NAO, Let’s play the Xylophone

24/25WS, Humanoid Robotic Systems – Final Project Presentation

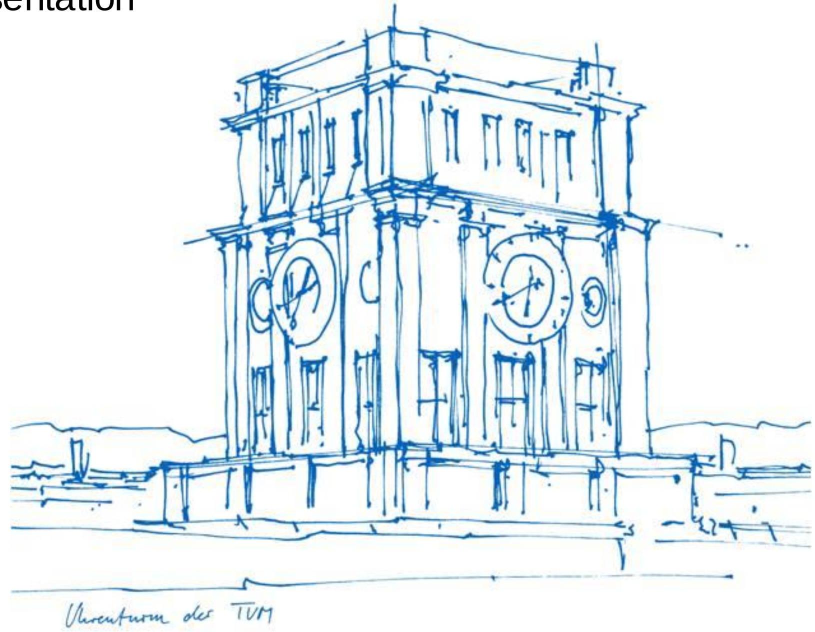
Team C: Zhiyu Wang, Yijia Qian, Yuan Cao

Chair for Cognitive Systems

Prof. Dr. Gordon Cheng

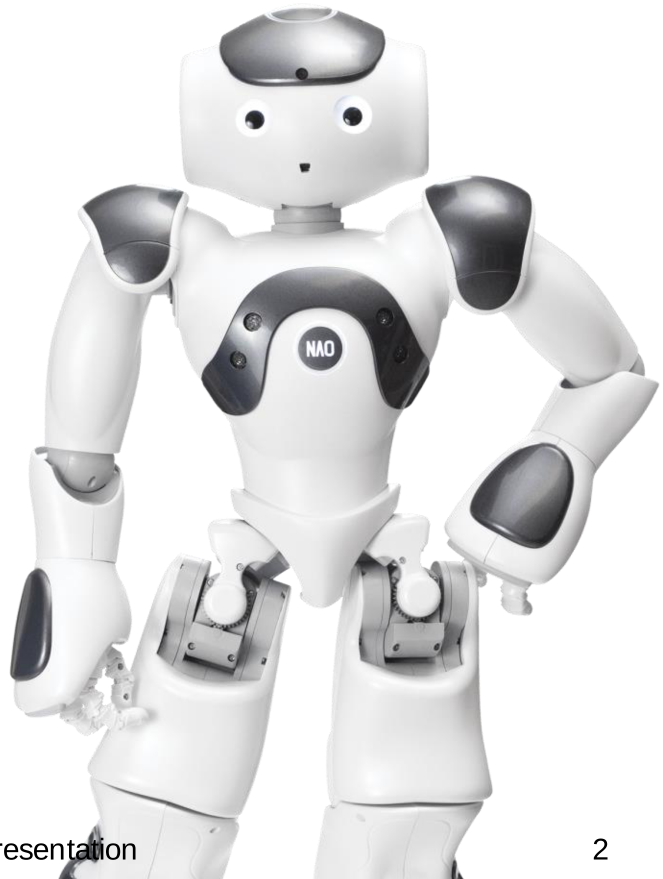
Technical University of Munich

Munich, 3 February 2025



Chair for Cognitive Systems
Department of Electrical and Computer Engineering
Technical University of Munich

Overview



Chair for Cognitive Systems
Department of Electrical and Computer Engineering
Technical University of Munich

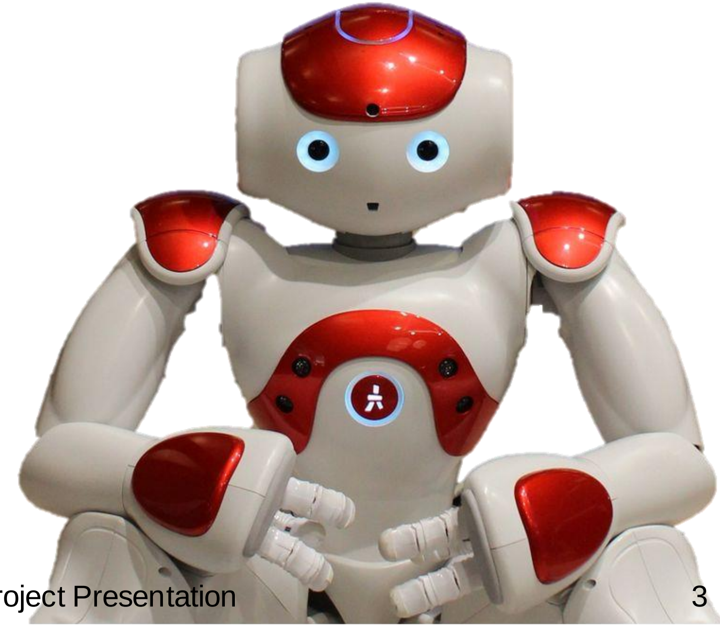
Introduction: Project Description and Division of Tasks

Project Description:

Enable NAO to listen to a melody, extract the notes, and reproduce the song on a xylophone by grasping the stick and replicating the timing and sequence.

Division of Tasks:

Zhiyu Wang	Overall Architecture, User Interface, Note Detection
Yijia Qian	Key-coordinate / Hand-strike Positions Mapping to Playing
Yuan Cao	Grasping Algorithm, ArUco Marker Algorithm Optimization



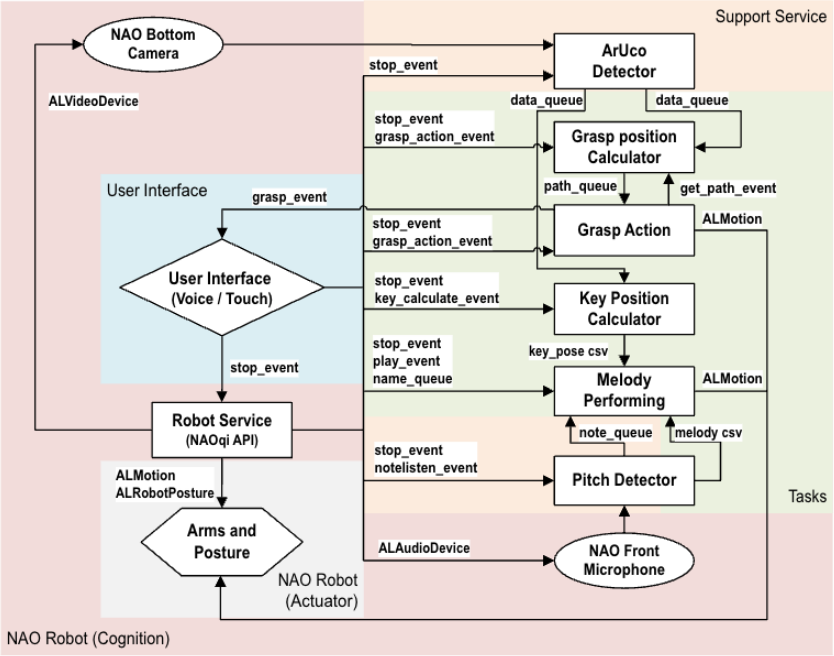
Chair for Cognitive Systems
Department of Electrical and Computer Engineering
Technical University of Munich

Introduction: System Architecture

Modular task execution:
where tasks are triggered on demand but executed dependently.

Event-driven interaction flow:
enabling efficient task coordination.

