Maastricht University

# Xylobot

## Project 3-1

Ewan Demeur (i6155238), Rik Dijkstra (i6135069), Rhys Evans (i6150369),
Kailhan Hokstam (i6154002), Valentin Maican (i6150862),
Benjamin Rodrigues de Miranda (i6152983), Cas Teeuwen (i6169583)

# Contents

# Problem statement

Find the best way to configure a Xylophone-playing robot. Best, in this case would mean a configuration in which it is most enjoyable.

# Research Questions

- In what way can the robot be controlled, to yield the most accurate results at any given tempo?

- What  window size and hit detection thresholds gain the best performance in the Short-Time Fast Fourier Transform for detecting notes?

- What is the best way to tweak a Markov Chain for using it to play music based on what the user played?

# State of the Art

Xylophone-playing robot utilized for teaching autistic children to play music.

Significant improvement in in their skill level.

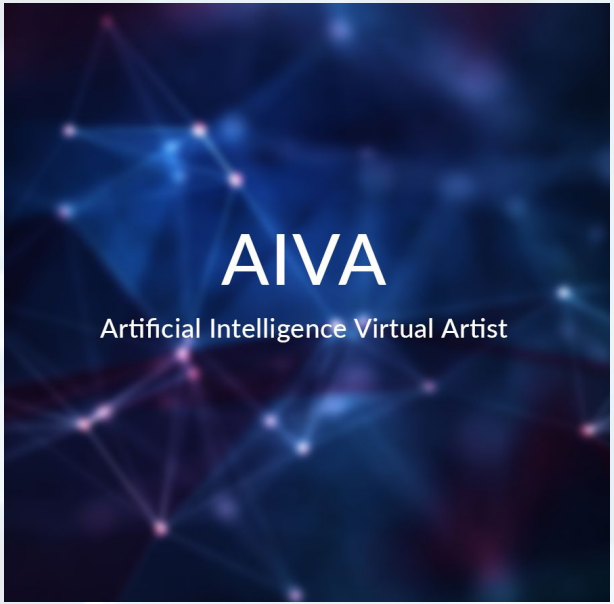
Teaching music to children with autism: A social robotics challenge A. Taheri

# State of the Art

When looking at the AI part of things, AI has become quite good at composing music.

"Aiva" is an AI that composes music using a neural network trained on large amounts of music.

It passed the 'musical Turing test' on multiple occasions.

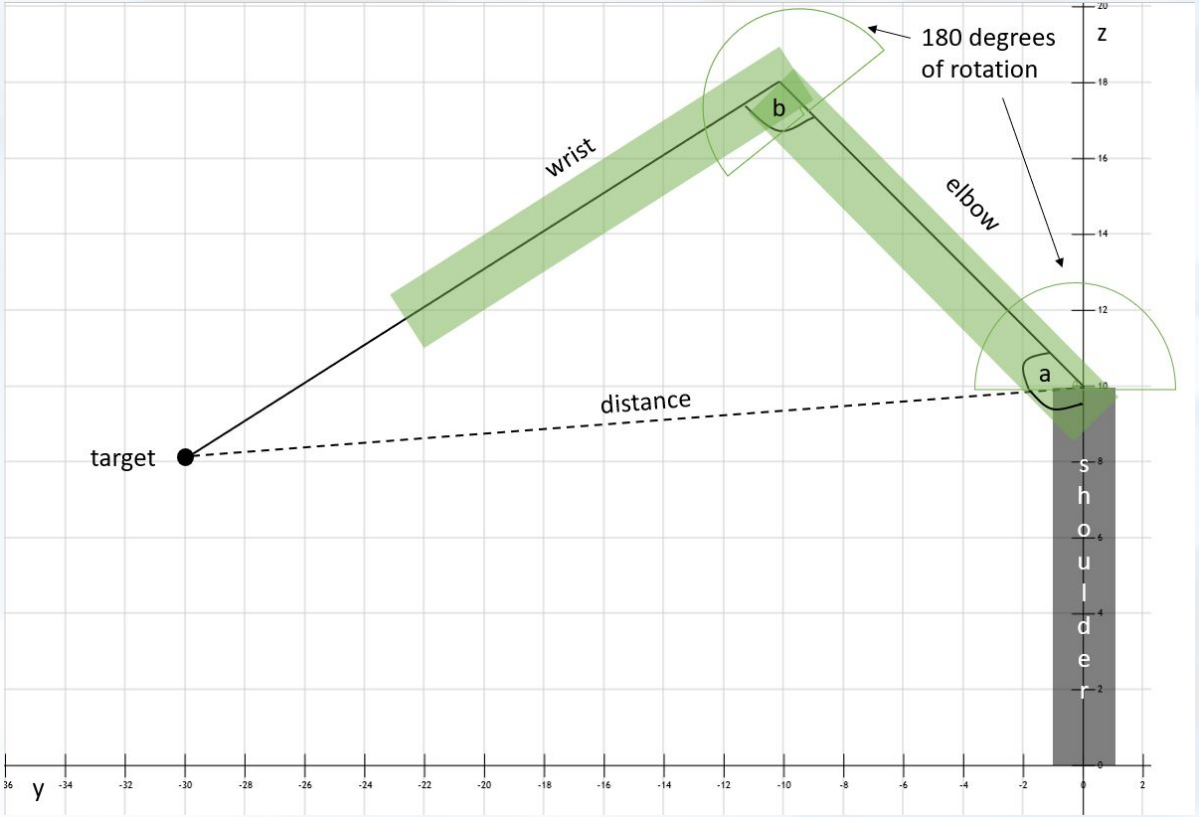



https://futurism.com/a-new-ai-can-write-music-as-well-as-a-human-composer

# Control

- Structure
- Safety layer
- Kinematics
- Hit methods
- Velocity & power control
- Class design
- Experiments

