Problem

- Can the Skype experience be made more realistic? Can there be a way for someone over Skype to take more of a physical form or be able to move around?

Hypothesis

- The Skype experience can be greatly improved by being able to remotely control the location of the Skype phone.
  - This is because, often in Skype calls, we move the Skype phone or computer around to let the caller see and talk to other people. A more natural experience would be to be able to control the location of the Skype phone independently from the other end of the call.
- The location of a Skype phone can be remotely controlled over the Internet using Dual-tone multi-frequency codes (DTMF) and a Lego Mindstorms robot.
  - I believe that DTMF will work well for transmitting commands to a remote robot over Skype because it is designed to send commands over an audio link. This means that it can be plugged into the audio input of a Skype phone without changing the Skype software or hardware.
  - I believe that the Lego Mindstorms robot will work well to move the remote Skype phone because it is an existing programmable robot so I do not have to build my own robot. A robot is a very complex thing to build. Lego will also be a good way to build supports to hold the iPad Skype phone.

Background Research: Dual-Tone Multi-Frequency

- Dual-tone multi-frequency (DTMF) is a system designed to encode symbols over an audio medium. DTMF is the basis of telephone systems. As the buttons are pressed on the phone keypad, two different frequencies or tones are being produced simultaneously. Two tones are used to encode each key on the keypad.

Background Research: DTMF continued…

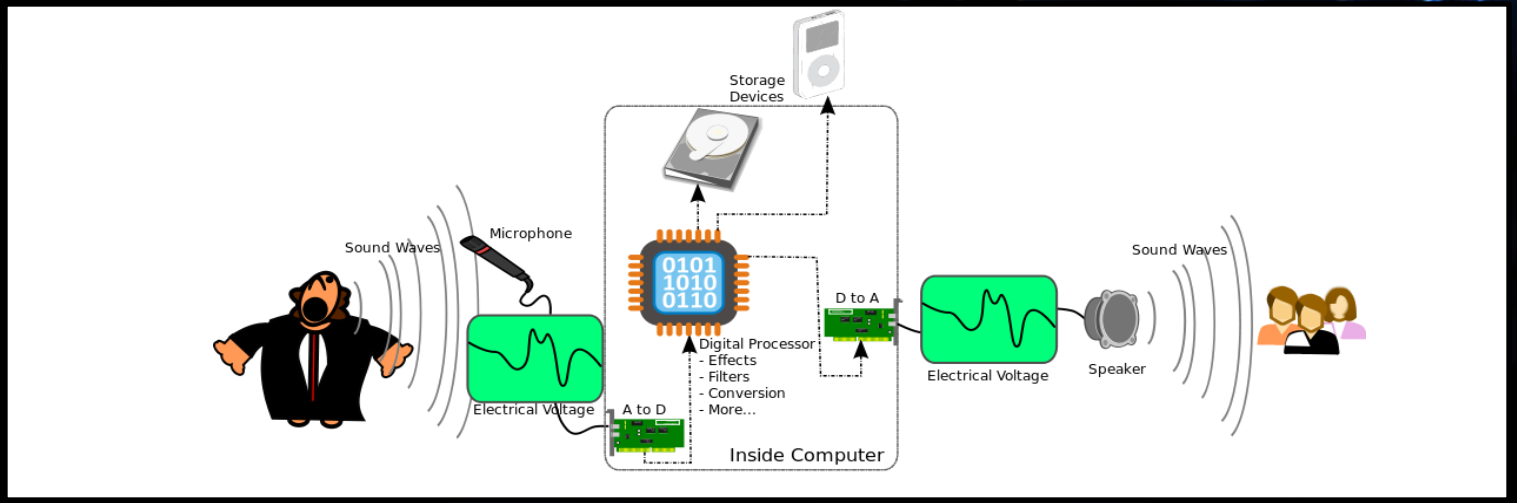
DTMF tones are organised as seen in the chart below. The first frequency is a row frequency and the second frequency is a column frequency. These two frequencies together identify the key pressed. For example when the key 0 is pressed, the row frequency 941 Hz and the column frequency 1136 Hz are produced simultaneously.