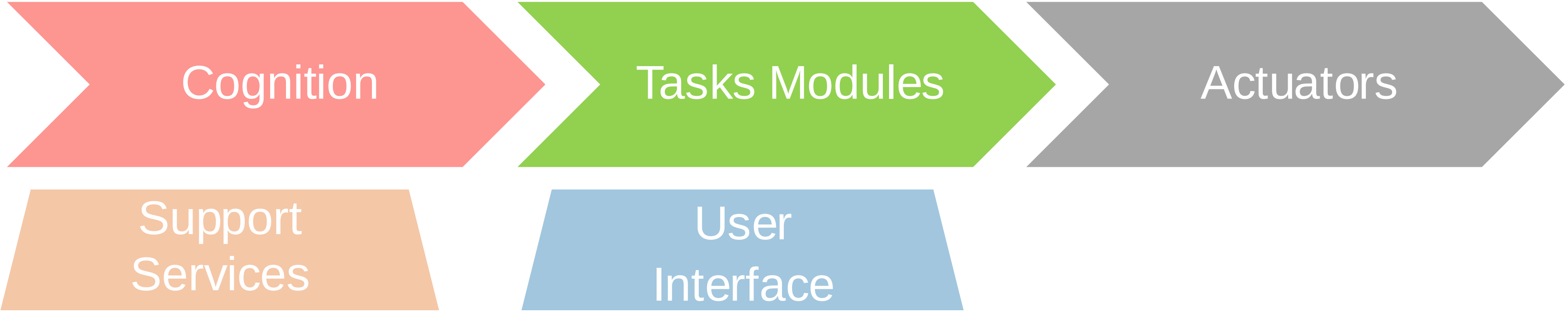
Hierarchical task structure:
where different subtasks are executed sequentially to achieve the overall goal.



Chair for Cognitive Systems
Department of Electrical and Computer Engineering
Technical University of Munich

Introduction: System Architecture

The system is a **Task-Based architecture**, characterized by the following features:



Support Modules: User Interface

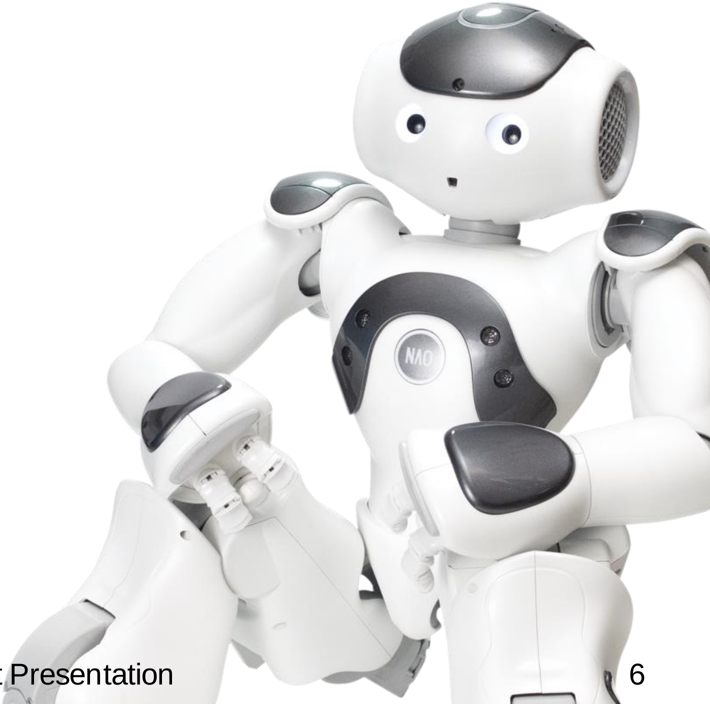
In a nutshell, User interface is NAO.

Touch

- Robot touch sensors enable interaction;
- Each touch sensors are assigned to different tasks.

Voice

- speech recognition & TTS
- Vocabulary preset:
  - “play, replay, listen, grasp, etc.



Support Modules: User Interface - Security Mechanisms

Safety First !

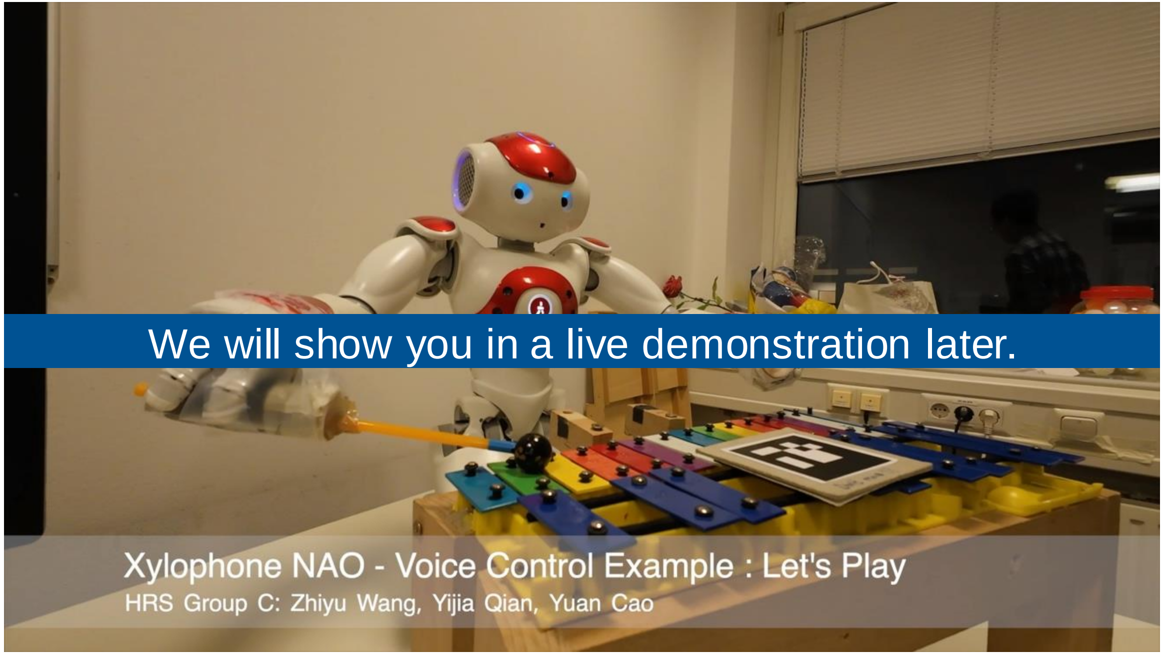
Triggered by **right foot touch** ("RFoot/Bumper/Right")

Sets stop_event to:

- Instantly terminate all tasks
- Halt robot movements
- Prevent hazards

Chair for Cognitive Systems
Department of Electrical and Computer Engineering
Technical University of Munich

Support Modules: User Interface - Voice Control



We will show you in a live demonstration later.

Chair for Cognitive Systems
Department of Electrical and Computer Engineering
Technical University of Munich

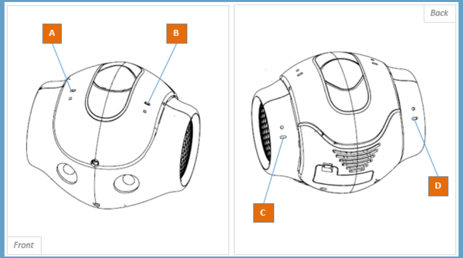
Support Modules: Pitch Detection - Audio Signal Processing

Signal

NAO robot's ALAudioDevice
16,000 Hz Mono Front Mic

RAW Signal Format:
16-bit Little Endian PCM

Target Signal Format:
32-bit Float Digital Audio

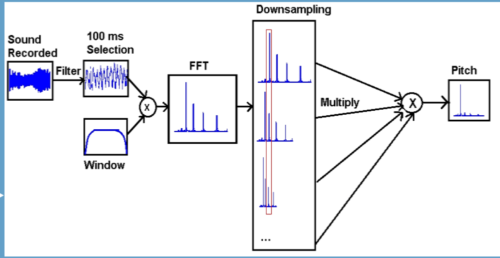


FFT

De-noise the audio data

Separate the spectrum

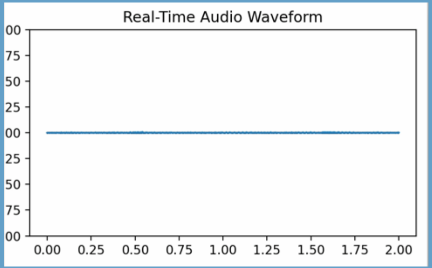
Determine the notes, based on the 12-tone equal temperament and 440hz pitch standard.



