Propulsion

The active robot uses powered propulsion coils to generate a magnetic field to repulse or attract the externally applied field in two dimensions. While the passive coil derived its current from the varying magnetic field, our active robot must have batteries to generate the force. To correctly power the coil, the robot needs a pair of orthogonal magnetic field detectors and current amplifiers.

The original magnetic field detector consists of two solenoids, in parallel. They had equivalent lengths, turns and diameter. They were attached directly to a LM393 dual comparator. The voltage across the coils is proportional to the change in the magnetic field, the gradient. By comparing the voltage in the coils, direction of the gradient can be found. The LM393 comparator produces digital output. In our test environment, the duty cycle was around 55%.

The LM393 produces two signals each providing the sign of the gradient along one dimension. To figure out which way to power the coils, we need to gate this information with the signals provided by the remote control decoding PALs. To conserve space, this combinational logic is placed on decoding PALs itself. The output of the comparator swings from -5 to +5V, so a voltage converter is needed to provide the PAL
with CMOS voltage levels. A voltage clamp is placed on the output so the negative output has a minimum of -0.6V. This signal is sent to the decoding PAL.

The decoding PAL has all the information necessary for determining which way to power the coil. For each direction, we take the exclusive-or of the gradient and the intended direction. The robot can only move in one direction at a time since the remote control circuitry allows only compass point movement. A multiplexer is used to switch between the directional inputs. When the VALID signal is enabled, the output of the PAL is the XOR of the Y-axis when VERT is enabled, the XOR of X-axis otherwise. This output is +5 to power the coil one way, 0 the other. The circuit diagram in Figure 11 shows the routing of the signals.

I originally connected this PAL output directly to an analog switch, the SW-06 made by Analog Devices. The SW-06 provides only about 100mA of current. This far below the amount of current needed to drive the propulsion coil. Since the propulsion coil has a very small resistance, the switch’s short-circuit protection kicked in to reduce the current and prevented any switching from taking place. Unless I added a 20 ohm resistor in series with the coil, it didn’t work. No other analog switches readily available provided more power, so I switched to using a current amplifier.

The power circuitry requires dual ±5V input, since we need to supply current to the coil in both directions. This conversion is required only because the PAL logic is used. The output of the PAL signals the input to an op-amp in an adding-inverting configuration with a gain of two. The negative power supply, -5V, is added in with unity gain. By using high valued resistors, 100 and 200K, the output rails to ±5V. This output drives both the propulsion coils.

The op-amp provides nowhere near the amount of current needed to drive the coil. To increase the current capabilities, a push-pull transistor pair is used to power the coil. To drive the coil in current mode, an op-amp and a sense resistor are used to ensure the base voltage provides the correct amount of current. The current going through the coil is proportional to the voltage in the circuit, up to the rise time of the op-amp. Due to the high current and small resistance in the circuit, I used a non-inverting op-amp configuration to increase the current.3

---

3 Horowitz and Hill, 186
I wanted to use MOSFETs to power the induction coil, but power bipolar transistors were cheaper and more plentiful. I used TIP31 and TIP32 transistors. They provide around 3A of continuous current, and
our system uses only around 1A. The sense resistor is rated at 5W and .1Ω. The voltage across this resistor is proportional to the current going through the main coil. This voltage is used as feedback into the op-amp that drives the base. I used a separate op-amp from the CMOS to ±5V converter. The TL084 quad op-amp is general purpose enough and small enough to fulfill our needs.

A larger power problem was the use of the ±5 power supplies. I used this to model using a 9 volt battery with an artificial ground. The amount of current drawn from the battery discharged it within minutes. For all testing, I used an external dual power supply at ±5V connected using flexible wires. This alleviated the power problem temporarily. Switching to four 3V lithium batteries with low-dropout regulators became the battery-powered solution.

At this point, we were still using solenoids; our output was still off by 90°. At this angle, we are not performing any net propulsion. To delay the circuit, we originally used a RC circuit with a time constant much larger than the period of the magnetic field. The resulting triwave has its maximum 90 after the incoming square-wave. The robot had extra connectors to allow variance of the resistor and capacitor used in the circuit, and to jumper the circuit when the switch to a different gradient detection system is used. This didn’t work very well at all. The noise in the solenoids caused the output to be very noisy and remain below ground nearly all the time. A different solution was needed.

![Picture of final Magnetic Field Detector and Power Circuitry](image)

**Figure 12 - Picture of final Magnetic Field Detector and Power Circuitry**

Very late in the project I caved in and switched to using Hall Effect transducers. They directly replace the solenoids. They are active elements with output directly proportional to the magnetic field. The null point, the output with no magnetic field, is 2.5V. For every gauss of field, up to ±500 gauss, the output increases 1.875mV. This is fine for most measurements. Micro Switch, a division of Honeywell, manufactures the ones used, and others with larger field strength capabilities. By placing these on the two
ends of the robot, the desired movement was found. The active robot moved in both all directions based on the input from a remote control.

4. Possible Enhancements
As the project progressed and the time pressure increased, I didn’t have enough time to perform all the optimizations necessary to improve the robot. This section contains these optimizations I couldn’t try, and other conclusions based on the robots behavior.