Dual-Tone Multi-Frequency (DTMF) table of frequency combinations

"Low Group" frequencies [Hz] \ "High Group" frequencies [Hz]	1209	1336	1477	1633	
697	1	2	3	A	(Row 1)
770	4	5	6	B	(Row 2)
852	7	8	9	C	(Row 3)
941	*	0	#	D	(Row 4)
	(Column 1)	(Column 2)	(Column 3)	(Column 4)	

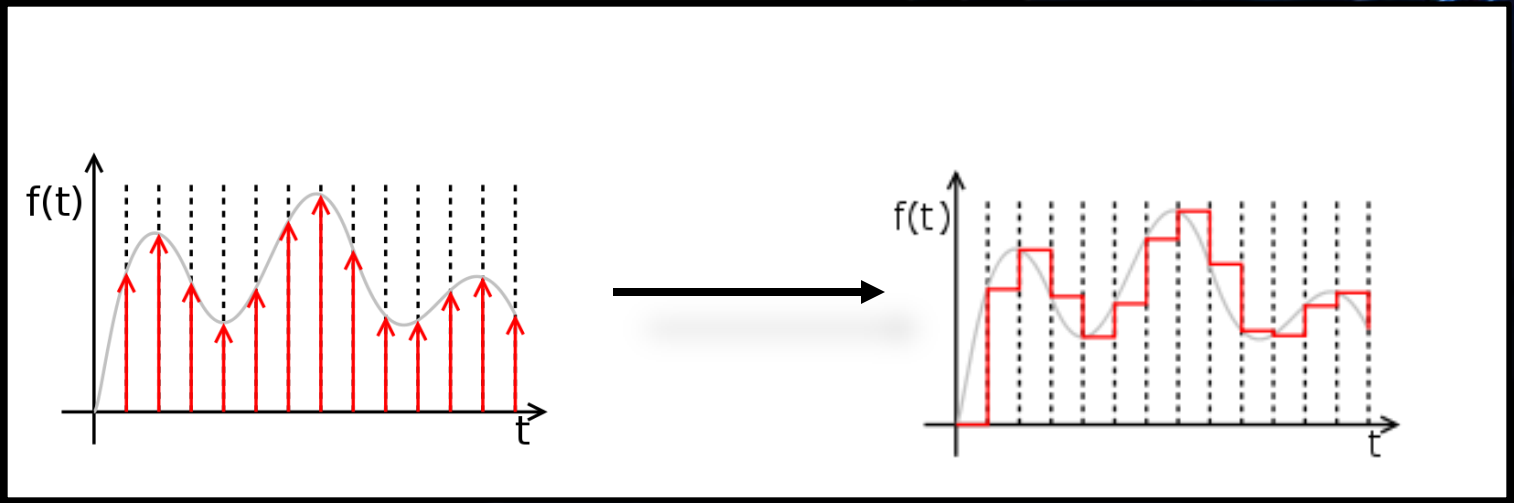
Background Research: Computer Generated Audio

- Audio comes in two different forms: analog and digital. Computers can only manipulate audio when it is in digital form. Audio in the world, however, only comes in analog form. Computers have codecs (coder/decoder chips) in their motherboards which convert the audio from digital signals to analog signals. This conversion is called DAC or digital-to-analog conversion.