Pitch

Duration Quantization

Duration Normalization



Chair for Cognitive Systems
Department of Electrical and Computer Engineering
Technical University of Munich

Support Modules: Pitch Detection - From Sheet to CSV

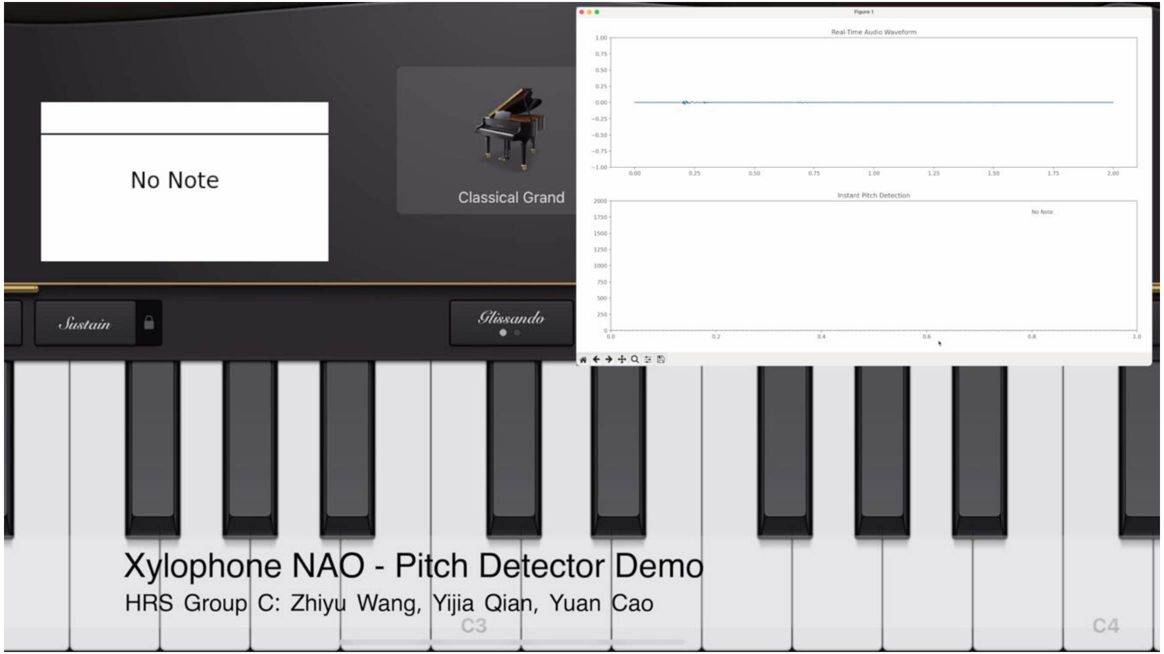
A similar method has been used to directly translate the music scores.
So that the robot can read and operate them directly.

- Note (reproject to Xylophone range X4-X7)
- Duration Quantization
- Duration Normalization

note	lasting_time
E7	1
E7	1
F7	1
G7	1
G7	1
F7	1
E7	1
D7	1
C7	1
C7	1
D7	1
E7	1
E7	1.5
D7	0.5
D7	2

Chair for Cognitive Systems
Department of Electrical and Computer Engineering
Technical University of Munich

Support Modules: Pitch Detection - Demonstration



Support Modules: Pitch Detection - Result Output

The processed results will be output as:

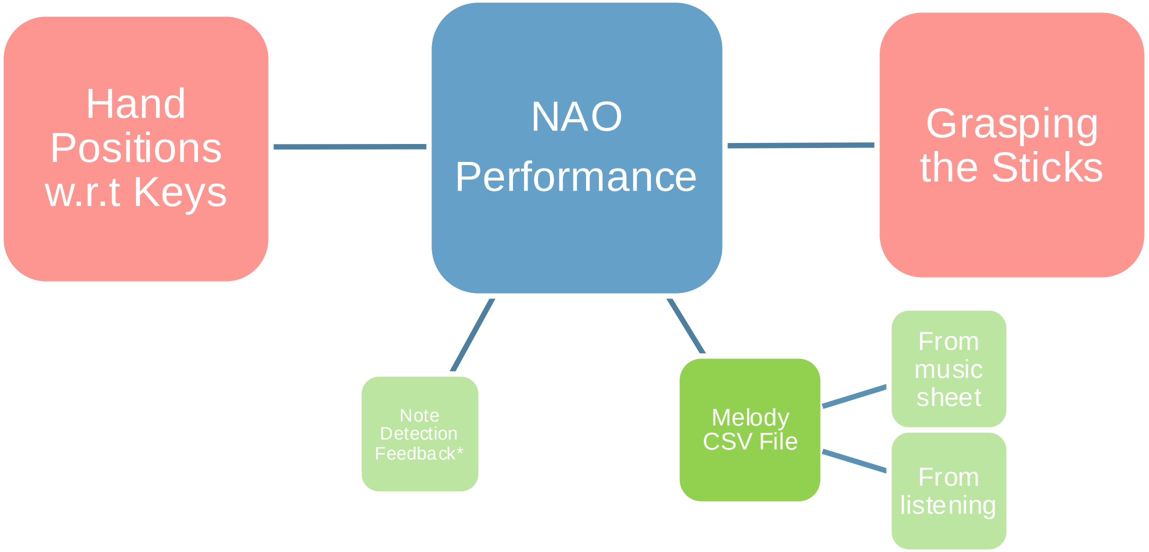
Pitch Detector

replay.csv
notes & lasting_time
on demand

note_queue
note detected
real-time*

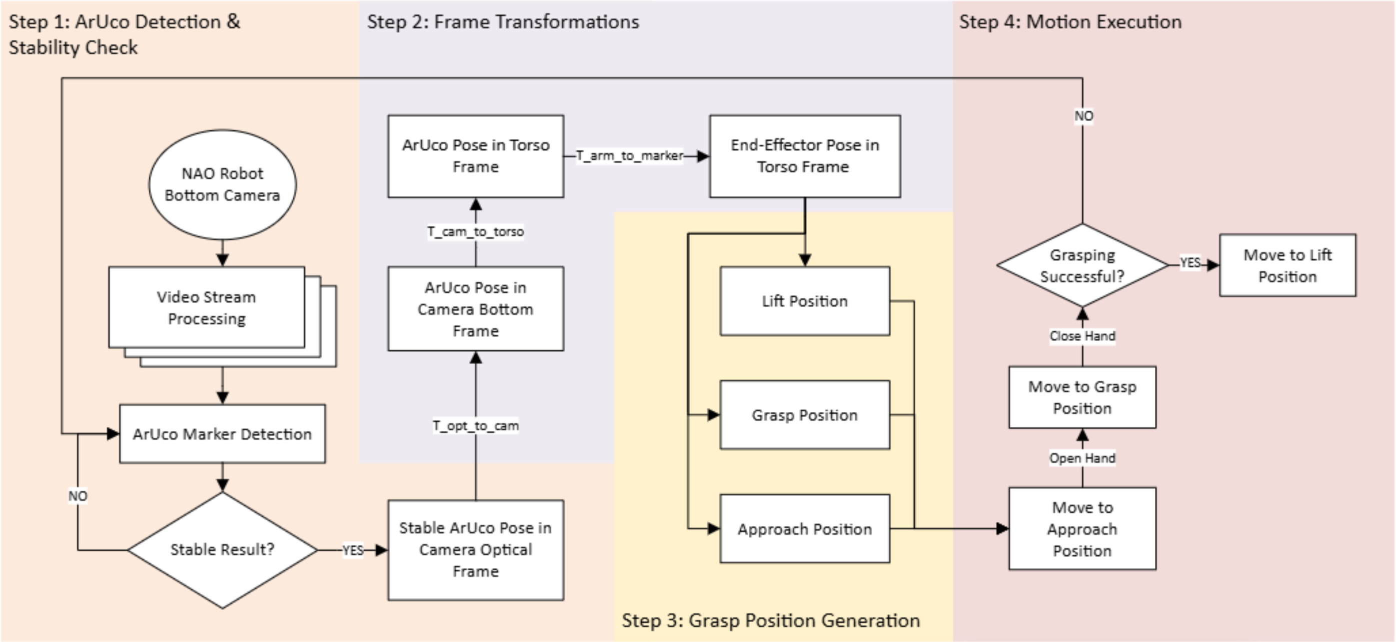
note	lasting_time
C5	4
E5	2
G5	2
B4	2
C5	0.5
D5	0.5
C5	4
A5	4
G5	2
C6	2
G5	2
F5	1
E5	4

Main Tasks:

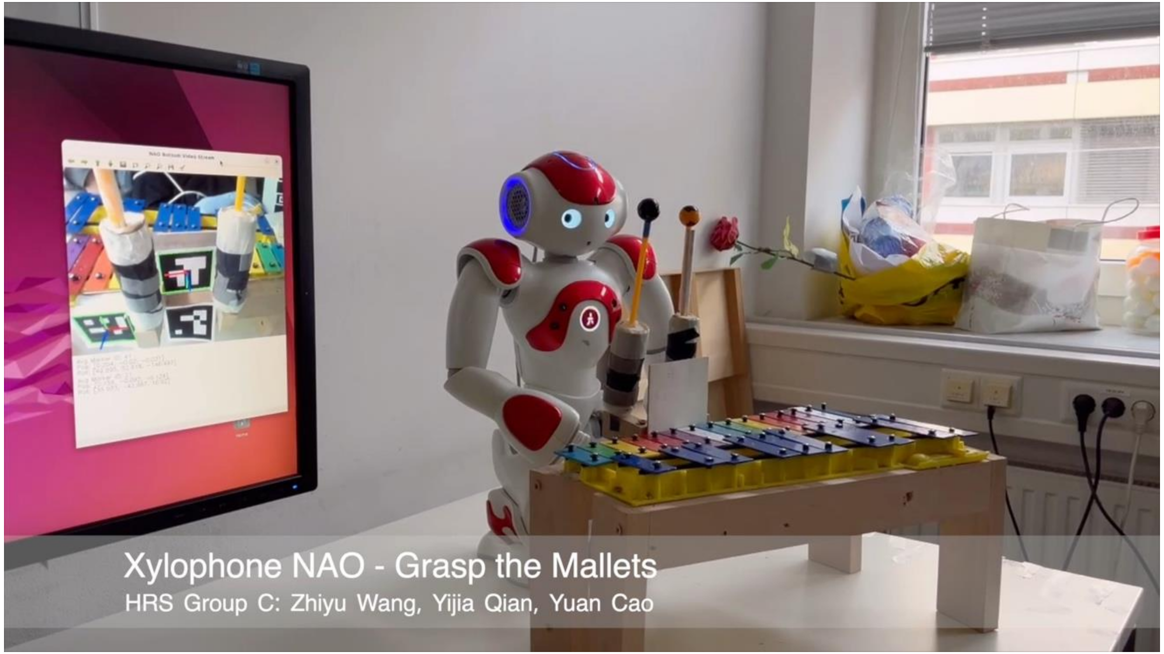

Hand Positions w.r.t Keys
NAO Performance
Grasping the Sticks
Note Detection Feedback*
Melody CSV File
From music sheet
From listening