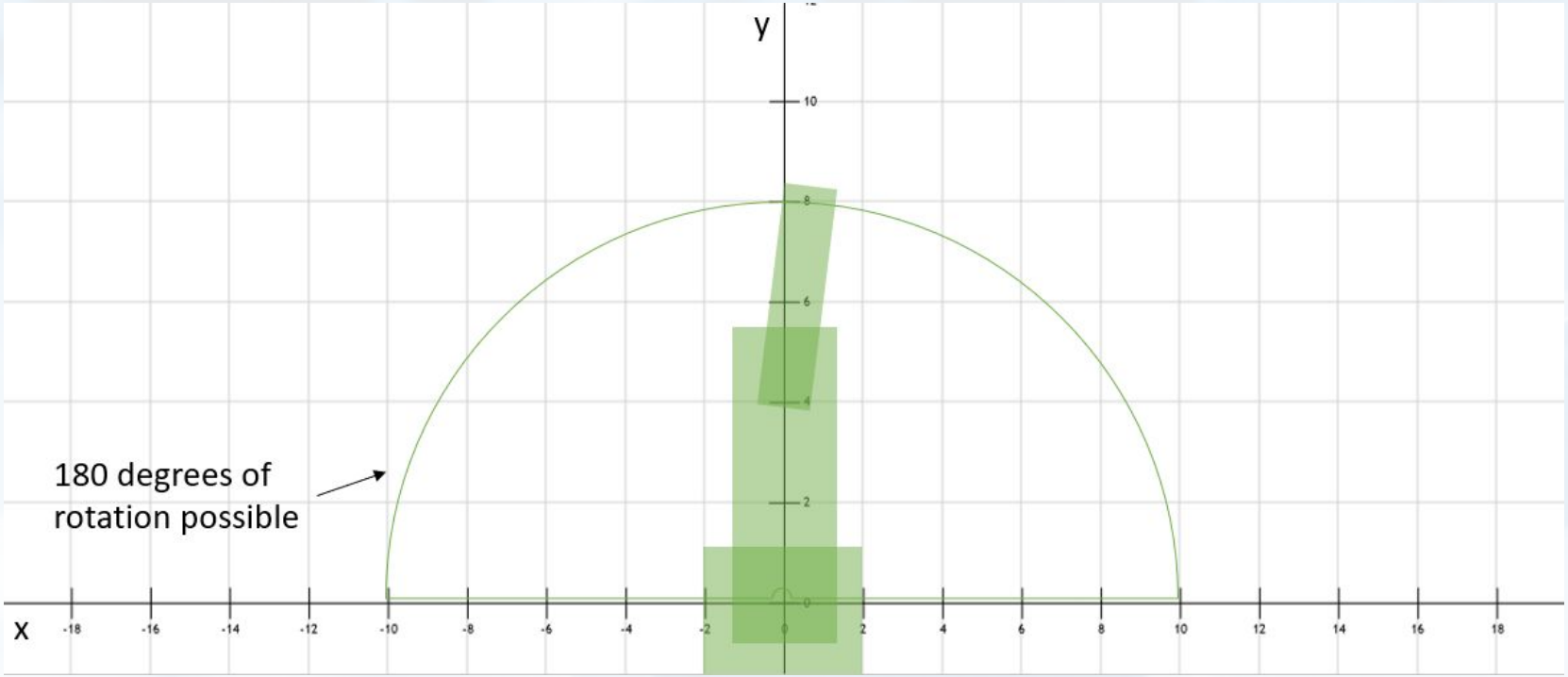
# Structure



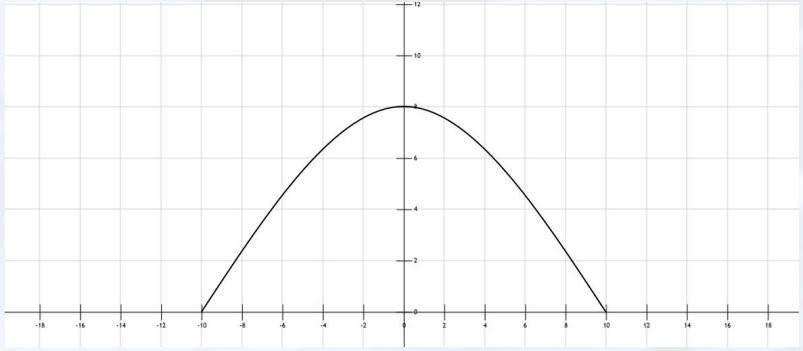

Shoulder - 18.8 cm
Elbow - 10.5 cm
Wrist - 18.7 cm

3 DOF in total
Arduino communication

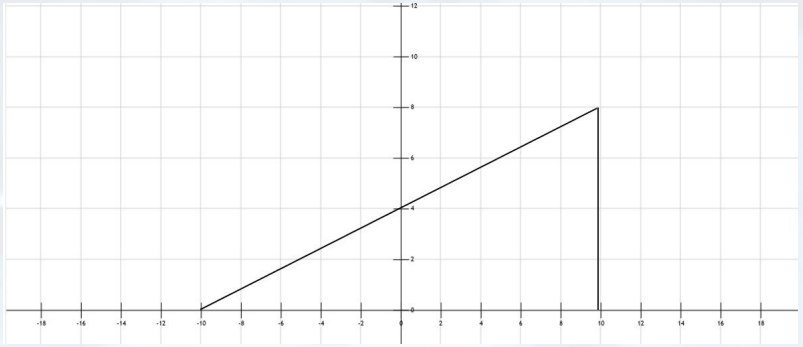
# Structure


y
10
8
6
4
2
0
x
-18
-16
-14
-12
-10
-8
-6
-4
-2
0
2
4
6
8
10
12
14
16
18
180 degrees of
rotation possible

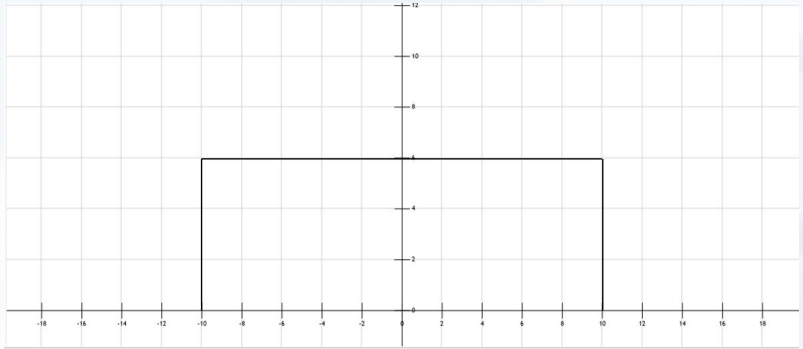
# Hit Methods



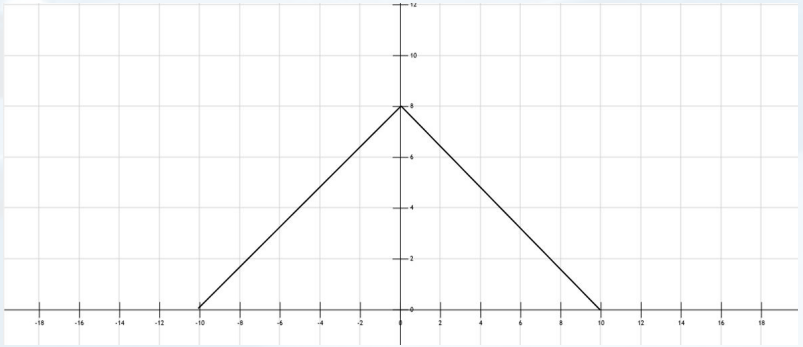
Quadratic



Triangle 1



Uniform

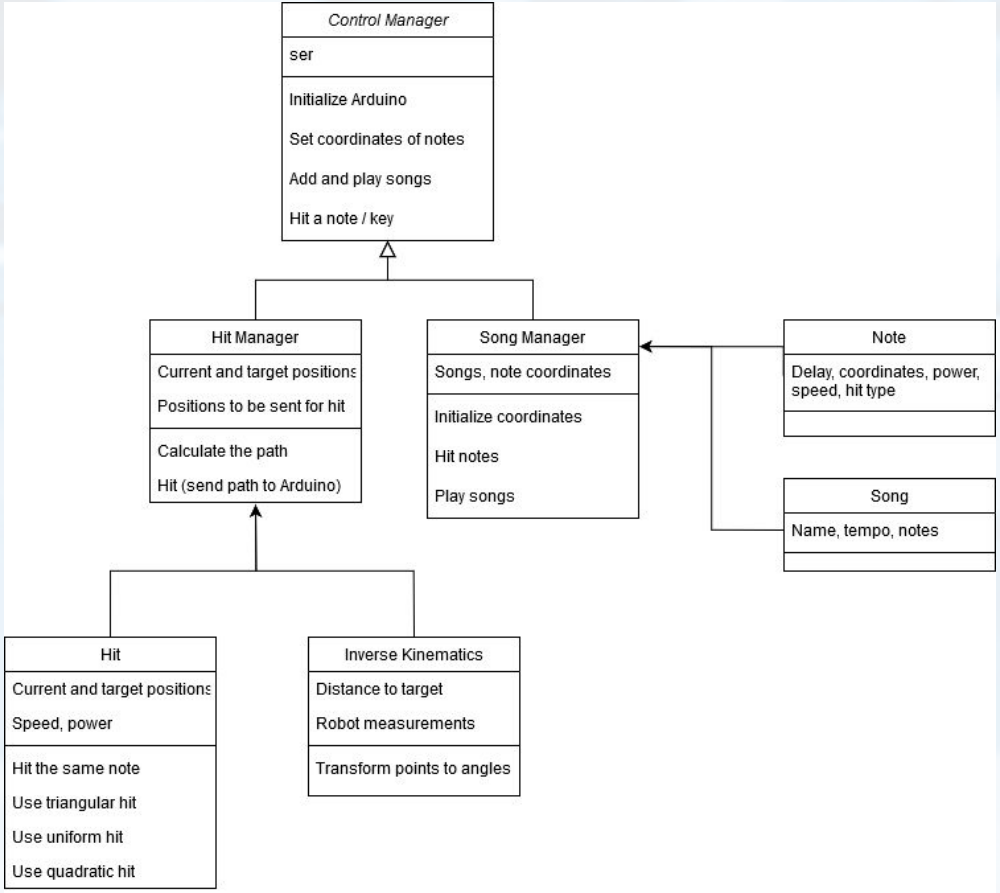


Triangle 2

# Velocity & Power Control

- When performing a hit, consider:
  - Song tempo
  - Distance between notes
  - Note dynamics (‘pp’, ‘p’, ‘mp’, ‘mf’, ‘f’, ‘ff’)
- Speed:
  - Amount of movements between origin and target
  - One movement roughly equal to distance between two neighbour keys
  - Tempo affects the amount of movement between origin and target
- Power:
  - Added integer to the standard height
  - Bounce effect - modifies the standard height