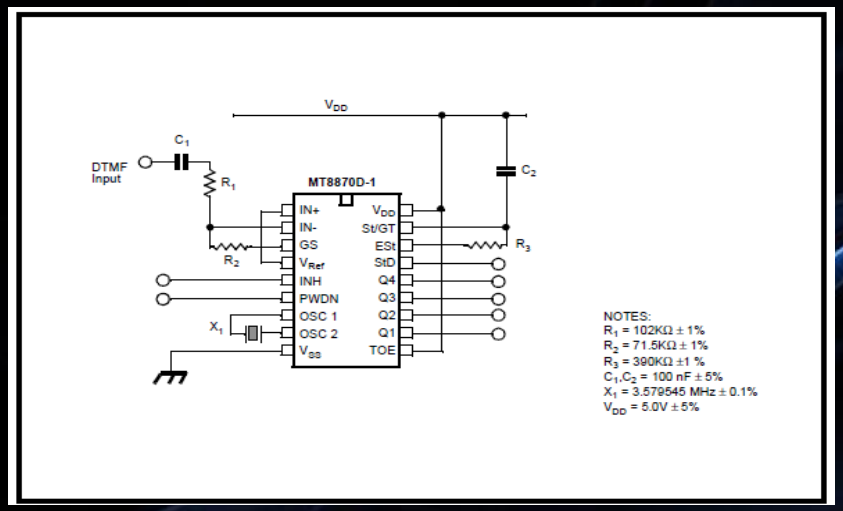
Background Research: Computer Generated Sound

- A computer can generate audio without making a recording by creating software that writes a sequence of numbers that represents an audio wave form (such as a sine wave) and directing them to the computer's digital-to-analog converter.



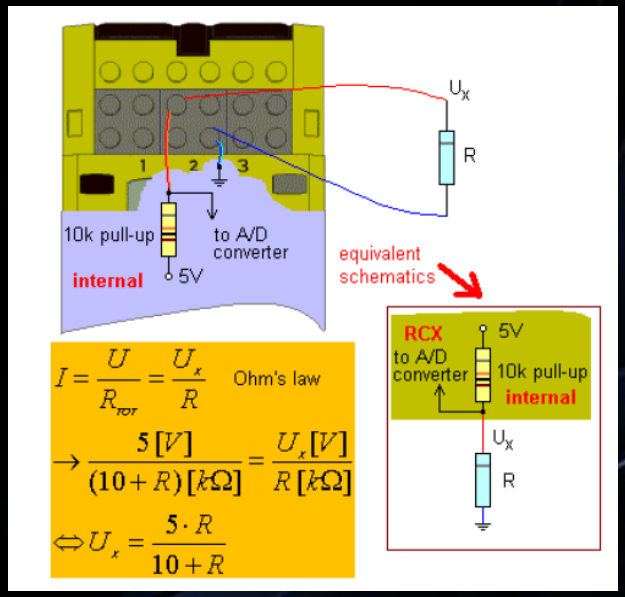
Background Research: DTMF Decoding Circuits

- The MT8870DE is an integrated circuit that can be used to build a DTMF decoder. The MT8870DE decoder circuit uses a digital counting technique to detect the frequencies and verify that they are the standard DTMF frequencies. The decoder detects the frequencies and produces the binary symbol equivalent to that tone on the keypad.