4.1 Function Generator
One thing you may have noticed is how the function generator has far more capabilities than used by this system. This extra functionality was built in to simplify optimizations later on. The quadrature oscillator was built to allow both coils on an axis to be powered at the same frequency, but at different phases. What would this do? From Figure 2 we see the tradeoff between picking the coils to have current flow in the same direction or opposite directions. If they flow in the same direction, there are many times when there is no magnetic field at all. Our robot, however, is still sending 1A through its coils. This is a large waste of energy. Powering the coils in opposite directions doesn’t alleviate this problem either, but adds another. In the center of the bowl, the magnetic field is always zero. These are the extremes, when the coils are powered at 0° of 180° phase difference. By varying the phase of the coils, the zero points (which will inevitably happen), can vary in space as well as in time. This will increase the efficiency of the robot.

The low-pass filter on the cosine AD639 should be changed. It operates unbuffered, attached to the signal portion of the output potentiometer. This has the affect of changing the filter based on the desired signal amplitude. This is very bad form. A new circuit should have the output buffered before it is filtered. I chose to do this because in our system, the amplitude of the signals is going to be relatively high and the frequencies chosen are relatively well known. The capacitor used works well from 50% to 100% of the maximum amplitude up to around 2kHz. So the filtering doesn’t change much, but the output amplitude does change based on the driven frequency. This was the tradeoff I made for simplicity.

4.2 Radio Control
The remote control circuitry is far too large. The main culprits are the two large PALs. The number of needed input pins is two and the number of needed output pins is three. So an 8-pin DIP contains the correct number of I/O pins, but most common PALs require all outputs of equations to be placed on an output pin. Without internal nodes, our state has to be wasted on unconnected pins. To change this, we
can switch to a fancier and smaller pal with internal nodes, or switch to a small PIC microprocessor. PALs exist with enough internal nodes to build the two necessary FSMs, but they expensive and rare. With the movement toward FPGAs, smaller programmable logic will be harder to find. Using a PIC is probably a better way to go.

Using nanoprocessors allows smaller chip sizes and more flexibility in the codes used. The newer models of PICs have 8-pin DIP sizes. They are easily programmed and can hold up to 4K of code. On the Internet you can find many programs for easily programming PICs. At the MIT Media Lab, a version of Logo is used to program the processors. When I worked there, I used a PIC that could decode Sony remote control codes, so well tested code existed. But, interfacing with PIC processors is much harder than with PALs. With so few pins, a mediating chip would be needed to latch the output signals provided. The microprocessor couldn’t provide the logic needed to drive the propulsion coils either. A 12v4 or smaller PAL could be used to provide both function. This would reduce the decoding section from two 24-pin chips to a 16 pin and an 8 pin chip; a reduction of half.

The ‘124 is a 16-pin dual VCO, but we are only using one side of it. The ‘8038 would be smaller, but require twice the number of components, due to the duty cycle compensation resistors). Using the simple 555 timer chip would have also worked but isn’t voltage controlled. The ‘724 mini-dip, though rare and expensive, is the same size as the 555 and requires only a capacitor and a resistor. These last two chips have 8 pins cutting the size needed by half. The Sharp IR decoder itself is rather large, as can be seen in the picture of the circuit in Figure 8.

With a custom transmitter and receiver, the number of components needed on both sides could be reduced further. On the robot, a nanoprocessor could do the PLL and demodulation functions of the Sharp IR decoder in software. Software like this is much more complicated, and the oscillator has to be run much faster, consuming more power. Using a custom IR transmitter, the size of the codes sent and the complexity of the logic needed to decode could be reduced. Using a joystick input, more directions of movement than just the four compass points could be sent. The next prototype should upgrade the remote control system.

The infrared transmissions of remote controls all operate near the same frequency, 40 kHz. Controlling two robots at the same time is hard without coordination. Two remote controls operating at the same time cancel each other out. The robots can be trained to select different signals, but they must be sent in serial to work. Furthermore, fluorescent lights cause interference that isn’t easily filtered out. In the future, these robots will not operate in glass cases, but inside the body. Infrared transmission will obviously not
work. Switching to radio control is the simplest solution. By tuning different robots to different frequencies, many robots can be independently controlled.

I didn’t use radio waves in the current system for many reasons. First, it is an analog system; using RC Car radio transmitters would work but I didn’t want the robot to operate in a mixed mode environment. Second, the receivers for these signals require antennae, in addition to FM demodulators and assorted other decoding equipment. I didn’t know how the large magnetic field changes would affect the reception of the antennae. I also figured the number of chips needed to decode the radio would be larger than those needed to decode IR. Future prototypes will have to switch to radio control, so the problem should be investigated soon

4.3 Magnetic Field Propulsion

The use of two Hall Effect transducers greatly increased the efficiency of the system. However, detecting magnetic field gradient forces the magnetic coils to work in opposition. The magnetic field in the center of the cube is zero, providing no force. Having the coils wired in the same direction would allow more energy transfer and a relatively constant magnetic field. The robot only needs knowledge of direction of the magnetic field, not the gradient. The transducers used, however, are not designed for the detection of orthogonal magnetic fields, but magnetic fields in general. The field across the chip, not just through it, is registered. By not having the same measurement space as solenoids, they can’t be used as effectively in two dimensions. Using the gradient reduces this effect, since the gradient generated by the off-axis coils is rather constant.