# Class design



# Experiments

Triangle 1 and Triangle 2

- The same 20 note sequence
- Compare accuracies
- Higher tempo variable means more waypoints -> slower
- Tempo 0 -> 3 waypoints
- Tempo 1 -> 4 waypoints

# Experiments


One-sided
In favor of Triangle 2
p = 0.0014
α = 0.25

# Experiments

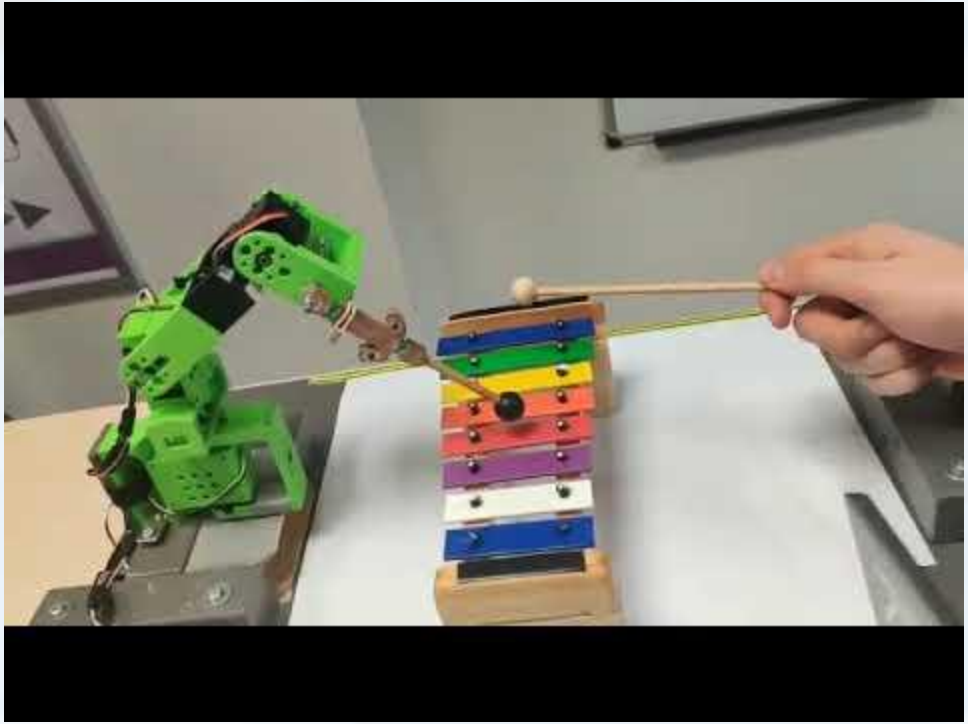
Triangle 2

- Tempo and delay
- For the best found method, we researched tempos and delays

# Experiments


two-sided
Tempo 2 & 6
$p = 0.44$
$\alpha = 0.5$

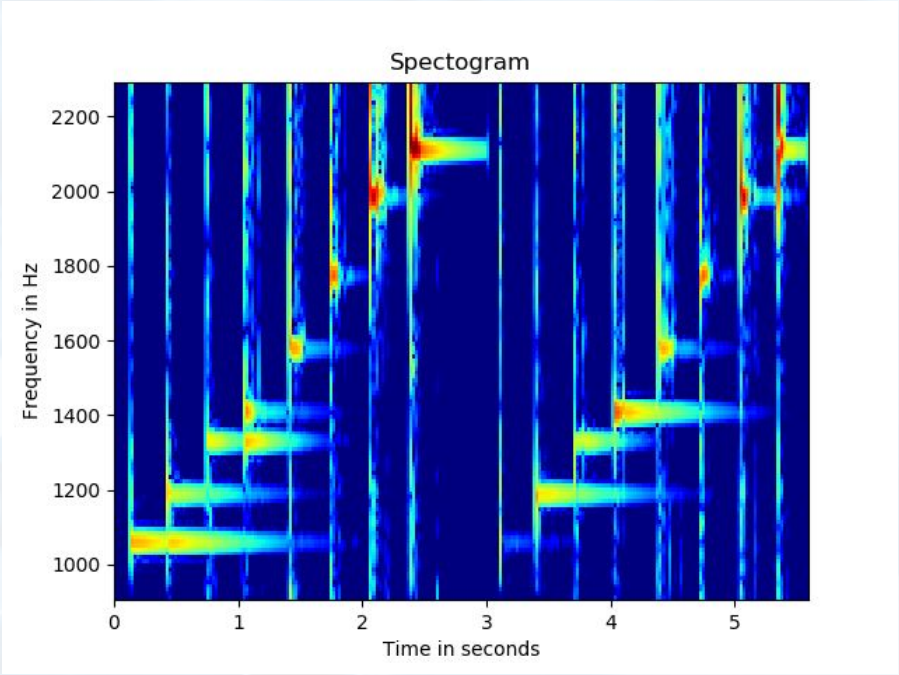
# Demo



https://youtu.be/V62S-WgCPqY

# Sound Processing

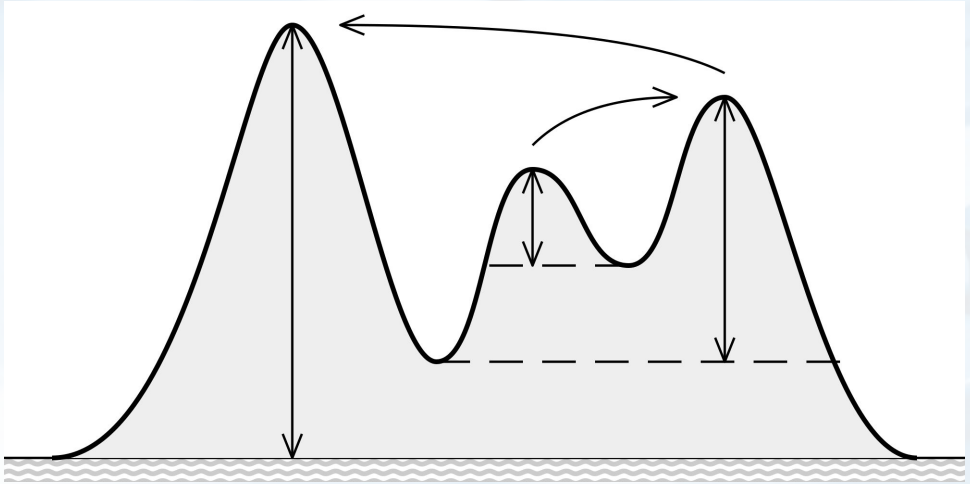
- Identify keys played on xylophone
- Use the concept of short-time Fourier transform[1] to utilize fast Fourier transform (FFT) to calculate discrete Fourier transform (DFT)[2] over time on recorded hits for performing power spectrum analysis
- Returns ‘frequency bins’ that ‘collect’ the energy from a range of frequencies
- Midpoint of bin is frequency identified
- If this is in frequency range for key then that key identified



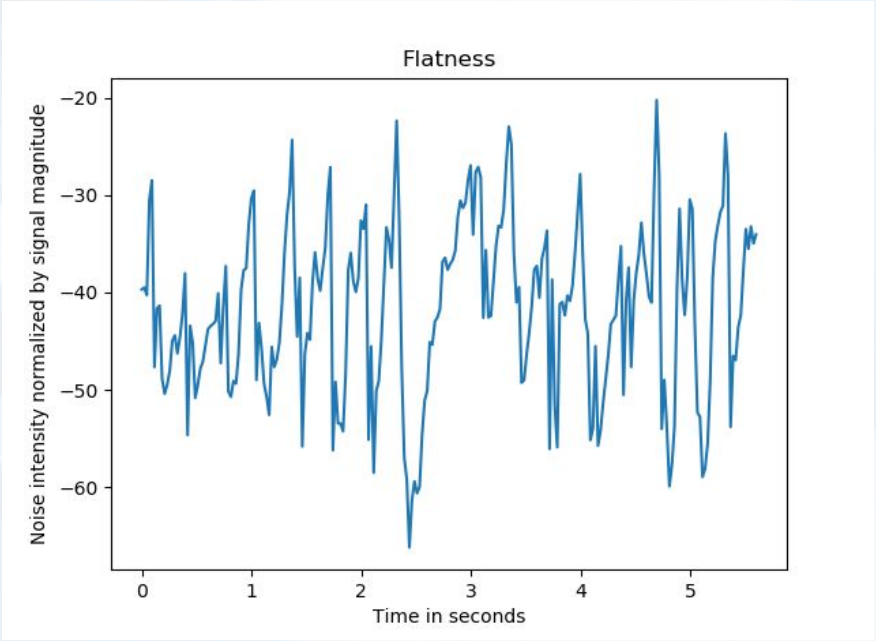

1. Smith III, Juliu O. Spectral audio sound processing. W3K publishing, 2011.
2. Cochran, William T., et al. "What is the fast Fourier transform?." Proceedings of the IEEE 55.10 (1967): 1664-1674.