Background Research: Interfacing to Lego Mindstorms

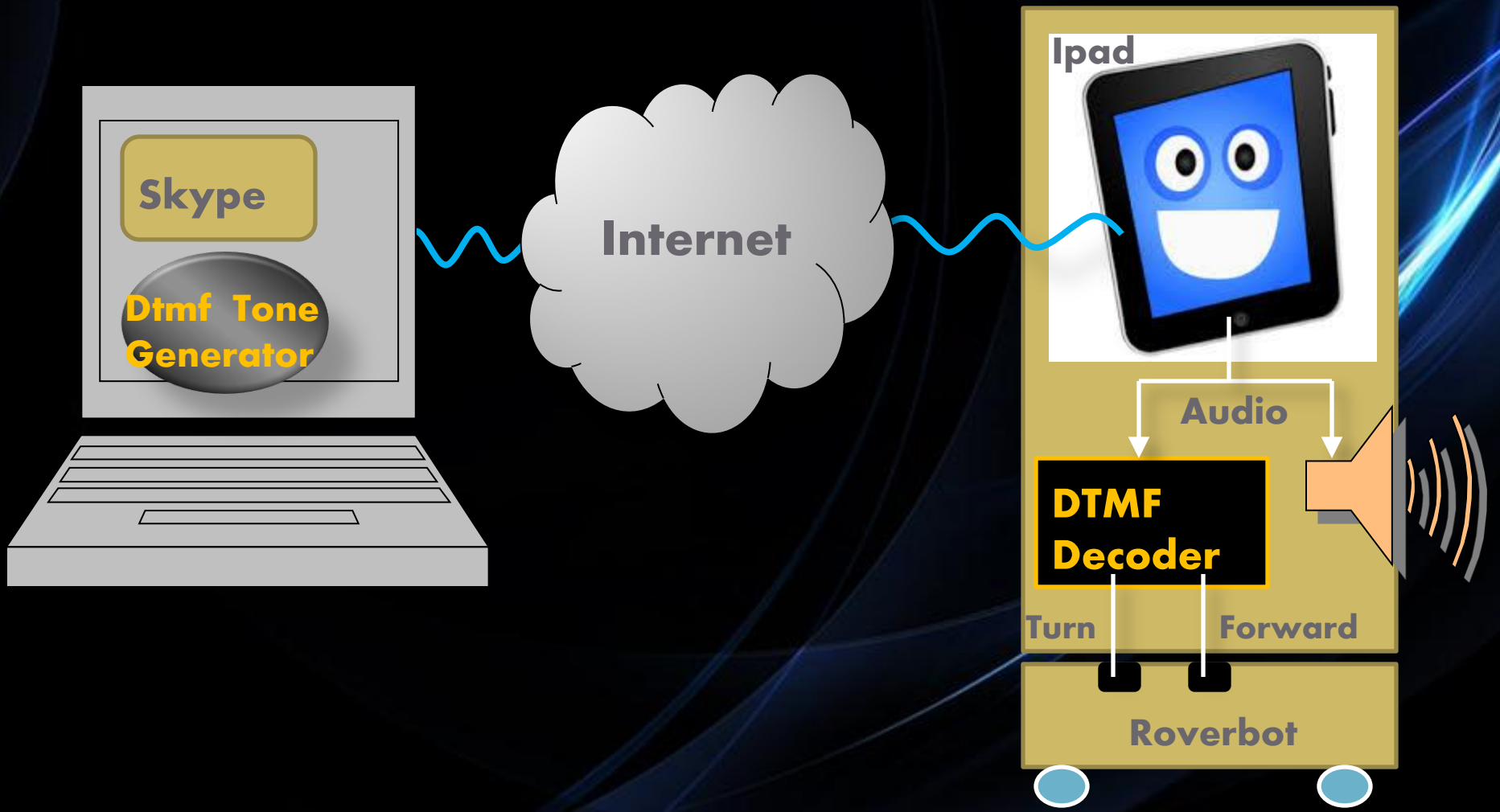
- The following circuit shows how the Lego Mindstorms sensor works when it is in unpowered mode. Here an external resistor in the sensor forms part of a voltage divider circuit. The Mindstorms RCX measures the divider voltage in an ADC (analog-to-digital converter).



Materials

- Lego Mindstorms roverbot
- iPad with internet access
- Laptop computer with internet access
- 32 bit PC computer
- Python 2.7 (IDLE GUI)
- DTMF decoder chip
- Detailed materials are available in the lab report

Procedure


Skype
Dtmf Tone Generator
Internet
Ipad
Audio
DTMF Decoder
Turn
Forward
Roverbot
Toronto
Ottawa

Imagine!

Internet

Toronto

Ottawa

Procedure

1. Create a robot on Lego Mindstorms and program it to move forward and to turn when different sensors are pressed.
2. Using Python 2.7, create a program that creates one DTMF audio code when the forward key is pressed and another DTMF audio code when the turn key is pressed and silence when any other key is pressed.
3. Create a DTMF decoder circuit based on the MT8870DE integrated circuit. It will monitor the audio output of the Skype application and detect when DTMF codes are transmitted from the other end of the call.
4. Connect the circuit to the two sensors on the roverbot so that the robot will move forward when the forward DTMF command is detected, turn when the turn DTMF command is detected and stop when no DTMF commands are detected.
5. The full detailed procedure is available in the lab report.