Switching to a single transducer would also allow analog detection of the field strength. Whether the field is near zero or at full power, the robot uses the same amount of current. This is wastes battery power and unnecessarily heats the transistors. By modulating the current through the propulsion coils based on the field strength, less force is generated, but the efficiency is increased. Using MOSFET transistors should also reduce the power consumed.

5. Conclusions

The active robot moved with about the same force as the active robot. The passive robot weighed around 30g, and the active robot, without the battery pack, weighed about 70g. The difference in acceleration was about half, which is to be expected with similar force. The main force limitation was the current flowing
through the propulsion coils. By running at ±5V with 2W transistors, the current never got above 1.5A. The current going through the passive coils was around twice that value. While the force can't get much higher without using more voltage or large transistors, reducing the mass is the easiest way to increase the force. The quickest way to do this is through reducing board size.

An easy way to reduce the size of all the boards would be to use printed circuit boards. I wired most of the circuits pretty tight. Only using a two-sided board or surface mount components would save much space. Reducing the chip count, through optimizations or using custom chips, would reduce the weight of the robot further. By reducing the weight, the acceleration increases and the larger a battery the robot can hold. The magnetic field strength is another limiting factor.

An increase in the field strength could be accomplished by changing the phase of the coils. The extra variable phase output of the function generators makes this easy, but with only two amplifiers, I couldn't get it working in time. An amplifier with a higher voltage restriction could produce higher current. Increasing the number of turns in the coils could also increase the field strength. Adding ferromagnetic materials to the coils may be less efficient, but would boost the strength if used correctly.

The theory behind the magnebot is basic, but very hard to engineer. Many factors limit the power we can use; the power consumed has to be large to generate the forces necessary. This prototype shows how a remote controlled magnebot can be built, and where to go from here. Optimization will be a hard task, but should yield results quickly.
6. Appendices

6.1 Appendix A - Remote Control

6.1.1 Analysis
For simplicity, I didn’t want to build both ends of the remote control system. When building both
transmitter and receiver, more control is gained at the expense of reliability and simplicity. The decision
to use infrared transmissions for the remote control hinged on the wide availability of many different
kinds of cheap transmitters. Universal remotes can generate hundreds of different codes. I spend a few
days going through the codes in our Allegro Universal Remote Control; attempting to find one that would
be easy to decode. Here is the information I gained from experimentation and web searching. A lot of
this information can be found the IR remotes page at http://falcon.arts.cornell.edu/~dnegro/IR/. This site
is down as of the publication date, but a search of the Internet should provide similar resources.

All television remotes use a transmission wave at a frequency of 1-3 kHz modulated at 32-40 kHz. An
example of the code is given in Figure 13. Building circuitry to decode detect the IR transmission,
amplify, PLL, demodulate, etc. would be an arduous task. The number of chips and transistors would be
far beyond our size requirements. However, Sharp Electronics manufactures a series of IR decoders.
These little metallic boxes include the IR photodiode and all the circuitry needed to demodulate the signal.
It operates on +5V and its output is the active low digital output of the modulated signal. When it detects
a modulated signal, it produces a 0, 1 otherwise. The box, smaller than your thumb, allows digital
circuitry to decode the final signal and the button pressed.

![Intended output signal](image)

Infrared signal detected at 940nm wavelength

![Output of Sharp IR Decoder/Demodulator](image)

Figure 13 - Infrared Transmission

The transmissions we receive from the decoder can vary widely. Each manufacturer uses their own
encoding of bits for their own equipment. The major standard is RC80. It uses spaces between 1’s to
transmit bits. If the single period 1 is followed by a single period 0, then it is a 1; if it is followed by a three period 0, then it is a 0. Nearly all of these codes use 32 bits to encode their data, but the very nature of the code allows it to be variable length. There is a wide standardization to the internal structure of these codes. Using this code would have allowed multiple remotes to easily control multiple robots. However, the code length was prohibitive. I wanted to use a PAL to decode the signal. To decode RC80 would have required more firepower.

6.1.2 RC-5 Code

The RC5 code, however, usually uses less than 22 bits. It is fixed length, since it uses transitions to transmit bits. A transition from low to high in a period is a 1, high to low a 0. Each bit takes 1.8 ms, with the low and high periods taking 889μs. The Goldstar remote code I use only has 14 bits. This code is used by many other manufacturers, but they are all the same. If we are slick, we can use a 28 state FSM oscillating at a period of .889ms to decode the signal. With only 5 bits of state needed, a PAL can barely contain both FSM and outputs. The actual encoding has more variation on where the transitions occur, as seen in Figure 14.