Chair for Cognitive Systems
Department of Electrical and Computer Engineering
Technical University of Munich

Main Tasks: Grasping the Sticks – Structure Overview

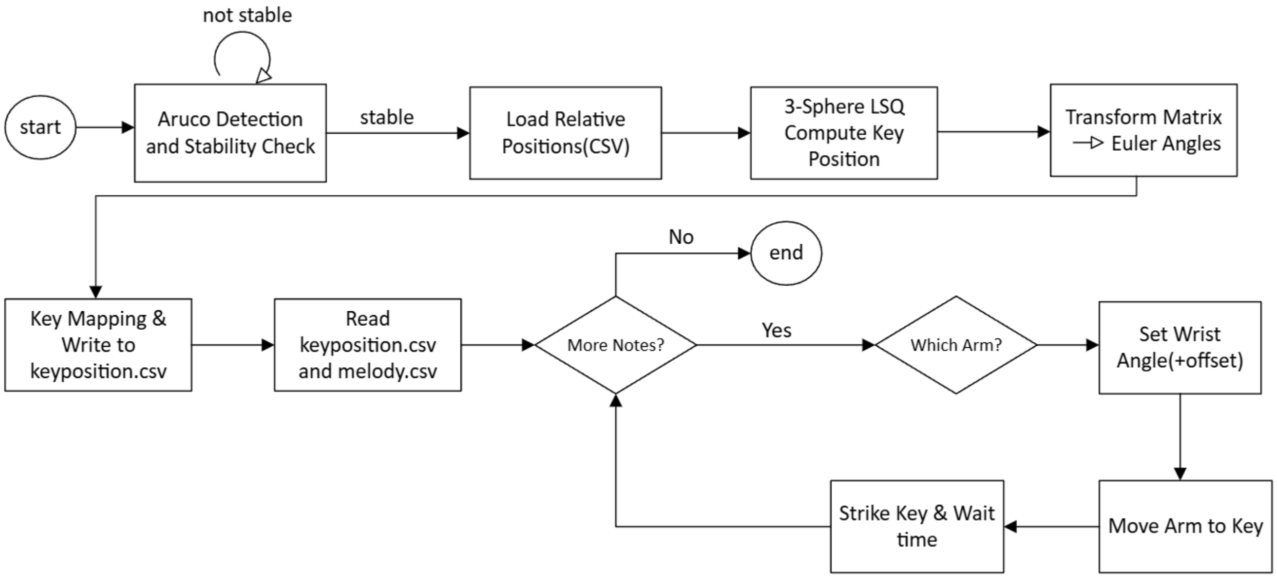


Main Tasks: Grasping the Sticks



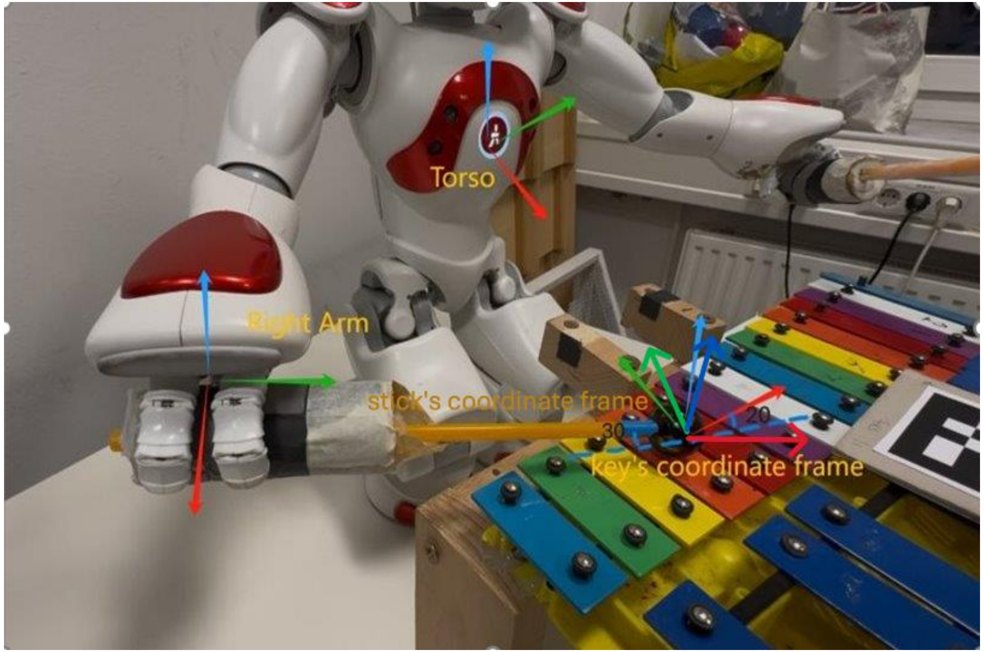
Chair for Cognitive Systems
Department of Electrical and Computer Engineering
Technical University of Munich

Main Tasks: Playing Notes on Xylophone



Chair for Cognitive Systems
Department of Electrical and Computer Engineering
Technical University of Munich

Main Tasks: Playing Notes on Xylophone



The total rotation matrix is expressed as:

$$R_{\text{total}} = R_{z1} \cdot R_{z2} \cdot R_y \cdot R_{z3}$$

$$\begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix} = R_{z1} \cdot R_{z2} \cdot R_y \cdot \text{stick\_direction} + \begin{bmatrix} x_e \\ y_e \\ z_e \end{bmatrix}$$

where:

- $\text{stick\_direction} = \begin{bmatrix} -L \\ 0 \\ 0 \end{bmatrix}$ represents the stick's direction in the stick's coordinate frame.
- $\begin{bmatrix} x_e, y_e, z_e \end{bmatrix}$ is the position of the xylophone key in the torso coordinate frame.

Main Tasks: Playing Notes on Xylophone

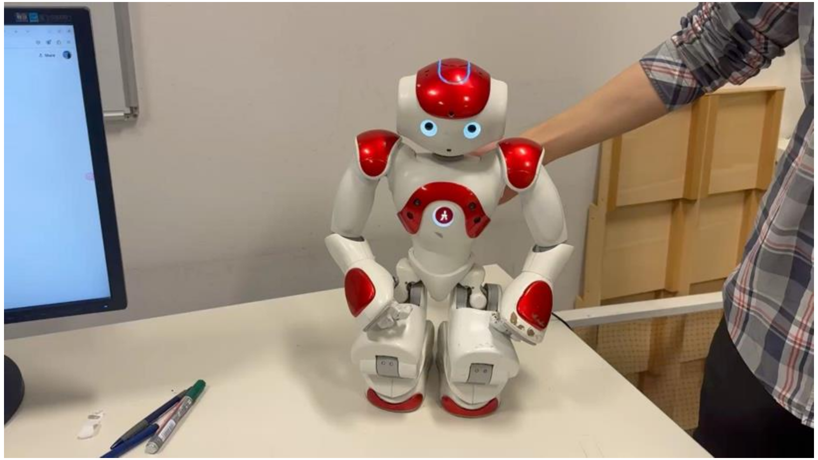
Chair for Cognitive Systems
Department of Electrical and Computer Engineering
Technical University of Munich

Main Tasks: Performance NAO – Play the Melody

We will show you in a live demonstration later.

Remaining Issues

1. Hardware Issues with the NAO Arm：
   - Imprecise Kinematic Model; Limited Joint Accuracy;
   - Lack of Direct End-Effector Feedback;
   - Over-heated; Joint Compliance and Friction...…
2. Performance NAO with Error Correction:
   - Audio Detector Issues;
   - Robot Arm Issues.
3. ArUco Marker Detector:
   - Poor Image Quality of the Camera;
   - Imperfect Optical Lens.