Observations: Skype Robot Testing

Test Description	Expected Result	Actual Result
1. Make a Skype call from the laptop to the iPad on the robot	Robot remains motionless Detection Light Emitting Diodes (LEDs) stay off	Passed
2. Hit the key "t" on the laptop end of the Skype call	The turn and the valid LEDs illuminate Robot turns right	Passed Sometimes detection LEDs flashed and robot stuttered
3. Hit the Spacebar on the laptop end of the Skype call	Detection lights turn off Robot stops moving	Passed
4. Hit the key "f" on the laptop end of the Skype call	The forward and valid LEDs illuminate Robot moves forward	Passed Sometimes detection LEDs flashed and robot stuttered
5. Hit the key "t" on the laptop end of the Skype call	The turn and the valid LEDs illuminate Robot turns right	Passed Sometimes detection LEDs flashed and robot stuttered

Observations:
DTMF Decoding Thresholds

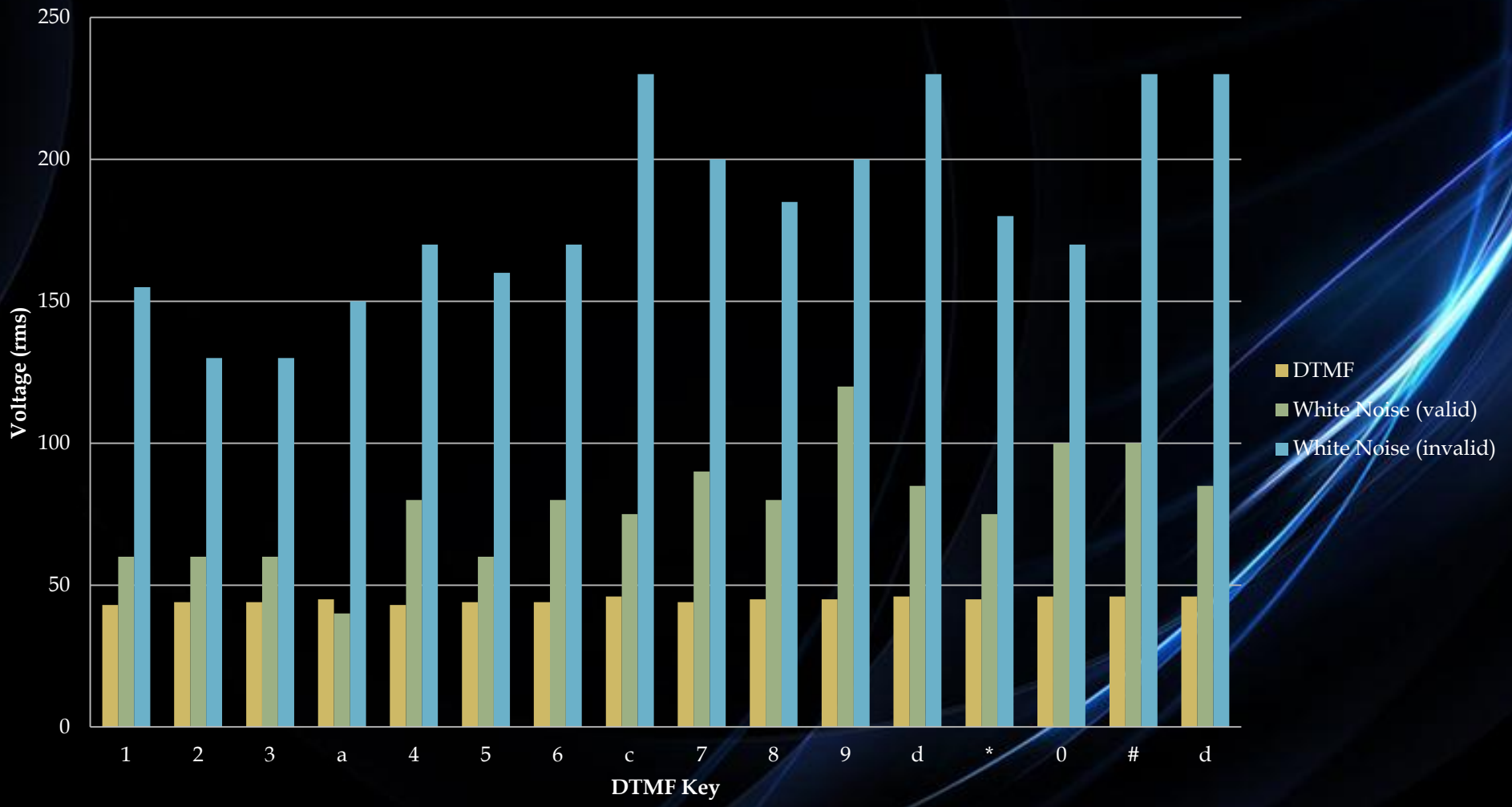
Keys	DTMF Voltage mV RMS	White Noise Voltage mV RMS (valid)	White Noise Voltage mV RMS (invalid)	Average of validity and invalidity
A	45	40	150	95
1	43	60	155	125
2	44	60	130	152.5
3	44	60	130	157.5
B	46	60	190	107.5
4	43	80	170	125
5	44	60	160	145
6	44	80	170	127.5
C	46	75	230	95
7	44	90	200	110
8	45	80	185	132.5
9	45	120	200	160
D	46	85	230	157.5
*	45	75	180	127.5
0	46	100	170	135
#	46	100	230	165