# Sound Processing - Hit Detection

- Use concept of topographic prominence to detect peaks



- Calculate spectral quality e.g. randomness or stochasticity using librosa[1] that exists in a sound[2]




1. https://librosa.github.io/librosa/generated/librosa.feature.spectral_flatness.html
2. Dubnov, Shlomo. "Generalization of spectral flatness measure for non-gaussian linear processes." IEEE Sound Processing Letters 11.8 (2004): 698-701.

# Sound Processing - Experiments

- To minimize the amount of misidentified keys we try to optimize parameters, STFFT Window Size and Hit Detection Sensitivity
- Hit 8 keys of xylophone slow and fast and from from low to high and high to low
- Repeat 5 times for each combination, so 20 samples in total
- STFFT Window Size values : 128, 256, 512, 1024, 2048
- Hit Detection Sensitivity: 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1

# Sound Processing - Results

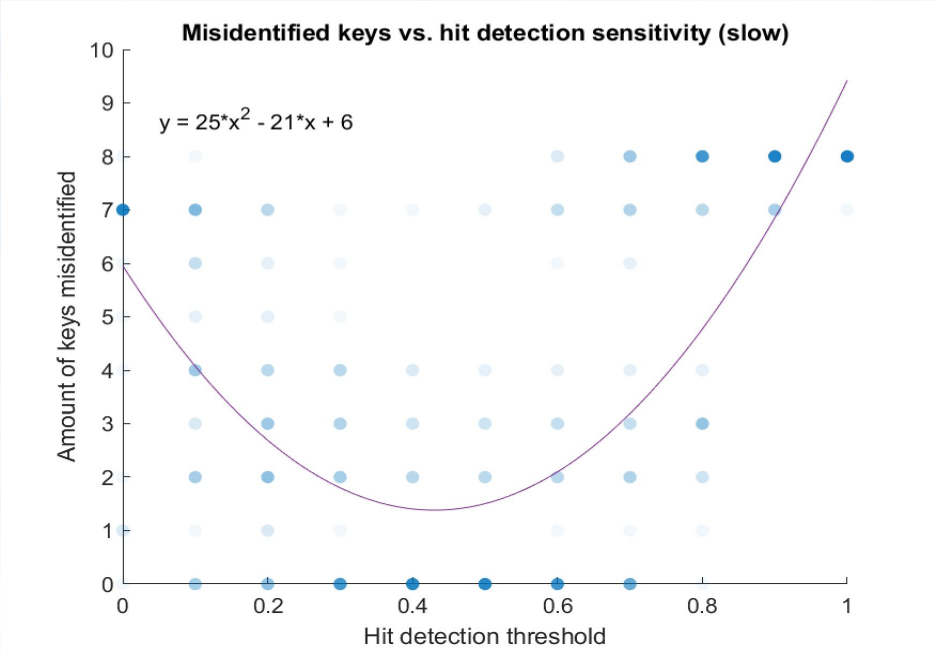
Summary of results over all combinations of parameters

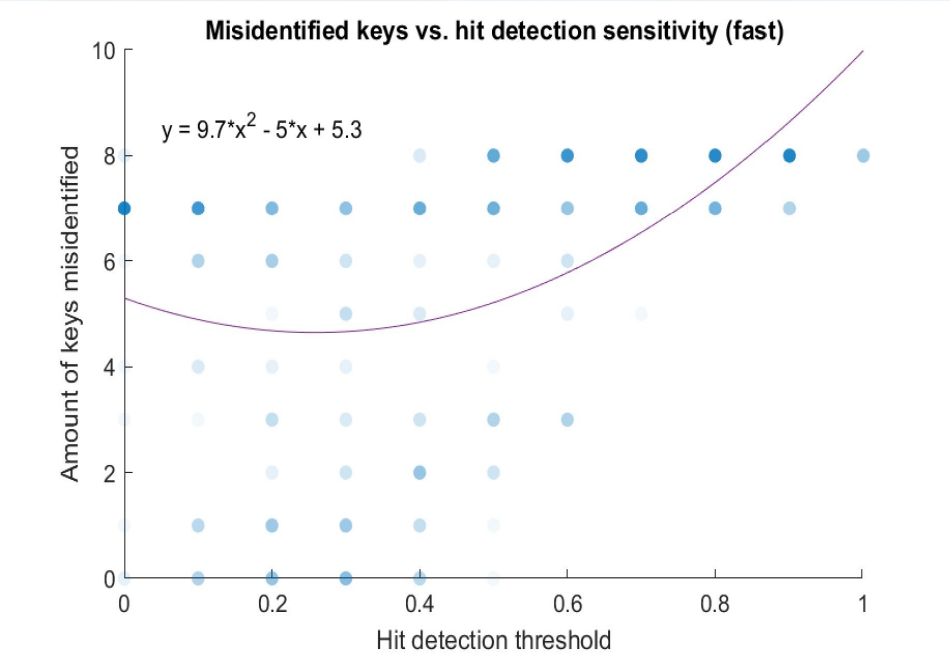
| | Slow (Mean) | Slow (SD) | Fast (Mean) | Fast (SD) | Two-sample t-test |
|---|---|---|---|---|---|
| Misidentified keys: | 3.958 | 3.284 | 5.855 | 2.6681 | p <= 0.05 |
| Flatness: | 0.01842 | 0.006820 | 0.002215 | 0.001006 | p <= 0.05 |
| Effective Duration: | 15.5005 | 6.4866 | 1.4234 | 1.0017 | p <= 0.05 |

Misidentified keys for STFFT Window Size = 2048 and Hit Detection Sensitivity = 0.34, significant improvement compared to average over all window sizes and sensitivities

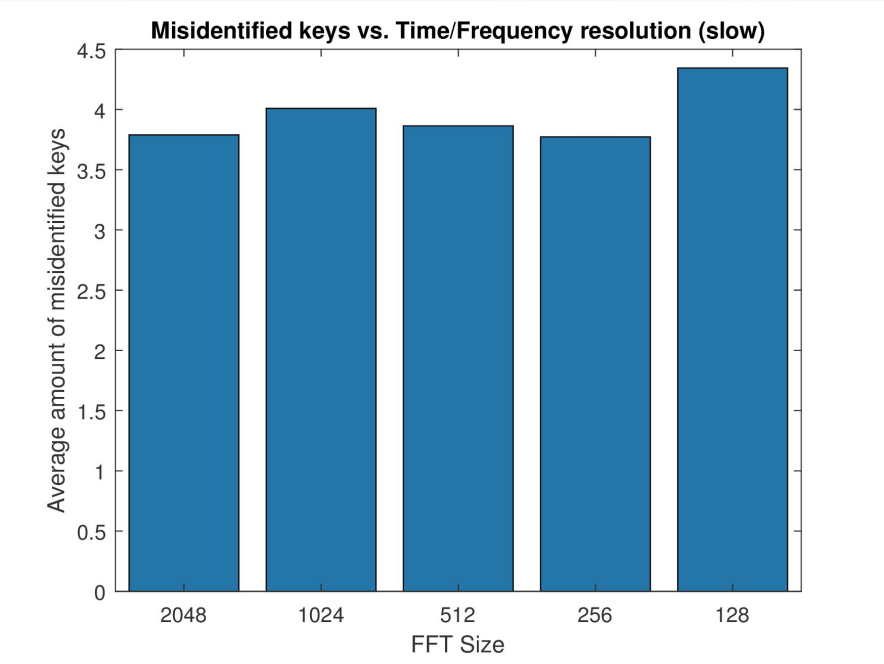
| | Slow (Mean) | Slow (SD) | Fast (Mean) | Fast (SD) | Two-sample t-test |
|---|---|---|---|---|---|
| Misidentified keys: | 0.2 | 0.6325 | 3 | 2.1269 | p <= 0.05 |