![Diagram of RC-5 Bit Encoding]

**Theoretical Bit Encoding for 1 and 0**

![Theoretical Bit Encoding Diagram]

**Actual Bit Encoding**

![Actual Bit Encoding Diagram]

*Figure 14 - Bit Encodings for RC-5*
The 14 bit encoding for the Goldstar Remote was determined by reverse engineering. The first two bits are always 1, allowing the decoding circuitry to lock in on the signal. The next bit toggles with every button press. This is to cope with the mindless pressing of the channel up button in order to make the channels flick faster. The next 5 bits are the group number, signifying the kind of device. A television is 0, VCR is 1, and a CD 17. We used the TV Volume and Channel buttons, so the group number was 0. The final 6 bits contain the actual button pressed, also known as a command number. The following table is a list of the codes

Table 1 - RC5 Command Codes

<table>
<thead>
<tr>
<th>Command Number</th>
<th>Button Pressed</th>
<th>PAL Outputs (Valid, Vert, Pos)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0..9</td>
<td>0..9</td>
<td>---</td>
</tr>
<tr>
<td>12</td>
<td>Power</td>
<td>0,0,0</td>
</tr>
<tr>
<td>13</td>
<td>Mute</td>
<td>0,0,1</td>
</tr>
<tr>
<td>16</td>
<td>Volume +</td>
<td>1,0,0</td>
</tr>
<tr>
<td>17</td>
<td>Volume -</td>
<td>1,0,1</td>
</tr>
<tr>
<td>32</td>
<td>Channel +</td>
<td>1,1,0</td>
</tr>
<tr>
<td>33</td>
<td>Channel -</td>
<td>1,1,1</td>
</tr>
<tr>
<td>34</td>
<td>Prev. Channel</td>
<td>1,1,0</td>
</tr>
<tr>
<td>46, 28, 29</td>
<td>Menu, Menu +, Menu -</td>
<td>---</td>
</tr>
</tbody>
</table>

Figure 15 - RC5 encoding for Volume +

6.2 Appendix B - Measurements

6.2.1 Induction Coil

John Hu built the fishbowl and the current circuitry. I started the project after the passive system had been built. A tethered semi-active system, closing and opening the propulsion coils, but providing no power, was also built. This section contains the measurements of the fishbowl and the external coils as if they were a black box. From this approach, I developed the theory and the rest of the thesis.
The inside of the fishbowl measures 18.2 cm long and wide and 17.6 cm deep. It is made of ¼” waterproof clear plastic and sealed with epoxy. The water level is 5cm. The external coils are composed of 22 gauge magnetic wire wrapped into square coils measuring 15.6 cm wide and 17.1 cm tall. The total diameter of the coils is around 1.6cm. The coils are wrapped in electrical tape and attached to the four sides of the cube. The distance between the centers of opposing coils is 21 cm.

The coils are attached to a large switchboard. One switch connects the function generator to the current amplifier. Two other double pole switches connect the output of the current amplifier to one side of the coils. The other coil connection is connected to a large capacitor, rated at .47µF at 630 VDC. These capacitors, used to balance the reactance of the induction coils, have their other lead connected to ground. Two current amplifiers are used: identical Copley Controls Corp model 413H PWM servo amplifiers rated at 90V and 15A continuous.

The function generator supplies a sine wave to the current amplifiers. When switched on, the current amplifiers see the impedance of an RLC circuit at 10.2 ohms with a breakpoint frequency around 1.3kHz. The function generators, which are composed of voltage controlled oscillators, are adjusted to this breakpoint frequency using potentiometers. This adjustment is done by ear; when close to the resonance frequency, the sound from the circuitry can become deafening.

The current amplifier, although rated at 30A, can only produce 8A in the coils. This limitation reduces the largest possible magnetic field. However, the “extra capacity” of the current amplifier allows us to connect the amplifier to two of the coils and have the same current flowing through each coil, due to the reduced resistance. Instead of measuring the current through the coils, I used a test coil to measure the magnetic field produced.

**6.2.2 Magnetic Field**

The test coil is a 5 cm square, 5 turn coil of magnetic wire. Its resistance is .17 ohms. The voltage across the coil will give us the change in the magnetic field. By placing the test coil in the center of each induction coil, while only powering one coil at a time, we can derive the magnetic field’s derivative. Knowing that the magnetic field is a sinusoid, we can derive the magnitude of the driving field. The test coil was used for all the measurements in the following table.
Table 2 - Various Test Coil Voltages

All voltages are peak to peak voltages.

<table>
<thead>
<tr>
<th>Frequency</th>
<th>Left Coil (Vpp)</th>
<th>Right Coil</th>
<th>Fore Coil</th>
<th>Aft Coil</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.313 kHz</td>
<td>1.40 V</td>
<td>1.19 V</td>
<td>1.04 V</td>
<td>.77 V</td>
</tr>
<tr>
<td>1.333 kHz</td>
<td>1.33 V</td>
<td>1.41 V</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.258 kHz</td>
<td></td>
<td>1.28 V</td>
<td>1.21 V</td>
<td></td>
</tr>
<tr>
<td>1.340 kHz</td>
<td>1.33 V</td>
<td>1.30 V</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The maximum voltage from the test coil corresponds to the breakpoint frequency for each coil (which, as we have seen, are not identical).

\[
V = -\frac{d\phi}{dt} = -NA \frac{dB}{dt} = -NA \frac{d(\vec{B}\sin \omega t)}{dt} = -NA\vec{B}\omega \cos \omega t
\]

\[
\vec{B} = \frac{\sqrt{V}}{NA\omega} = \frac{1}{.0125 \cdot 2\pi} \frac{\sqrt{V}}{f} = 12.7 \frac{\sqrt{V}}{f}
\]

This gives us an peak-to-peak magnetic field strength of .014 T. The \(B_{rm} \) would be around .01T. This is around 200X stronger than the earth’s magnetic field. Since our robot generated around 1 Amp in its propulsion coil (which is a meter in length), the maximum force generated would be around .01 N.