Observations: DTMF Decoding Sensitivity

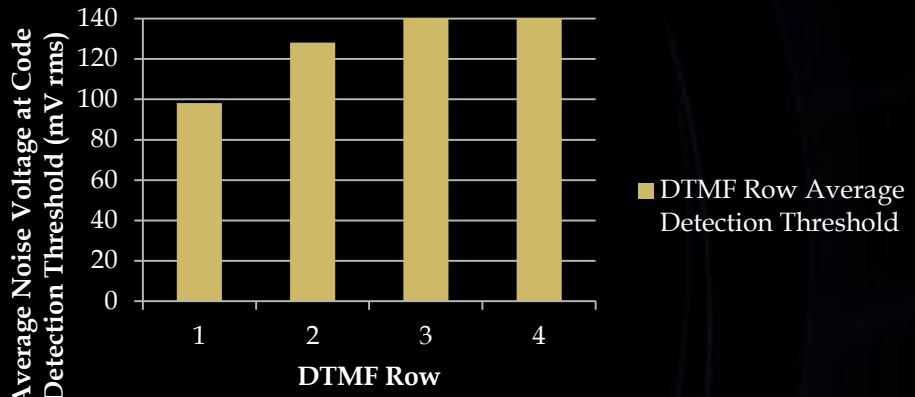
DTMF Voltage (mV rms)	Detected/Not Detected
50	Not detected
51	Not detected
60	Not detected
70	Not detected
83	Detected
100	Detected
200	Detected
300	Detected
400	Detected
500	Detected
600	Detected
700	Detected
800	Detected
900	Detected
1000	Detected
1100	Detected
1200	Detected