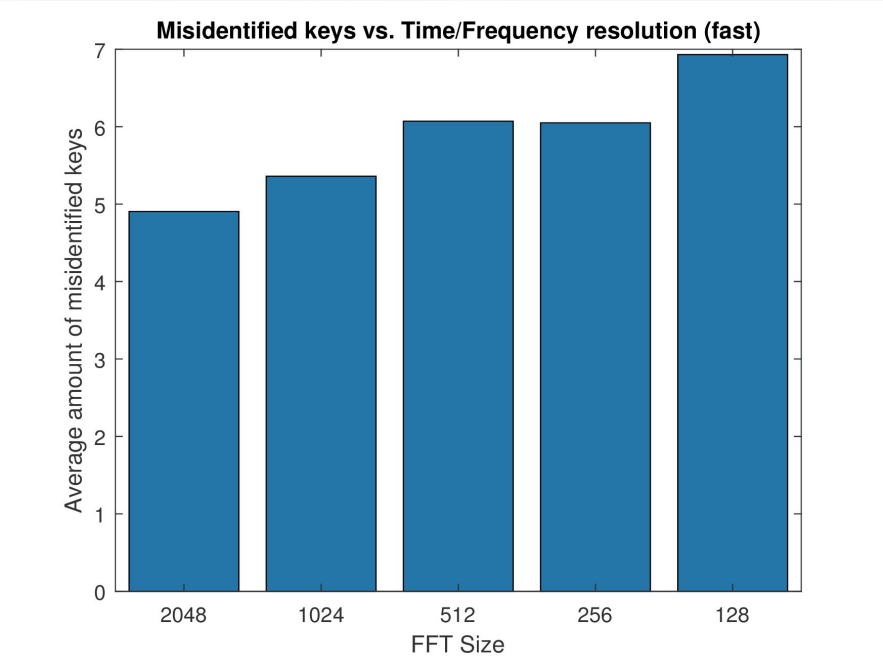
# Sound Processing - Results



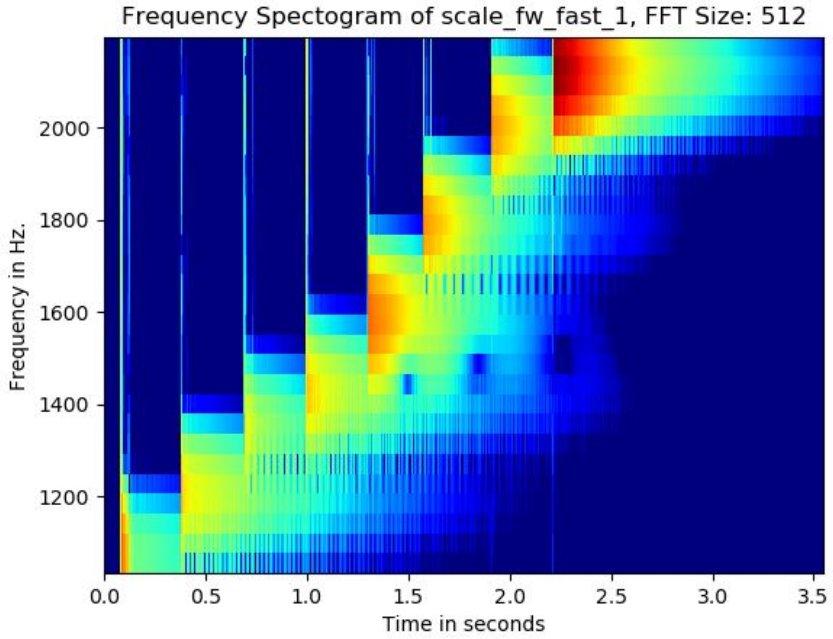



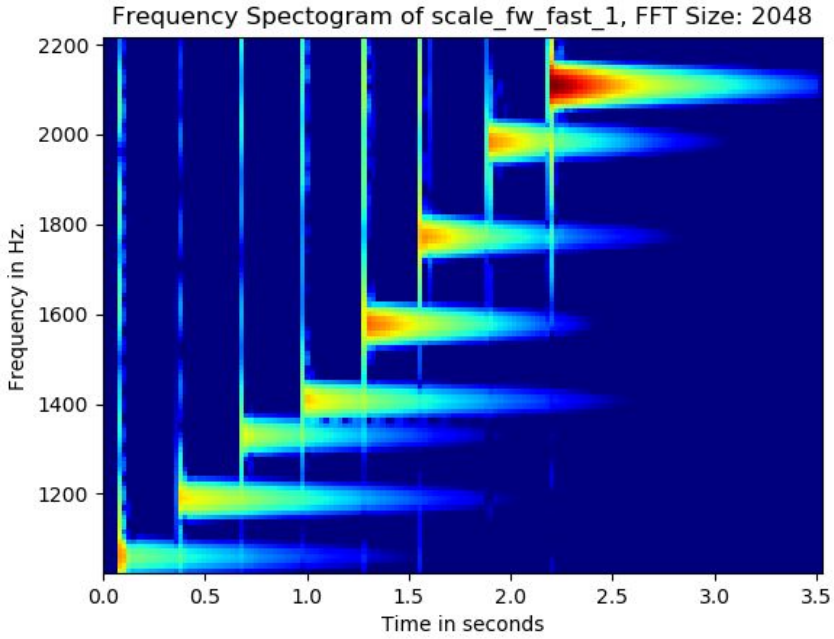
# Sound Processing - Results



# Sound Processing - Results



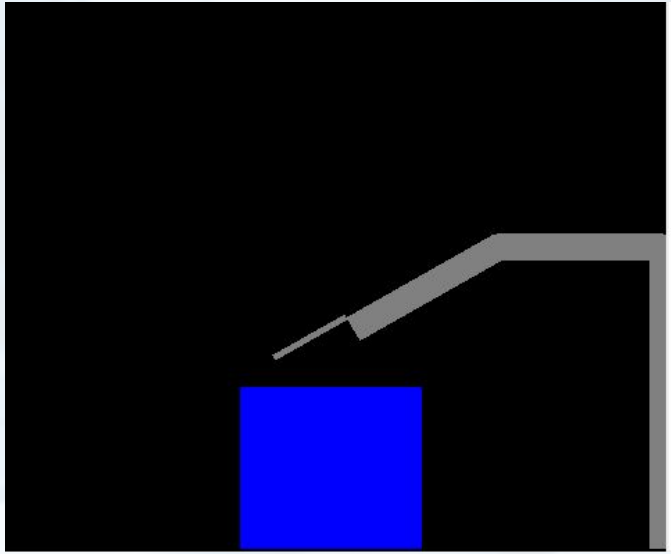

| STFFT Size: | 128 | 256 | 512 | 1024 | 2048 |
|---|---|---|---|---|---|
| Time Resolution (in milliseconds): | 1.4395496125 | 2.879099225 | 5.75819845 | 11.59636325 | 23.15275627 |