When the test coil was placed along the center x-axis, the voltage observed was within 5% of the theory’s predictions. These tests were done with both one and two powered coils. The distance between the coils is larger than two radii, closer to 2.5. When the coil is placed directly in the center of the fishbowl at 1.340 kHz, the observed voltage is .74Vpp. We would expect to see about 1.00Vpp if the coils were two radii apart. When the actual distance is used in the theoretical equations, the observed voltage should be .77Vpp.
6.3 Appendix C - Logic Code

This appendix contains the final code used to program the remote control decoding PALs. The format used is Abel 4.2. The first two listings are the code for the PALs used in the final prototype. The next listing contains the initial unreliable clocking PAL. The final two listings are the attempts to build the decoding FSM on a single PAL. The first operates at double the bit rate, exploiting the transition encoding. The second is an earlier attempt to clock at the bit frequency, taking our chances that we are examining the correct portion of the signal.

Listing 1 - Final Clocking PAL

module rcclk
TITLE 'Remote Control Decoder Clock Generator PAL'
rcclk device 'P22V10';

*Input pins
CLK    pin 1;  "Connected to Pin 23 (Which is the oscillator
!INP   pin 2;  "Output from IR detector (Inverted)
RESET pin 3;

*Outputs
Q0     pin 15;
Q2     pin 16;
Q1     pin 17;
C0     pin 18;
C1     pin 19;
C2     pin 20;
RCOUT  pin 21;
CLKOUT pin 22;
DRESET pin 23;

Q=[Q0..Q2];
CTR=[C2..C0];

Q isType 'buffer';
CTR isType 'buffer';

"This chip assumes that it is going to get clocked at a standard rate
"Some fundamental multiple of the frequency of the signal
"This chip then uses the first two bits of the signal 0101 to lock
"On to a correct clock signal at the correct frequency.
"Assume we are clocking at 12X the actual frequency
"We are building a state machine that will wait until it receives a 1
"Then it will wait another 12/HighDutyCycle-1, make sure we have another 1
"and then wait another few cycles, make sure we have a 0 for the remainder
"of that cycle, and then get a 1 for the rest of the cycle. Once we have
"This, we turn the VALID output on. Now, we go into a different mode
"At the correct frequency of the signal, we produce a clock signal
"This should occur late in the cycle so that the second chip always gets
"a correct signal. We will produce the clock until we get a reset
"signal, then we reset back to the initial state of looking for the start
"signals.

"As before, the inverters are used to generate the clock signal for the pal
H,L,X,C,P = 1,0,..X,..C,..P.;
FreqMult = 12;  "frequency multiple
ClockWait = 7; "After confirmation of bit, clock 6 times, then look again
Start = 0;
Trans = 1;
Valid1 = 2;
Is0 = 3; "This means that the next bit will trans from 1 to 0 (i.e. 0)
Trans01 = 4;
Isl = 5;
Trans10 = 6;
Valid0 = 7;

EQUATIONS
CLKOUT.CLK = CLK;
"When valid, produce CLKOUT when we got to ClockOffset
"And produce a clock on reset to let the decoder reset its state
RCOUT := INP;
RCOUT.CLK = CLK;
"This is a latch that syncronizes INP to the CLK. Thus reducing the
"Metastability for the output.
DRESET := RESET & !DRESET;
DRESET.CLK = CLK;
"This latch postpones reset so another clock is generated

state_diagram Q
State Start:
    CLKOUT := 1;
    if RCOUT then Trans else Start;
State Trans:
    CLKOUT := 0;
    CTR := 0;
    if RCOUT then Valid1 else Start;
State Valid1:
    CLKOUT := !DRESET;
    CTR := CTR + 1;
    if DRESET then Start;
    if (CTR == ClockWait) & RCOUT then Is0
    else if (CTR == ClockWait) & !RCOUT then Isl
    else Valid1;
State Is0:
    CLKOUT := 0;
    if !RCOUT then goto Trans10 else Is0;
State Trans01:
    CTR := 0;
    if RCOUT then Valid1 else Start;
State Isl:
    CLKOUT := 0;
    if RCOUT then goto Trans01 else Isl;
State Trans10:
    CTR := 0;
    if !RCOUT then Valid0 else Start;
State Valid0:
    CLKOUT := !DRESET;
    CTR := CTR + 1;
    if DRESET then Start
    else if (CTR == ClockWait) & RCOUT then Is0
    else if (CTR == ClockWait) & !RCOUT then Isl
    else Valid0;

end

Listing 2 - Final Decoding PAL

module rcdec
TITLE 'Remote Control Decoder PAL for Magnebots'
rcdec device 'P22V10';

*Input pins

CLK pin 1;  "Connected to Clock output of the rcClk pal
INP pin 2;  "Synchronized output from rcClk pal
INRES pin 3;  "Used only for the simulations
VERTGRD pin 4;  "Output from the vertical magnetic field gradient
HORZGRD pin 5;  "Outputs

SWITCHOUT pin 15;
QO pin 16;
Q1 pin 17;
Q2 pin 18;
Q3 pin 19;
VALID pin 20;
VERT pin 21;
POS pin 22;
RESET pin 23 isType 'buffer';

Q=[Q3..Q0];
H,L,C,P,X = 1,0,.C.,.P.,.X.;

Toggle = 1;
TestState1 = 4;
ChannelState = 8;
PositiveState = 13;
ValidState = 10;
NotValidState = 11;
ResetState = 14;

EQUATIONS

SWITCHOUT = VALID & (VERT & (POS $ VERTGRD)) # (!VERT & (POS $ HORZGRD));
*If the input is valid, and we are travelling vertically
*Then return the XOR of positive direction and the gradient
*Maybe this should be XNOR?