Observations: Number of Times the Roverbot Hesitates in a 7 Foot Journey

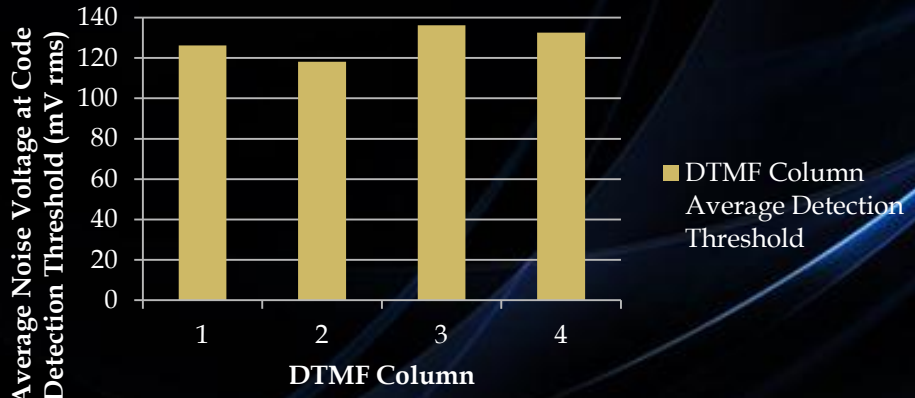
DTMF Circuit Detection Time Constant (The amount of time it takes to declare a code valid or invalid) (ms)	Number of Times the Roverbot Hesitates in a 7 Foot Journey		
	Trial 1	Trial 2	Trial 3
0	6	24	41
30	19	39	6
60	14	1	15
90	4	1	2


DTMF Detection Thresholds
250
200
150
100
50
0
Voltage (rms)
1
2
3
a
4
5
6
c
7
8
9
d
*
0
#
d
DTMF Key
DTMF
White Noise (valid)
White Noise (invalid)