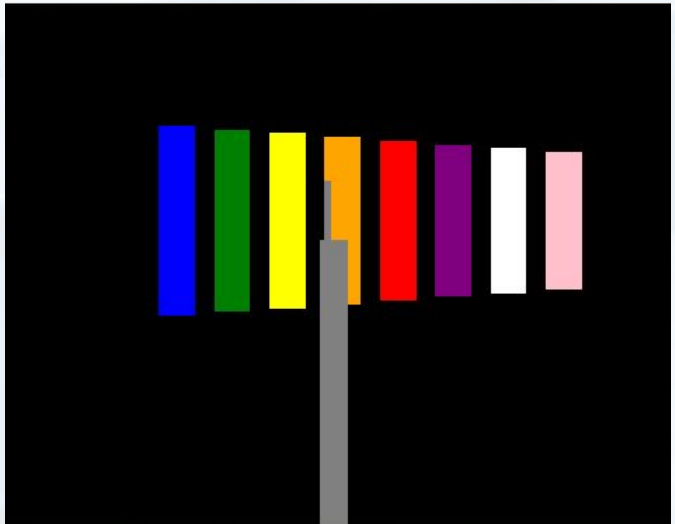
# Sound Processing - Comments

- Explore what sound processing identifies when misidentifying
- How robust against noise (introduce white noise)
- Look at flatness level when identifying and wait until it decreases within time period until identifying
- Use optimization procedure to find best parameters instead of trying all combinations of a set group of options

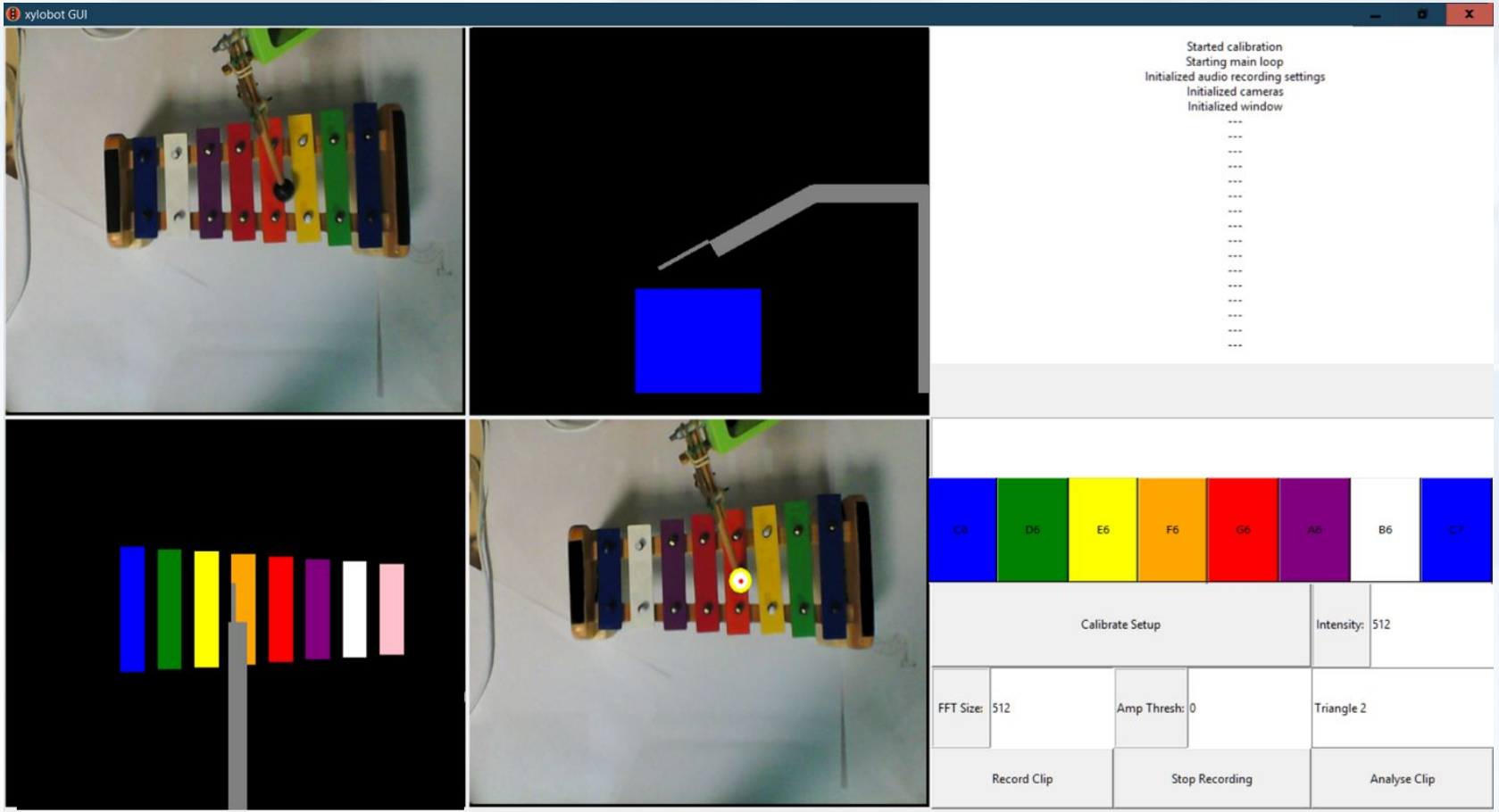
# Simulation

- Takes values from computer vision and control
- Based on the calibration, the xylophone is put in the right position in the simulation.
- Updates the movement of the simulation-robot to the angle sent to the physical one.