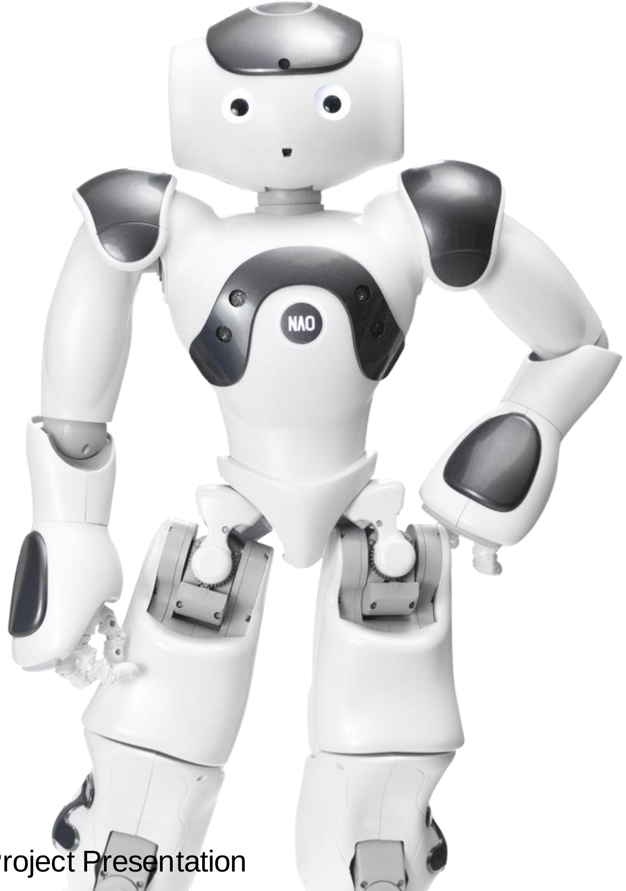
Future Works

We have already seen the potential of humanoid robots.

- Autonomy and intelligence
  - Generative model and reinforcement learning methods
- Expansion of application scenarios
  - Through LfD training
- Collaboration with humans

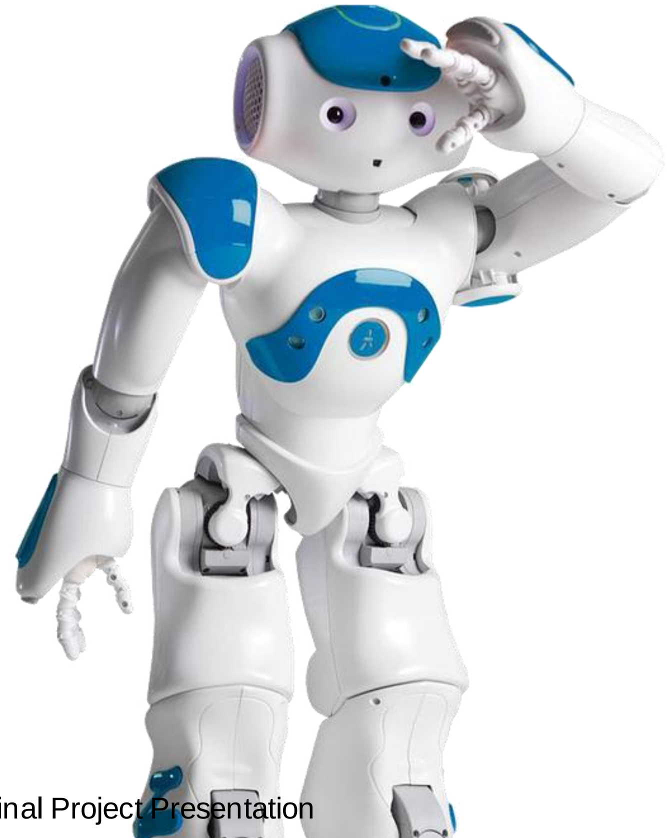
Chair for Cognitive Systems
Department of Electrical and Computer Engineering
Technical University of Munich

Thank you for your attention.



Chair for Cognitive Systems
Department of Electrical and Computer Engineering
Technical University of Munich

Do you have any questions?



Chair for Cognitive Systems
Department of Electrical and Computer Engineering
Technical University of Munich

Appendix ArUco Marker Detection

Connect to Nao's Bottom Camera through ALVideoDeviceProxy::subscribeCamera(), and by comparing the results of various resolutions, finally choose 640P 30fps connection rate, using BGR mode.

In the acquired image, the pose of the ArUco marker in the camera optical frame can be obtained directly using the PnP method in the cv2 library.

Through the transformation, we can get the coordinates of the markers in the torso frame.

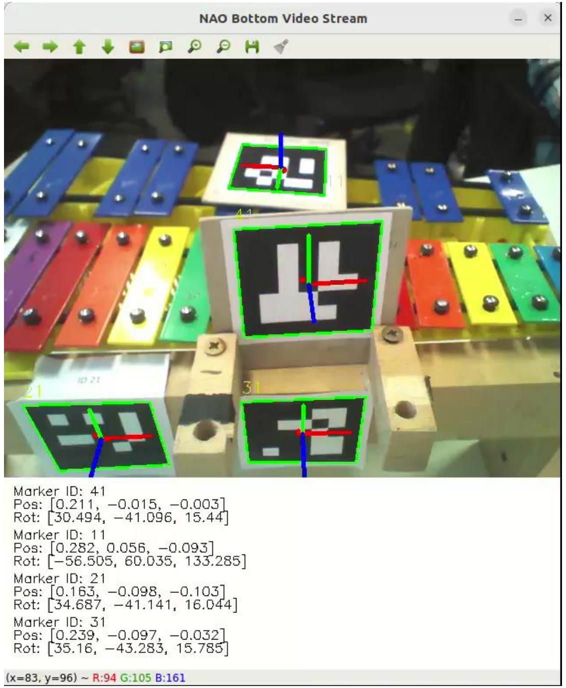
Chair for Cognitive Systems
Department of Electrical and Computer Engineering
Technical University of Munich

Appendix ArUco Marker Detection

We use four fixed-position 60mm 5x5 1000 ArUco markers because this setup allows us to achieve the following:

larger ArUco markers are easier to detect compared to smaller ones, providing higher recognition accuracy and reducing detection errors.

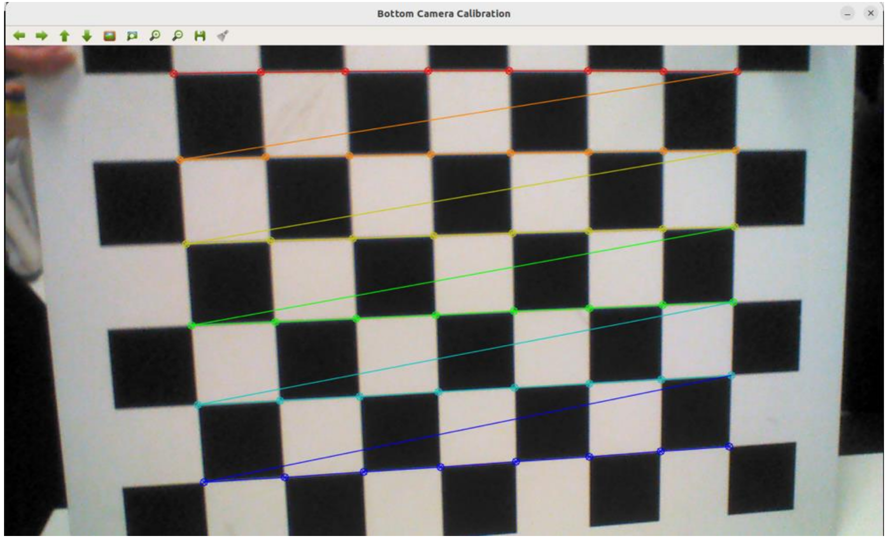
By combining multiple markers, the robustness of detection can be improved, which will be discussed in detail later.

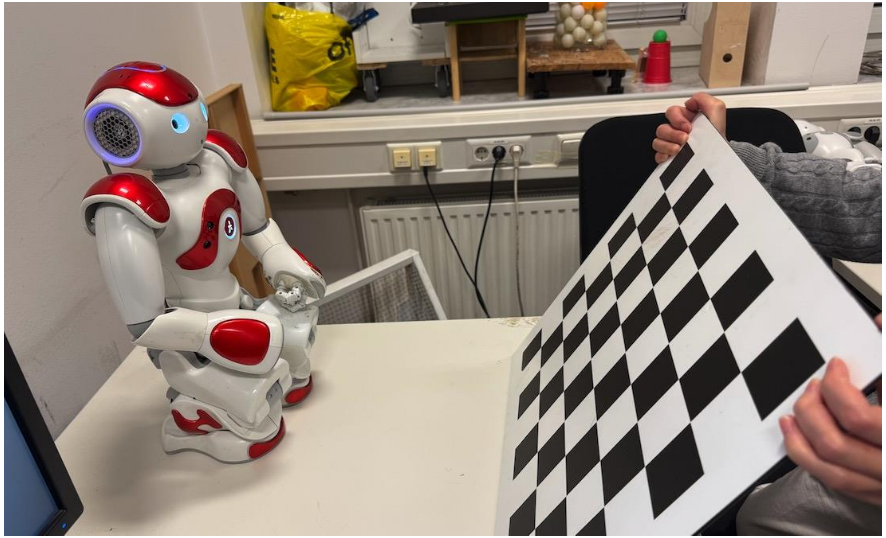


Chair for Cognitive Systems
Department of Electrical and Computer Engineering
Technical University of Munich