Q := !INRES & ((Q+1) & !RESET.Q);
*Increment Q, making sure that the 4th bit is a 0

VALID := !INRES & !(Q==NotValidState) & INP) & ((Q==ValidState) # VALID);
*Turn valid on is Q is at ValidState. Or clear if Q is at notvalidstate
*and the input is on (which can only occur for the shutoff button)

RESET := !INRES & ((Q==ResetState) # ((Q==TestState1) & INP));
POS := !INRES & ((POS & (Q!=PositiveState)) # (Q==PositiveState) & INP));
VERT := !INRES & ((VERT & (Q!=ChannelState)) # (Q==ChannelState) & INP));
*Reset is only on at the final stage (either Reset or failure)
*POS and VERT are latches on INP clocked at their respective states

test_vectors 'Test Channel +'
   ([CLK, Q, INP, INRES] -> [Q, VALID, RESET, POS, VERT])
   [C,X,0,0,1] -> [ 0,0,0,0,0];
   [C,X,1,0,0] -> [ 1,X,0,0,0];
   [C,X,0,0,0] -> [ 2,X,0,X,X];
   [C,X,0,0,0] -> [ 3,X,0,X,X];
   [C,X,0,0,0] -> [ 4,X,0,X,X];
   [C,X,0,0,0] -> [ 5,X,0,X,X];
   [C,X,0,0,0] -> [ 6,X,0,X,X];
   [C,X,0,0,0] -> [ 7,X,0,X,X];
   [C,X,1,0,0] -> [ 8,X,0,X,X];
   [C,X,0,0,0] -> [ 9,X,0,X,X];
   [C,X,0,0,0] -> [10,X,0,X,X];
   [C,X,0,0,0] -> [11,X,0,X,X];
   [C,X,1,0,0] -> [12,X,0,X,X];
   [C,X,1,0,0] -> [13,1,0,1,1];
   [C,X,0,0,0] -> [ 0,1,1,1,1];
Listing 3 - Initial Clocking PAL

module rcclk
TITLE 'Remote Control Decoder Clock Generator PAL'
rcclk device 'P22V10';

*Input pins

CLK      pin 1;  "Connected to Pin 23 (Which is the oscillator
!INP     pin 2;  "Output from IR detector (Inverted)
RESET    pin 3;

*Outputs

DRESET   pin 15;
Q0        pin 16;
Q1        pin 17;
Q2        pin 18;
Q3        pin 19;
Q4        pin 20;
RCOUT     pin 21;
CLKOUT    pin 22;
VALID     pin 23;

Q=[Q4..Q0];

"This chip assumes that it is going to get clocked at a standard rate
"Some fundamental multiple of the frequency of the signal
"This chip then uses the first two bits of the signal 0101 to lock
"On to a correct clock signal at the correct frequency.
"Assume we are clocking at 12X the actual frequency
"We are building a state machine that will wait until it receives a 1
"Then it will wait another 12/HighDutyCycle=1, make sure we have another 1
"and then wait another few cycles, make sure we have a 0 for the remainder
"of that cycle, and then get a 1 for the rest of the cycle. Once we have
"This, we turn the VALID output on. Now, we go into a different mode
"At the correct frequency of the signal, we produce a clock signal
"This should occur late in the cycle so that the second chip always gets
"a correct signal. We will produce the clock until we get a reset
"signal, then we reset back to the initial state of looking for the start
"signals.

"As before, the inverters are used to generate the clock signal for the pal

H,L,X,C,P = 1,0,.X.,.C.,.P.;

FreqMult = 12;
TS1 = 4;
TS2 = 10;
TS3 = 16;
IsValid = 24;
ClockOffset = 2;  "Clock base on 2 minus frequency (right near end)

EQUATIONS
Q := !DRESET &
  (!VALID & ((Q+1) & (Q!=IsValid)) &  "Increment while not valid
  !((Q==0) & !RCOUT) &      "Don't advance past 0 until high
  !((RCOUT & (Q==TS1)) &    "Make sure we are high at TS1
  !(RCOUT & (Q==TS2)) &     "low at TS2
  !(RCOUT & (Q==TS3))) #   "high at TS3
  (VALID & ((Q+1) & (Q+1<FreqMult))));  "If Valid, increment until FreqMult
Q.CLK = CLK;
"If we reset, go to state 1
"If we are not valid, increment our state until we get to IsValid
"Making sure that at TransPoint1, INP is high, and that at transpoint2,
"INP is low. If we are valid, then increment until we get to FreqMult

VALID := !DRESET & (VALID # (Q==IsValid));
VALID.CLK = CLK;
"When our nonvalid state gets to IsValid (i.e. to the last state)
"Then set valid

CLKOUT := (VALID & (Q==ClockOffset)) # DRESET;
CLKOUT.CLK = CLK;
"When valid, produce CLKOUT when we got to ClockOffset
"And produce a clock on reset to let the decoder reset its state

RCOUT := INP;
RCOUT.CLK = CLK;
"This is a latch that synchronizes INP to the CLK. Thus reducing the
"Metastability for the output.