# GUI Showing Simulation and Calibration



# Calibration through Computer Vision

Identifying the mid points of the keys

- Colour detection improvements

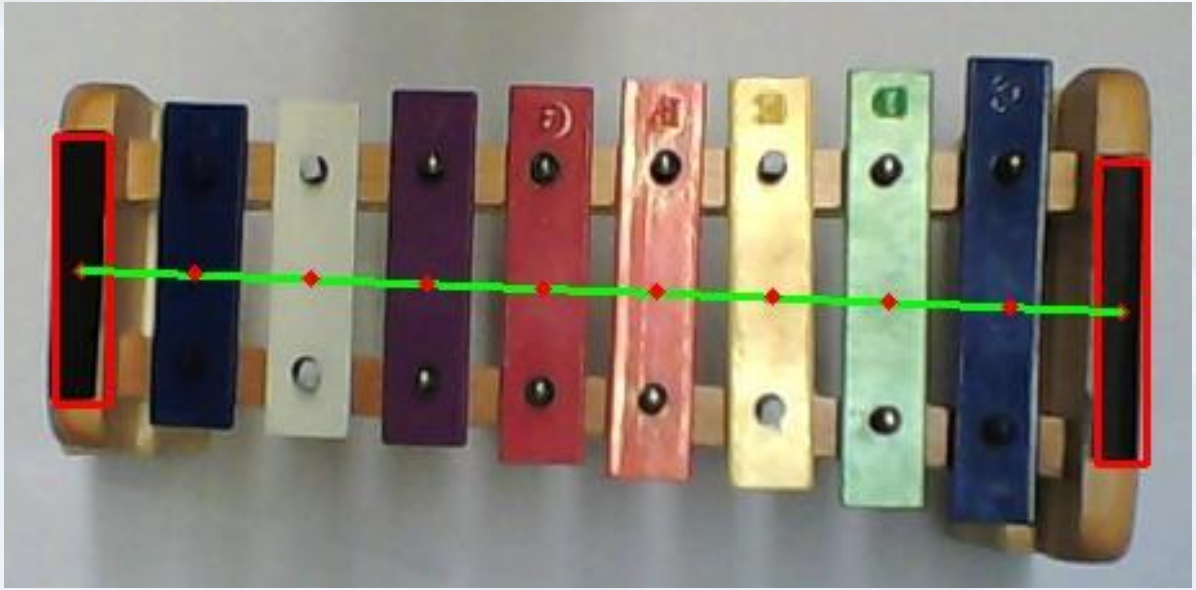
Identifying the Wand

- Colour detection

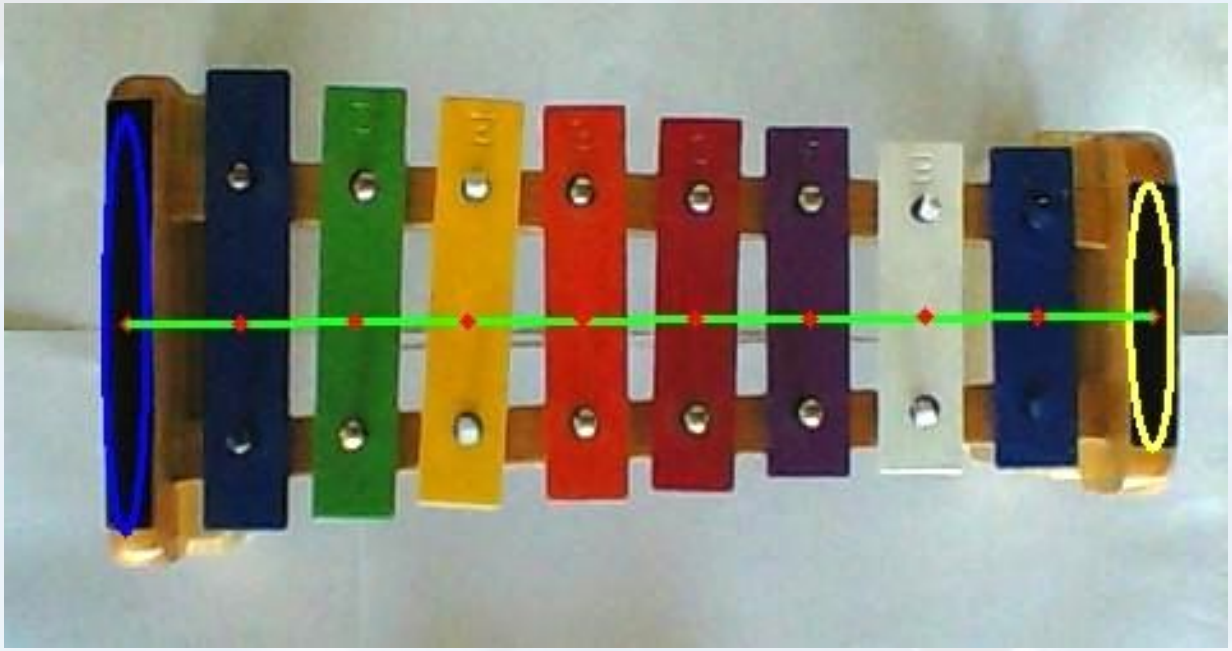
Key Calibration

- Closing the Loop

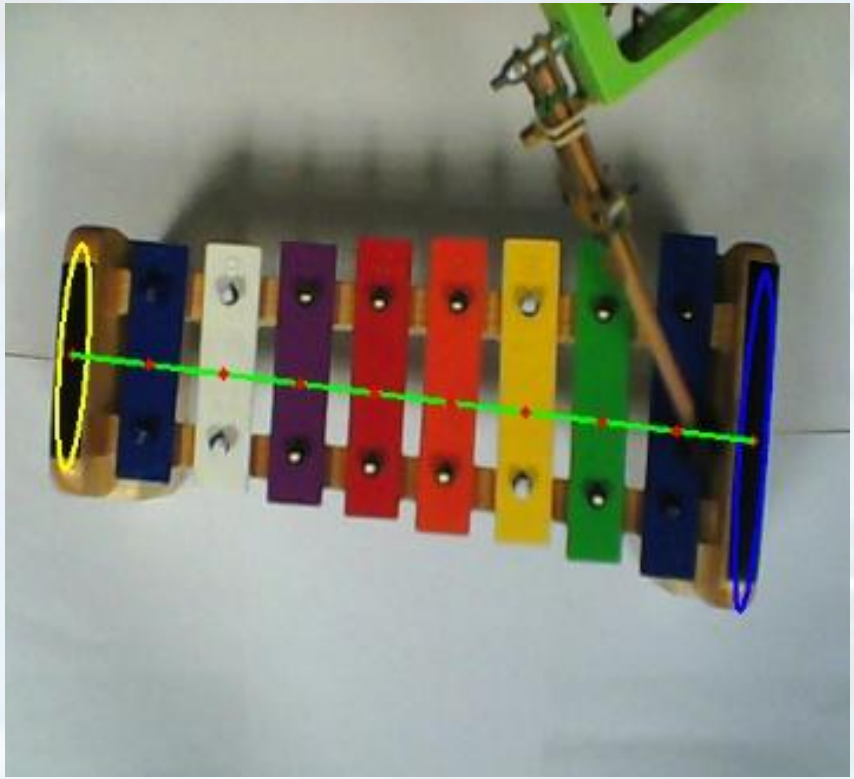
# Midpoint Detection - Colour detection with Rectangles



# Midpoint Detection - Colour detection with Ellipses



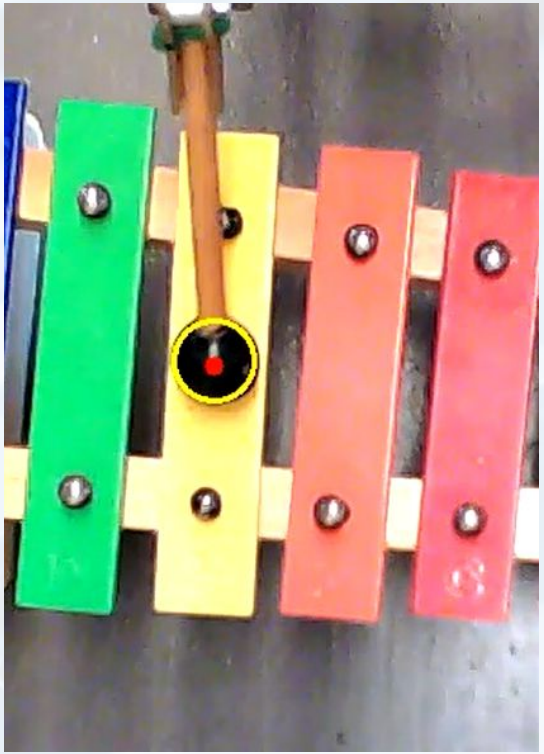
# Midpoint Detection - Colour detection with Rotated Ellipses



# Wand Detection - Colour Detection

- Colour detection with restrictions on area and location was implemented to find the mallet head while reducing false identifications.

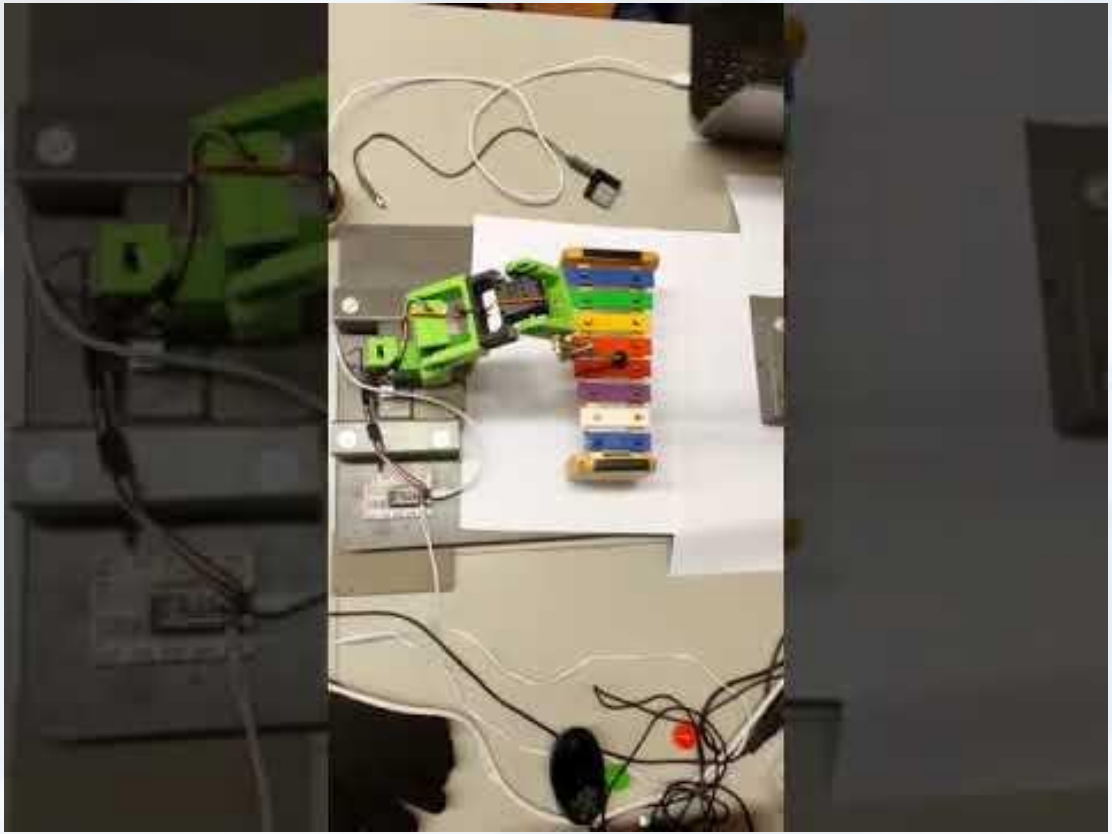
- Wand colour detection was used to calibrate keys since there is not much motion from frame to frame. The detection finds the mallet head and returns the centerpoint value.