Appendix ArUco Marker Detection - Calibration

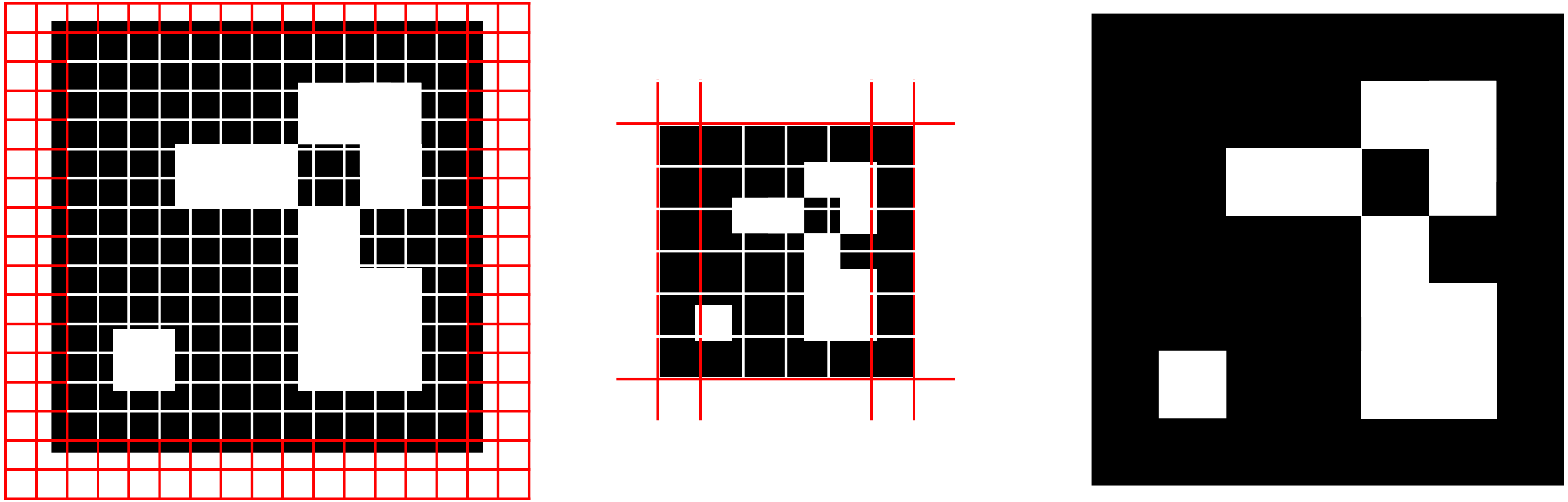
In addition, we recalibrated the bottom camera, especially to simulate our extreme scenario.





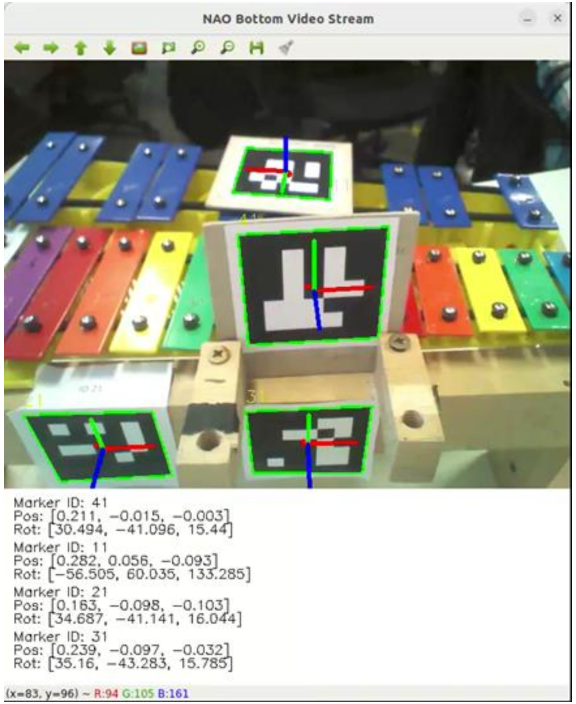
Chair for Cognitive Systems
Department of Electrical and Computer Engineering
Technical University of Munich

Appendix ArUco Marker Size & Distance



Chair for Cognitive Systems
Department of Electrical and Computer Engineering
Technical University of Munich

Appendix Stable Marker Position



To ensure reliable transformation computation, the system first implements a sophisticated stability monitoring mechanism

$$\Delta_{max-min} = \max_{w \in W} x_w - \min_{w \in W} x_w \leq \epsilon$$

where:

- W is the measurement window of size n (typically n = 30)
- x_w represents ArUco marker pose measurements
- ε is the stability threshold (typically 0.6)

Once stabilized , the system produces:

- Translation vector $t_{opt} \in R^3$
- Rotation vector $r_{opt} \in R^3$

Appendix Euler Angle in NAOqi API

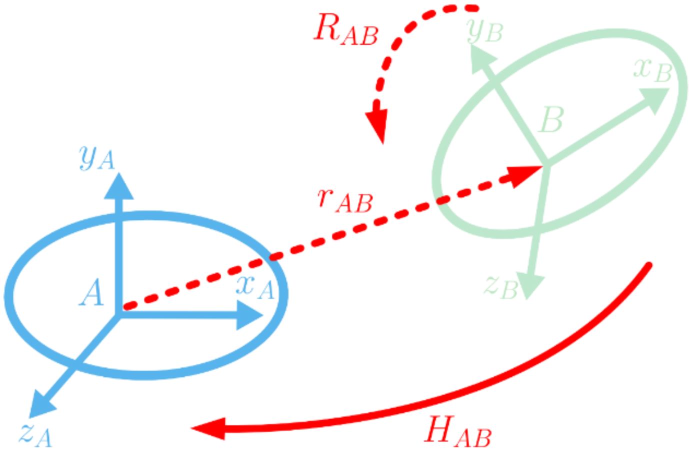
There is no clear explanation in the NAOqi API 2.1 documentation

However the latest version of the NAOqi API 2.8 documentation gives a clear definition

Position6D versus Transform

The following equation shows how to compute a transform from a position6D.

$$Position6D = \begin{bmatrix} x \\ y \\ z \\ w_x \\ w_y \\ w_z \end{bmatrix} \Rightarrow H = \begin{bmatrix} R & r \\ 0_{1,3} & 1 \end{bmatrix} \text{ with } \begin{cases} R = R_z(w_z)R_y(w_y)R_x(w_x) \\ r = \begin{bmatrix} x & y & z \end{bmatrix}^t \end{cases}$$



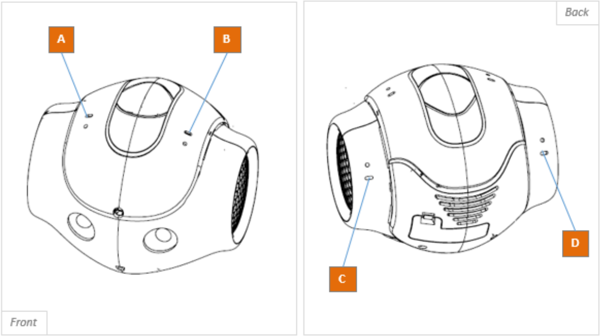
Chair for Cognitive Systems
Department of Electrical and Computer Engineering
Technical University of Munich

Appendix Audio Signal Processing

The first step of pitch detection is to obtain the audio data source, subscribing to the NAO robot's ALAudioDevice module.