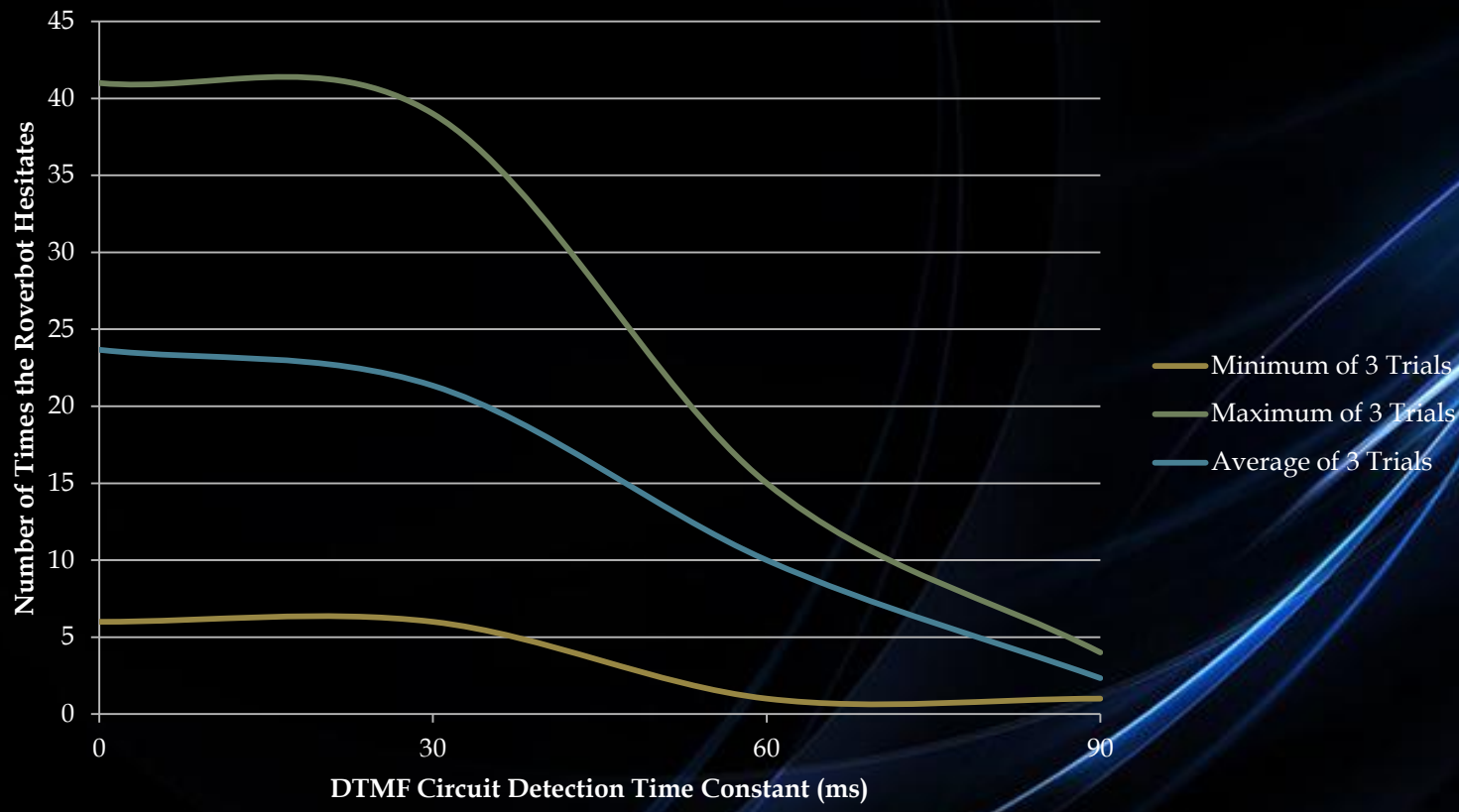
DTMF Row Average Detection Threshold



DTMF Column Average Detection Threshold



Number of Times the Roverbot Hesitates in a 7 Foot Journey



Discussion

- Codes with high thresholds for validity and invalidity are best. These high thresholds mean that the circuit would be able to detect the code with a great deal of background noise. The two DTMF codes that were best were the codes 9 and #. It was difficult to determine which codes were best, however, because some codes had a high threshold for invalidity but a low threshold for validity meaning that once the background noise has reached a level where the code is lost, it cannot be found again unless the background noise reduces significantly. This had to be taken into account so the codes that worked best were chosen because the average of the two thresholds was highest.
- Keys on the same row seem to have similar detection thresholds where as keys on the same column seem to have no correlation of pattern. There could be many reasons for this but the most likely reason may be an effect of the row and column frequencies used to create codes. The row frequencies are much lower (697, 770, 852 and 941 Hz) than the column frequencies (1209, 1336, 1477 and 1633 Hz). Inside the decoder chip, there are filters for each of the 8 DTMF frequencies and each of the filters has different characteristics. However, because the row frequencies are lower than the column frequencies, the differences between one row frequency and the next may be a lot more significant than the differences between one column frequency and the next.
- The roverbot hesitates often because of the audio fades in the Skype call which are something that will always occur and cannot be controlled. These fades may be due to the changes in wifi reception which is an uncontrollable variable. Although these fades cannot be eliminated, the effect they have can be reduced using longer DTMF detection time constants. They lead to fewer hesitations with the robot because longer DTMF detection time constants would mean that the circuit would take a longer time to declare the codes valid but it would also take a longer time to declare the codes invalid. When a time constant is much longer than the time of a fade, the code will be detected all through the fade.

Conclusion

- In conclusion, my hypothesis was strongly supported by the results. Using the Lego Mindstorms robot and DTMF control, I was able to fully control the motion of the iPad Skype phone by attaching it to a Lego Mindstorms robot and controlling it remotely over the Internet using DTMF codes.
- The Skype experience can be greatly improved; the added dimension of mobility brings us one step closer to virtual presence.
- There are also several conclusions dealing with the DTMF decoding circuit.
  1. Increasing the detection time constant is an effective way of compensating for audio fading, something that often occurs during internet Skype calls.
  2. The DTMF decoding circuit is more effective at detecting some codes than others. The two DTMF codes that were most effective were codes 9 and # because they both could be detected with very high levels of background noise.
  3. The DTMF decoder is also very sensitive. It can correctly detect the codes from 83 mV RMS to 1.2 V RMS.

For Next Time

- To be a useful product, the robot would have to be much larger to bring the robot up to the height of other people in the room.
- DTMF was a simple way to add controls without modifying the Skype software but what would be best is to modify the Skype software to include a control capability. This would eliminate the problem of audio fading affecting robot controls.
- More controls such as back and left as well as being able to change the camera angle would improve the experience
- A wide-angle camera that would make it easier to navigate from a remote location.