# Closing the Loop - Key Calibration with CV

1. The midpoints of the keys are detected with our key detection algorithm
2. The location of the wand head is detected and compared to the midpoints of the keys
3. The difference between the two points is found and the robot moves the distance of the offset until the wand head aligns to the midpoint.
4. This repeats for every key to calibrate the location of the keys.

# Closing the Loop - Key Calibration with CV



https://www.youtube.com/watch?v=iPZVsasncok&feature=emb_logo

# AI: Improvisation

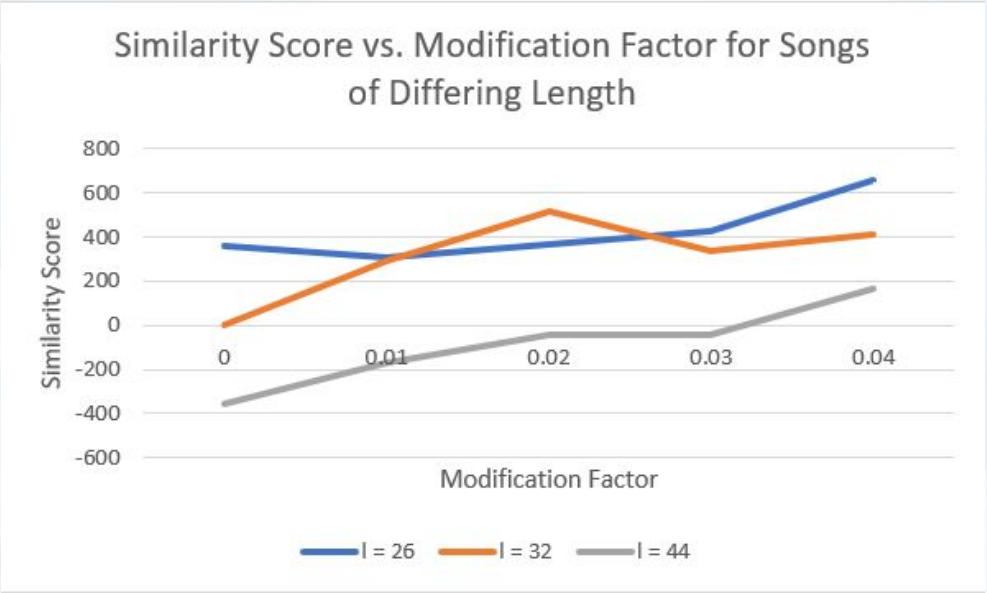
- Improvisation on a theme is possible with a Markov model.
  - Note Generation
    - Transition matrix represents probabilities of transitions between notes
    - This matrix is build based on a melodic theme played by a human user
    - Laplace estimates are used to ensure non-zero probabilities
  - Timing Generation
    - Discretizes input melody by quantizing based on the smallest interval between two notes
    - This causes all time intervals between notes to be a multiple of this interval
    - This means that the frequency of each time interval can be counted
    - A probability of a time interval between two notes occuring in the improvised section is assigned based on these frequencies

# AI: Pros and cons

- Advantages
  - Allows creative improvisation due to non-zero transition probabilities
  - Improvised sequence follows theme of input melody
  - Allows the adoption of complex rhythmic patterns
  - After the transition matrix is generated, there is no heavy computing necessary
- Disadvantages
  - Quantization only works if the smallest interval is not to large and the player providing the input melody has at least some sense of rhythm
  - Requires a long sequence to create a good transition matrix

# AI: Experimenting with how much creativity to allow

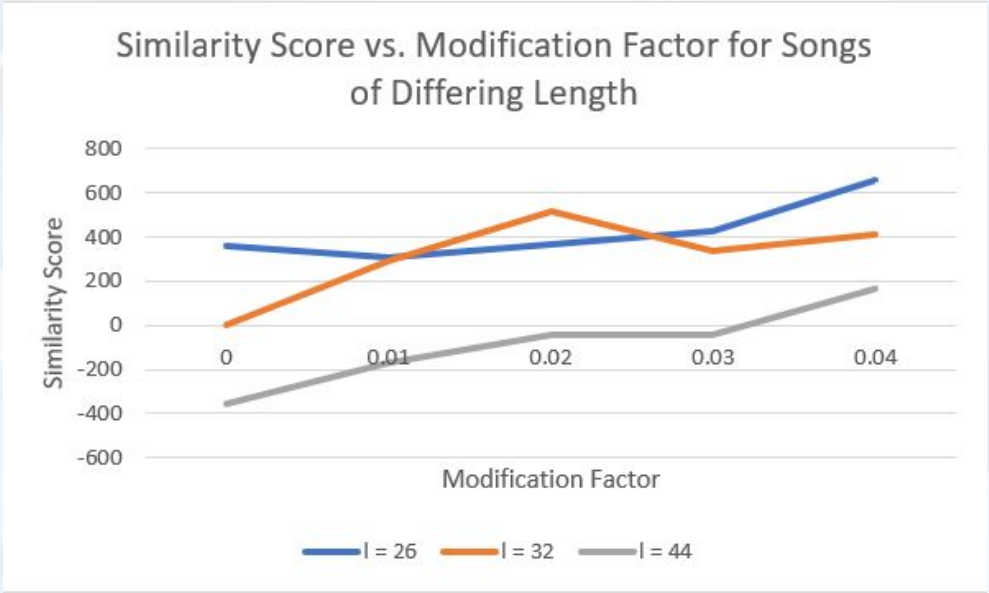
- Transitions with a low probability are raised and the effect on how similar the improvised melody is to the original melody is measured
- Similarity is based on a scoring system which looks at distances between notes and whether notes in the base triad




https://pdfs.semanticscholar.org/54ae/c8e0a48d17d32cbfe1e90eeb0f0e484f0be1.pdf

# AI: Experimental results

- As less likely transitions are made more probable, the less alike the input and improvised melodies are.
- The more notes in the input melody, the more similarity there is between the input and improvised melodies
- Correlation coefficients are 0.82, 0.70, and 0.96



# Conclusion

We have found that the triangle 2 method works best for hitting the keys.

We were able to optimize the sound detection to the optimal sample length and threshold.

The AI used can compose a piece of music based on a melody played by a human.


Thank you

# Who did what?

- Control: Rik & Valentin
- Sound Processing & GUI: Kailhan
- Simulation: Benjamin & Cas
- AI: Benjamin
- Computer Vision: Ewan & Rhys

# More Sources


- https://www.news.gatech.edu/2019/11/05/national-labs-georgia-tech-collaborate-ai-research
- https://dke.maastrichtuniversity.nl/gm.schoenmakers/wp-content/uploads/2015/09/Linskens-Final-Draft.pdf
- https://qz.com/689827/moogfest-shimon-music-robot/
- researchgate.net/publication/221786626_Robotic_Musicianship_-_Musical_Interactions_Between_Humans_and_Machines
- https://pdfs.semanticscholar.org/54ae/c8e0a48d17d32cbfe1e90eeb0f0e484f0be1.pdf
- https://interactiveaudiolab.github.io/assets/papers/Rafii-Pardo%20-%20Music-Voice%20Separation%20using%20the%20Similarity%20Matrix%20-%20ISMIR%202012.pdf
- https://www.mediacollege.com/glossary/q/quantization.html
- https://pdfs.semanticscholar.org/54ae/c8e0a48d17d32cbfe1e90eeb0f0e484f0be1.pdf