DRESET := RESET & !DRESET;
DRESET.CLK = CLK;
"This latch postpones reset so another clock is generated

test_vectors 'Ignore glitches'
((CLK, Q, VALID, INP, RESET) -> [Q,VALID])
[C,X,X,0,0] -> [X,X];  "Preset with the better reset
[C,X,X,0,1] -> [X,X];
[C,X,X,0,0] -> [X,X];
[C,X,X,0,1] -> [0,0];
[C,X,X,1,0] -> [1,0];
[C,X,X,0,0] -> [2,0];
[C,X,X,0,0] -> [X,0];
[C,X,X,0,0] -> [X,0];
[C,X,X,0,0] -> [X,0];
[C,X,X,0,0] -> [X,0];
[C,X,X,0,0] -> [0,0];

test_vectors 'Ensure data detection'
((CLK, Q, VALID, INP, RESET) -> [Q,VALID])
[C,X,X,0,0] -> [X,X];  "Recall that INP lies one behind the output
[C,X,X,0,0] -> [X,X];
[C,X,X,0,1] -> [X,X];
[C,X,X,0,0] -> [X,X];
[C,X,X,1,0] -> [0,0];
[C,X,0,1,0] -> [1,0];
[C,X,0,1,0] -> [2,0];
[C,X,0,1,0] -> [3,0];
[C,X,0,1,0] -> [4,0];
[C,X,0,1,0] -> [5,0];
[C,X,0,1,0] -> [6,0];
[C,X,0,1,0] -> [7,0];
[C,X,0,0,0] -> [8,0];
[C,X,0,0,0] -> [9,0];
[C,X,0,0,0] -> [10,0];
[C,X,0,0,0] -> [11,0];
[C,X,0,0,0] -> [12,0];
[C,X,0,1,0] -> [13,0];
[C,X,0,1,0] -> [14,0];
[C,X,0,1,0] -> [15,0];
[C,X,0,1,0] -> [16,0];
[C,X,0,1,0] -> [17,0];
[C,X,0,1,0] -> [18,0];
[C,X,0,1,0] -> [19,0];
[C,X,0,0,0] -> [20,0];
[C,X,0,0,0] -> [21,0];
[C,X,0,0,0] -> [22,0];
[C,X,0,0,0] -> [23,0];
[C,X,0,0,0] -> [24,0];
[C,X,X,0,0] -> [0,H];
[C,X,X,0,0] -> [1,H];
[C,X,X,0,0] -> [2,H];
[C,X,X,1,0] -> [3,H];
Listing 4 - Double Frequency Monolithic PAL

module rc
TITLE 'Remote Control Decoder 1 PAL for Magnebots'
rc device 'P22V10';

"Input pins
CLK    pin 1;  "Connected to Pin 23 (Which is the oscillator
!INP   pin 2;  "Output from IR detector (Inverted)
RESET  pin 3;  "Place in position zero
NC3    pin 4;
NC4    pin 5;
NC5    pin 6;
NC6    pin 7;
NC7    pin 8;
NC8    pin 9;
NC9    pin 10;
INV_I2  pin 11;
INV_I1  pin 13;

"Outputs
INV_O1  pin 14 ISTYPE 'com';
INV_O2  pin 15 ISTYPE 'com';
Q0      pin 16 ISTYPE 'buffer';
Q1      pin 17 ISTYPE 'buffer';
Q2      pin 18 ISTYPE 'buffer';
Q3      pin 19 ISTYPE 'buffer';
Q4      pin 20 ISTYPE 'buffer';
OUT     pin 21;  " Out is on if output recieved
UD      pin 22;  " Will be on if it is up/down
PN      pin 23;  " Will be on if going positive direction
Q=|Q4..Q0|;

Start  = 0;
TestH1 = 0;
TestL2 = 1;
TestH3 = 2;  "Must be 101
Toggle = 3;
TestOut= 11;
CH1    = 13;
CH2    = 14;
VOL1   = 15;
VOL2   = 16;
PN1    = 25;
PN2    = 26;

"01011010101010101011000110101010001 Is a sample string we are trying to find
"Language definition
"All strings really start with a 0. A 01 corresponds to a 1, a 10 corresponds
"The first two bits are always 1. The third bit toggles with each keypress
"To distinguish between alternate presses. This toggle is currently ignored
"The next four bits are all zero (to tell you it’s a TV)
"The next two bits are of major consequence
"If bit 8 is high, then the volume buttons are being pressed
"If bit 9 is high, then the channel buttons are being pressed
"Bits 10 through 13 (4 bits) should be all zeros
"Bit 14 then determines whether it is positive or negative
" It is active low (i.e. when negative, it is a zero

EQUATIONS

INV_01 = !INV_I1;
INV_02 = !INV_I2;

" Remember that the state lags one behind the actual value

OUT := !RESET & (((Q.Q!=TestOut) & OUT) # ((Q.Q==TestOut) & INP));
PN := !RESET & (((Q.Q!=PN1) & PN) # ((Q.Q==PN1) &!INP));
UD := !RESET & (((Q.Q!=CH2) & UD) # ((Q.Q==CH2) & INP));