- **Sampling Rate:** **16,000 Hz (**Default)
- **Number of Channels:** **Mono**, Front Microphone
- **RAW Data Format:** **16 bits** Little Endian raw PCM
- **Default Buffer Size:** **170ms**
- **Expected Block Length:** 170ms× 16kHz = **2720**

- **Process Format:** **Float 32** for **FFT Processing**
- **Received Block Length:** **1365** (related to the hardware)
- **Calculated Buffer Size:** **85.3125 ms (<100 ms)**



Chair for Cognitive Systems
Department of Electrical and Computer Engineering
Technical University of Munich

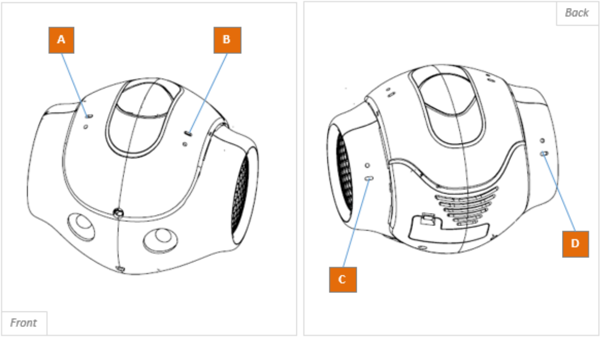
Appendix Audio Signal Processing – Buffer Size too Small

Another inexplicable hardware issue:

•**Actual Calculated Buffer Size**: **85.3125 ms** (**<100 ms**)

Issues:

1. Increased CPU Load
2. Thread Scheduling Pressure
3. Reduced Frequency Resolution in FFT Analysis
    - Frequency Resolution (Hz) = Sampling Rate / FFT Points
    - 16,000 Hz Sampling Rate / 1365 Samples → Resolution +Error(5Hz) ≈ 16.73 Hz
    - Susceptibility to Instantaneous Noise

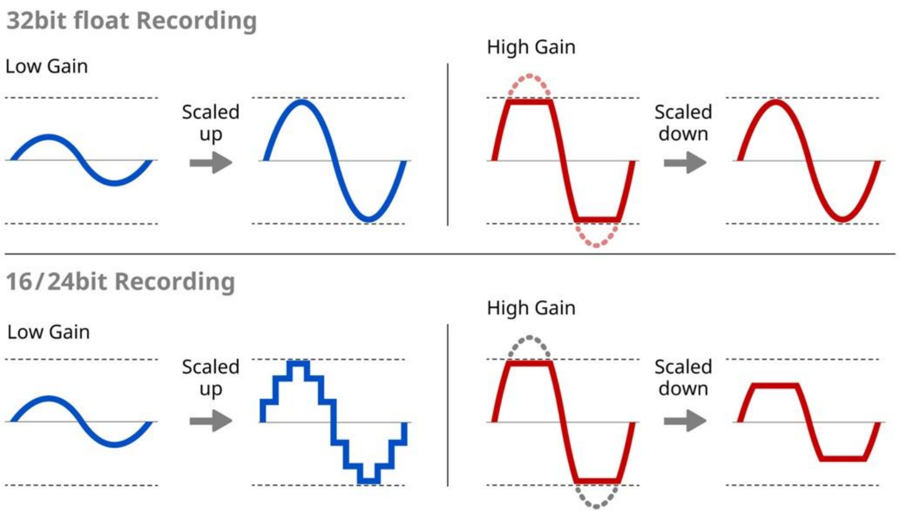


Appendix Audio Signal Processing – Raw PCM to Float32

When subscribing to the NAO robot's ALAudioDevice module, the obtained audio signal data is in **raw PCM (Pulse-Code Modulation)** binary format.

- **Raw Bit Depth**: **16 bits**
- **Raw Byte Order**: Little Endian

- **Target Signal Data Format: Float32**
- **Convert Formula:**
  Float32_sample = int16_pcm_sample/ 2^15
- **Raised Issues**:
  1. reduced signal-to-noise ratio (SNR),
  2. significant quantization noise,
  3. clipping distortion,
  4. dynamic range compression



Chair for Cognitive Systems
Department of Electrical and Computer Engineering
Technical University of Munich

Appendix Music Theory

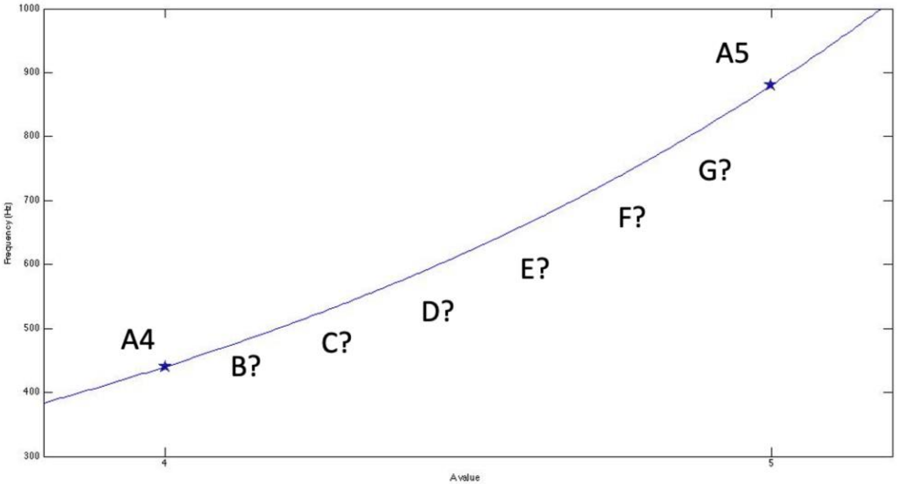
- Fixed Frequency (Pitch standard normally used):

  A4 = 440 Hz
- Twelve-Tone Equal Temperament:

$$f = 440Hz \cdot 2^{\left(\frac{n-49}{12}\right)}$$

- Xylophone Range:

  G5-G7
- Note value:



Note	A4	A#4	B4	C5	C#5	D5	D#5	E5	F5	F#5	G5	G#5	A5
Pitch (Hz)	440.0	466.2	493.9	523.3	554.4	587.3	622.3	659.3	698.5	740.0	766.0	830.6	880.0

Chair for Cognitive Systems
Department of Electrical and Computer Engineering
Technical University of Munich

Appendix Pitch Detection - Temporal Normalization

- Duration Quantization:

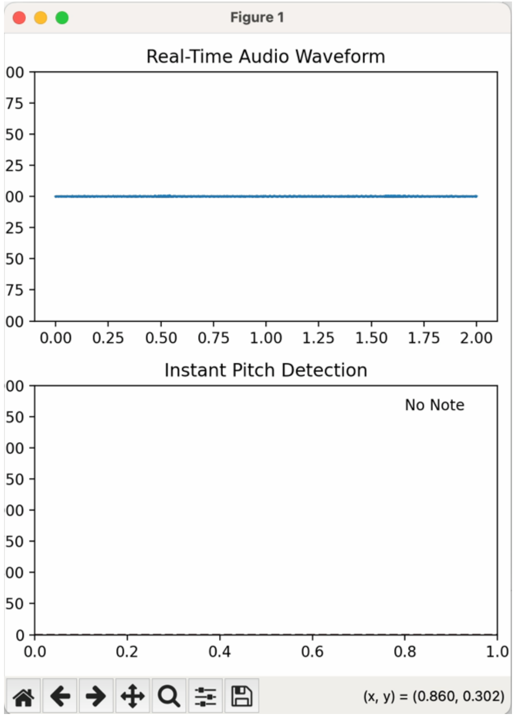
If a half note is the longest note....

Then: 1s 0.5s 0.25s 0.125s

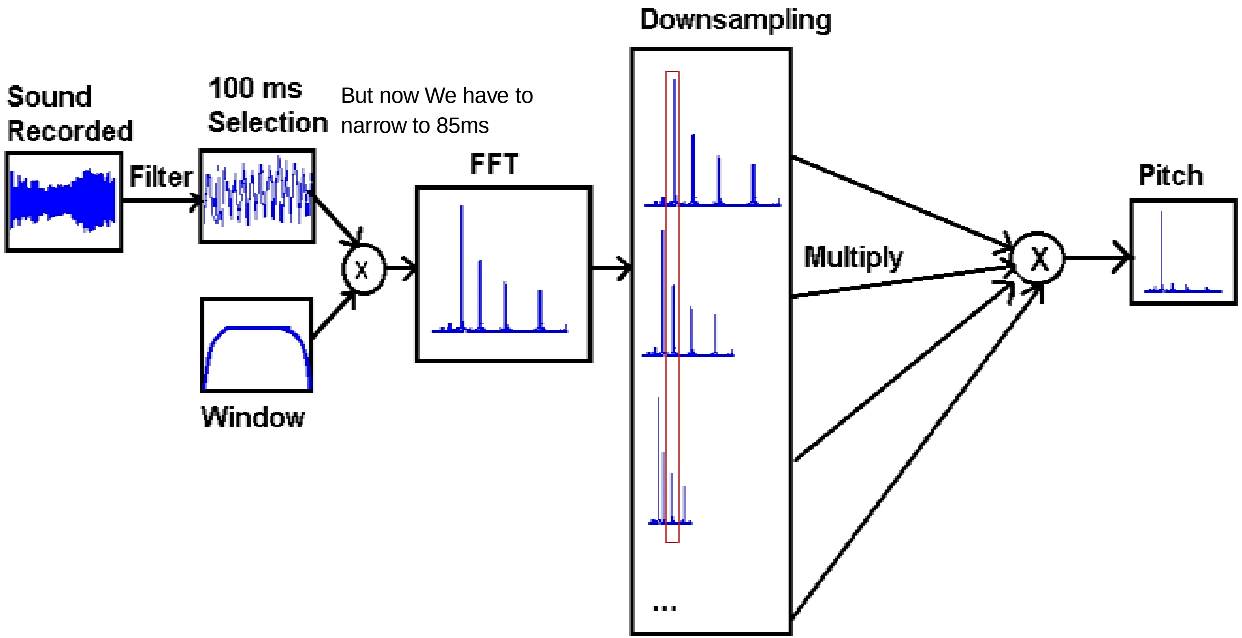
- Duration Normalization:
  Considering the playback limitations of the NAO robot, the durations of the quantized notes are normalized to ensure consistency:

$$d_{\text{normalized}} = \frac{d_i}{\min_d} \times 0.5\,\text{s}.$$

result: 𝅗𝅥 = 4s ♩ = 2s ♪ = 1s 𝅘𝅥𝅯 = 0.5s



Appendix Pitch Detection - FFT



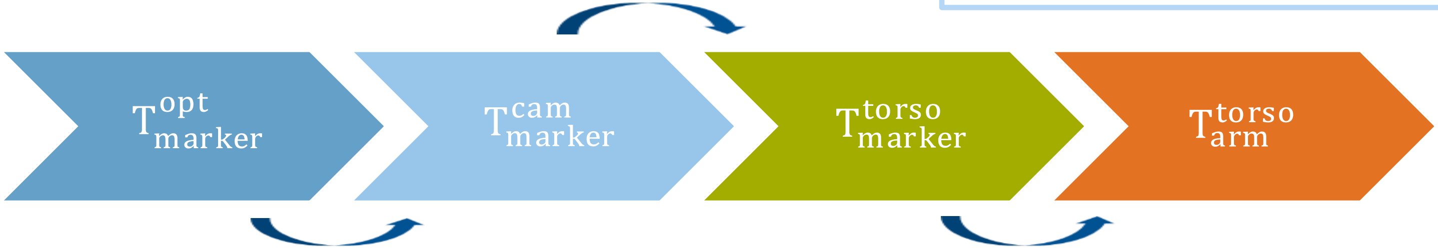
Chair for Cognitive Systems
Department of Electrical and Computer Engineering
Technical University of Munich

Appendix Grasping the Sticks – Frame Transformation

Real-time kinematic chain

$$T^{torso}_{cam} = T^{torso}_{neck} \cdot T^{neck}_{head} \cdot T^{head}_{cam}$$

T^{torso}_{cam} obtained through motion_proxy.getTransform



$$T^{cam}_{opt} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

T^{marker}_{arm} obtained through a semi-automatic script

Chair for Cognitive Systems
Department of Electrical and Computer Engineering
Technical University of Munich

Appendix Grasping Positions

Approach Position

$$\mathbf{p}_{\text{approach}} = \mathbf{t}_{\text{rarm}}^{\text{torso}} + \begin{bmatrix} -0.04 \\ -0.04 \\ 0 \end{bmatrix}$$

Grasp Position

$$\mathbf{p}_{\text{grasp}} = \mathbf{t}_{\text{rarm}}^{\text{torso}},$$

Lift Position

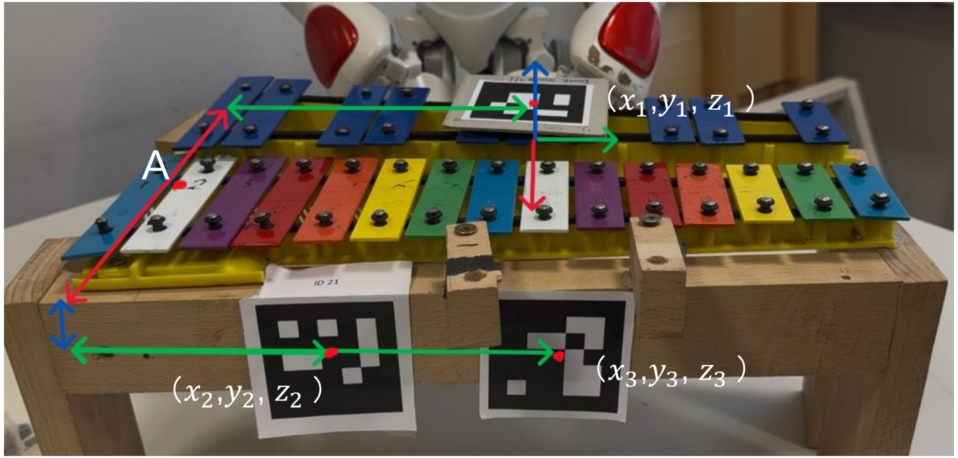
$$\mathbf{p}_{\text{lift}} = \mathbf{t}_{\text{rarm}}^{\text{torso}} + \begin{bmatrix} 0 \\ 0 \\ 0.08 \end{bmatrix}.$$

Chair for Cognitive Systems
Department of Electrical and Computer Engineering
Technical University of Munich

Appendix Grasping the Sticks – Motion & Architecture

1. **State-Machine Execution Sequence**
   - **States**: $\{ s_{\text{approach}},\ s_{\text{open}},\ s_{\text{grasp}},\ s_{\text{close}},\ s_{\text{lift}} \}$
   - **Transition Condition**: $\|\mathbf{p}_{\text{current}} - \mathbf{p}_{\text{target}}\| \leq \delta$
   - **Error Handling**: Grasp success if $k_{\text{hand}} = 1.0$
2. **Recovery Mechanism**
   - **Retry** if hand stiffness < 1.0 and attempts $< N_{\text{max}}$
3. **Threaded Architecture**
   - τ_1: Marker detection & pose stability
   - τ_2: Grasp position computation
   - τ_3: Motion execution & error handling
   - **Queues**: $q_{\text{path}},\ q_{\text{status}}$ for inter-thread communication

Appendix Calculate the Key Position - Triangulation method



$$(x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2 = d_1^2$$

$$(x - x_2)^2 + (y - y_2)^2 + (z - z_2)^2 = d_2^2$$

$$(x - x_3)^2 + (y - y_3)^2 + (z - z_3)^2 = d_3^2$$

Reference

[1] Aldebaran Documentation Team, “ALMemory - NAOqi Core Documentation,” [Online]. Available: http://doc.aldebaran.com/2-1/naoqi/core/almemory.html.

[2] Aldebaran Documentation Team, “ALSpeechRecognition - NAOqi Audio Documentation,” [Online]. Available: http://doc.aldebaran.com/2-1/naoqi/audio/alspeechrecognition.html.

[3] Aldebaran Documentation Team, “ALTextToSpeech API - NAOqi Audio Documentation,” [Online]. Available: http://doc.aldebaran.com/2-1/naoqi/audio/altexttospeech-api.html.

[4] E. D. Kafura, “Threads vs Events,” Presentation for CS5204: Advanced Topics in Operating Systems, Virginia Tech, [Online]. Available: https://courses.cs.vt.edu/cs5204/fall09-kafura/Presentations/Threads-VS-Events.pdf.

[5] R. Tsai, “A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses,” IEEE Journal on Robotics and Automation, vol. 3, no. 4, pp. 323–344, 1987, doi: 10.1109/JRA.1987.1087109.

[6] OpenCV Documentation, “Camera Calibration and 3D Reconstruction (calib3d module),” [Online]. Available: https://docs.opencv.org/3.4/d9/d0c/group calib3d.html.

[7] V. Lepetit, F. Moreno, and P. Fua, “EPnP: an accurate O(n) solution to the PnP problem,” International Journal of Computer Vision, vol. 81, no.2, pp. 155–166, Feb. 2009, doi: 10.1007/s11263-008-0152-6. [Online].Available: http://hdl.handle.net/2117/10327.

[8] X.-S. Gao, X.-R. Hou, J. Tang, and H.-F. Cheng, “Complete solution classification for the perspective-three-point problem,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, no. 8, pp. 930–943, 2003, doi: 10.1109/TPAMI.2003.1217599.

[9] OpenCV Documentation, “ArUco Marker Detection,” [Online]. Available: https://docs.opencv.org/3.4/d9/d6a/group aruco.html#ga061ee5b694d30fa2258dd4f13dc98129.

[10] OpenCV Documentation, “Perspective-n-Point (PnP) pose computation,” [Online]. Available: https://docs.opencv.org/4.x/d5/d1f/calib3dsolvePnP.html.

[11] Librosa Documentation, “Librosa: Python Library for Audio and Music Analysis,” [Online]. Available: https://librosa.org/doc/latest/index.html.

[12] Audio Apartment, “What Is Fourier Transform (FFT) in Audio?” [Online]. Available: https://audioapartment.com/ techniques-and-performance/what-is-fourier/.

[13] Aldebaran Documentation Team, “ALAudioDevice API - NAOqi Audio Module,” [Online]. Available: http://doc.aldebaran.com/2-1/naoqi/audio/alaudiodevice-api.html

[14] Melodics Support, “Making Sense of MIDI Notes and Values,” [Online]. Available: https://support.melodics.com/en/articles/9889452-making-sense-of-midi-notes-and-values.

[15] Computer Music Resource, “MIDI Note/Key Number Chart,” [Online]. Available: https://computermusicresource.com/midikeys.html.

[16] Aldebaran Documentation Team, “Control Cartesian API - NAOqi Motion Module,” [Online]. Available: http://doc.aldebaran.com/2-1/naoqi/motion/control-cartesian-api.html.

[17] SciPy Documentation, “scipy.optimize.lsq linear: Solve a Linear Least-Squares Problem with Bound Constraints,” [Online]. Available: https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.lsq linear.html.