"Remember to add async_reset

state_diagram Q
state TestH1: if INP then goto 1 else goto Start;
state TestL2: if !INP then goto 2 else goto Start;
state TestH3: if INP then goto 3 else goto Start;
state 3: goto 4;
state 4: goto 5;
state 5: goto 6;
state 6: goto 7;
state 7: goto 8;
state 8: goto 9;
state 9: if INP then goto 10 else goto Start;
state 10: goto 11;
state 11: goto 12;
state 12: goto 13;
state 13: goto 14;
state 14: goto 15;
state 15: goto 16;
state 16: goto 17;
state 17: goto 18;
state 18: goto 19;
state 19: goto 20;
state 20: goto 21;
state 21: goto 22;
state 22: goto 23;
state 23: goto 24;
state 24: goto 25;
state 25: goto 26;
state 26: goto 27;
state 27: goto 28;
state 28: goto Start;

test_vectors([INV_I1, INV_I2] -> [INV_01,INV_02])
    [0,0]->[1,1];
    [0,1]->[1,0];
    [1,0]->[0,1];
    [1,1]->[0,0];

end rc

Listing 5 - Single Frequency Monolithic PAL

module rc
TITLE 'Remote Control Decoder 1 PAL for Magnebots'

rc device 'P22V10';

"Input pins

CLK        pin 1;  "Connected to Pin 23 (Which is the oscillator
!INP       pin 2;  "Output from IR detector (Inverted)
RESET pin 3; "Place in position zero
NC3 pin 4;
NC4 pin 5;
NC5 pin 6;
NC6 pin 7;
NC7 pin 8;
NC8 pin 9;
NC9 pin 10;
INV_I2 pin 11;
INV_I1 pin 13;

"Outputs

INV_O1 pin 14 ISTYPE 'com';
INV_O2 pin 15 ISTYPE 'com';
Q0 pin 16 ISTYPE 'reg_D,buffer';
Q1 pin 17 ISTYPE 'reg_D,buffer';
Q2 pin 18 ISTYPE 'reg_D,buffer';
Q3 pin 19 ISTYPE 'reg_D,buffer';
Q4 pin 20 ISTYPE 'reg_D,buffer';
OUT pin 21; * Out is on if output recieved
UD pin 22; * Will be on if it is up/down
PN pin 23; * Will be on if going positive direction

Q=[Q4..Q0];
Start = 0;
Valid = 3;
UDSt = 9;
PNSt = 10;

EQUATIONS
[OUT,UD,PN,Q].c=CLK;

INV_O1 = !INV_I1;
INV_O2 = !INV_I2;

" The state machine works like this
" Look for two bits that are both on. Then
" Ignore the next 6 bits. Set Out to 1
" If The 9bit bit is on, set UD
" Ignore the next 4 bits
" If the 13th bit is on, set PN

OUT := !RESET & ((Q.Q=Valid) # ((Q.Q!=Valid)&OUT));
UD := !RESET & (INP&(Q.Q=UDSt) # ((Q.Q!=UDSt)&UD));
PN := !RESET & (((Q.Q==PNSt) & INP) # ((Q.Q!=PNSt) & PN));

state_diagram Q
state Start: if INP then goto 1 else goto Start;
state 1: if INP then goto 2 else goto Start;
state 2: goto 3;
state 3: goto 4;
state 4: goto 5;
state 5: goto 6;
state 6: goto 7;
state 7: goto 8;
state 8: goto 9;
state 9: goto 10;
state 10: goto 11;
state 11: goto 12;
state 12: goto 13;
state 13: goto 14;
state 14: goto Start;

"Q := !RESET & ((Q.Q=0) & INP)
" # (2 & (Q.Q=1) & INP)
" # ( (Q.Q<14) & Q+1 & (Q.Q>=2));

test_vectors([INV_I1, INV_I2] -> [INV_O1,INV_O2])
[0,0]->[1,1];
[0,1]->[1,0];
7. Acknowledgements

I’d like to thank Professor Gill Pratt for bailing me out when my previous thesis evaporated, allowing me to finally get my Master’s Degree three months late. John Hu for showing me what needed to be done, but letting me figure it out for myself, Bryan Kincy for letting me steal his room and helping me with all the 3D rendered diagrams, and to all the people in the MIT Leg Laboratory for providing a fun and extremely cool place to work. And Michael Samuel Golinko, physicist, editor, carpenter.

To Holly, for putting me in my place and letting me know I was doing the right thing. My brother Brian, for letting me use the car. And especially Katie, for everything, but most of all for the distractions, ;-).

I’d like to thank all the people at Calico Technology, for my future employment.

A special word has to be given to Paul Horowitz and Winfield Hill. Their book, The Art of Electronics, is the best friend to a computer scientist trying to make it as an electrical engineer. It’s the only textbook I know of worth more than I paid for it.

But I’d like to thank my family most of all. My grandparents, for giving me the opportunity to go here. My father, for letting me understand me that M.I.T. was the right place to go. My mother, for allowing me to become myself, even though I don’t know what that is yet. You both taught me that an education was a terrible thing to waste.
8. Bibliography


Negro, David. IR Remotes Page. Http://falcon.arts.cornell.edu/~d negro/IR/