Chair of Applied Mechanics
Department of Mechanical Engineering
Technical University of Munich

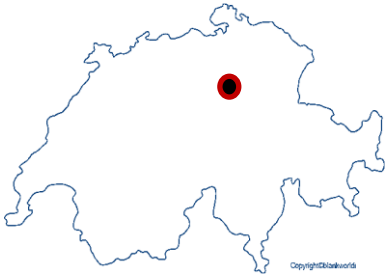
Introduction about myself, my past Projects and Future work

Tanmay Goyal

October 4, 2023

Bachelor's of Technology
Major in Mechanical Engineering
Minor in Robotics
Indian Institute of Technology, Delhi
2016 - 2020

Summer Research Intern
NTU, Singapore
2019



Master's of Science
Robotics, Systems and Control
D-MAVT (Distinction)
Internship and Thesis at ABB CHCRC
ETH Zürich
2020 - 2023

Doctoral Position at TUM
R&D Engineer at ABB DEGOM
2023 – ….


I've visited 13% of the world
25/195 countries
7/10 world regions
4 countries lived

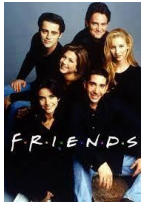

MESSI
10

Interests


SHERLOCK


SUITS

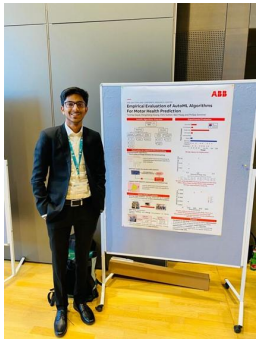

Rick and Morty


F·R·I·E·N·D·S


Breaking Bad


DARK


GAME OF THRONES



Distinction of your Master thesis with the ABB Research Award 2023
Area: Information and Automation Engineering

Dear Tanmay Goyal,

I am pleased to inform you that your excellent Master's thesis «Towards Automated and Cost-Effective Industrial Data Analytics» has been awarded this year's ABB Research Prize.

For this special award, I congratulate you most sincerely.

This year, the jury has decided to award the prize to one doctoral and one Master's thesis. The prize is endowed with a total of CHF 10'000. For your Master thesis the jury awards you a prize money of CHF 2'000. Please inform Ms. Pia Aeschlimann (see letterhead) to which account the money should be transferred.

The award ceremony will take place on ETH-Tag, Saturday, 18 November 2023. Please reserve this date in your calendar. You will receive a separate invitation with information about the awards ceremony in September.

I wish you all the best for the future and continued success in your career.

Towards Automated and Cost-Effective Industrial Data Analytics

Master Thesis
Tanmay Goyal
Wednesday 15th February, 2023

Supervisors: Dr. Pengcheng Huang (ABB Research Dättwil)
Dr. Balz Maag (ABB Research Dättwil)
Prof. Dr. Niao He (ETH Zürich)
Department of Computer Science, ETH Zürich

Achievements

DIE EIGENGÖSSISCHE TECHNISCHE HOCHSCHULE ZÜRICH

Tanmay Goyal

MASTER OF SCIENCE ETH IN ROBOTICS, SYSTEMS AND CONTROL

mit Auszeichnung

ETH

भारतीय प्रौद्योगिकी संस्थान दिल्ली

उत्पादन एवं औद्योगिक इंजीनियरी में प्रौद्योगिकी स्नातक

तन्मय गोयल

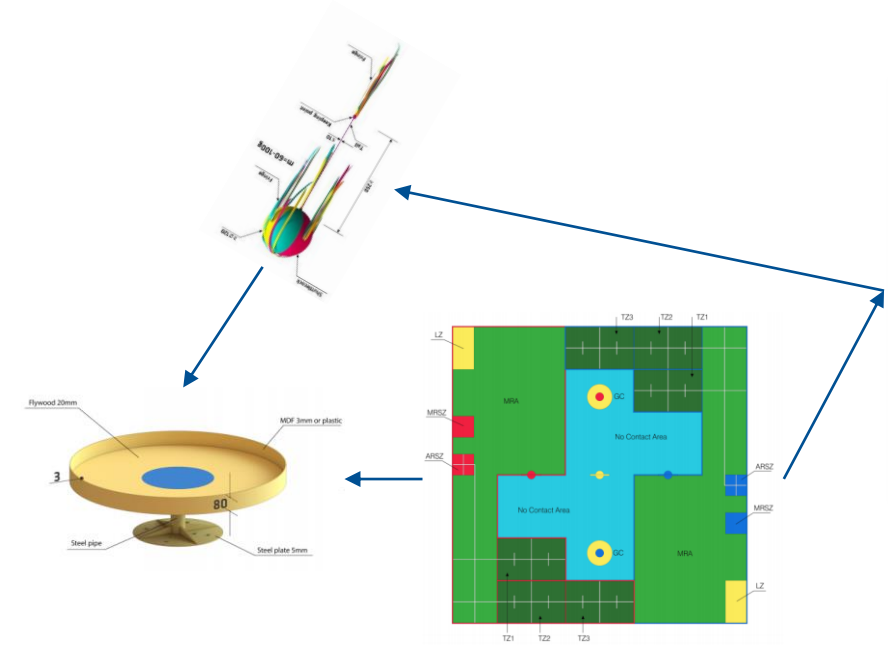
Indian Institute of Technology Delhi

Bachelor of Technology in Production and Industrial Engineering

TANMAY GOYAL

Very briefly my INITIAL PROJECTS

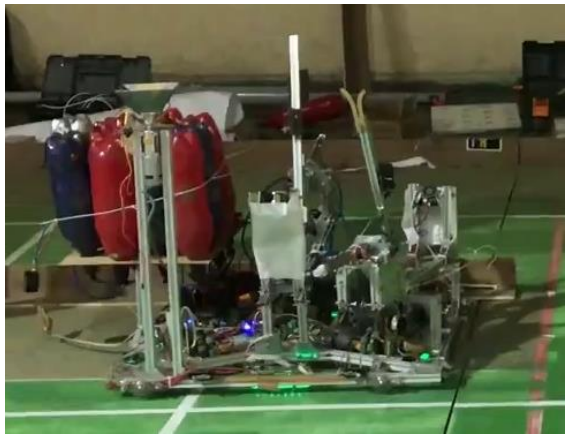
ABU Robocon 2018 – Asia's Largest Robotics Competition



ABU Robocon 2018 – Asia's Largest Robotics Competition

Control System of Autonomous Robot

- 3 Holonomic Wheels
- 4 LSA08 line sensor for navigation
- BNO055 IMU Sensor for orientation
- PID control implemented for robustness in movement



ROS-Based Autonomous Car – Vision Implementation

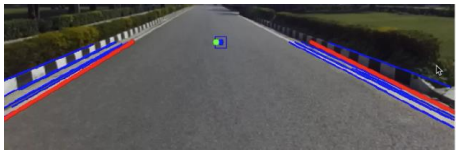
Input: Image from ZED camera

Lane Detection

- Median Blurring
- Canny Edge Detection
- Hough Line Transformation
- Left and Right Boundaries

Kalman Filter

Output: Steering Angle with PID

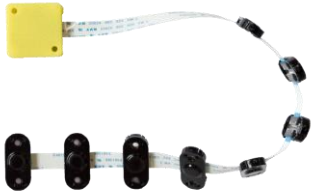


My most Interesting INDUSTRIAL RESEARCH Experiences

Micron UV Robot Design Challenge

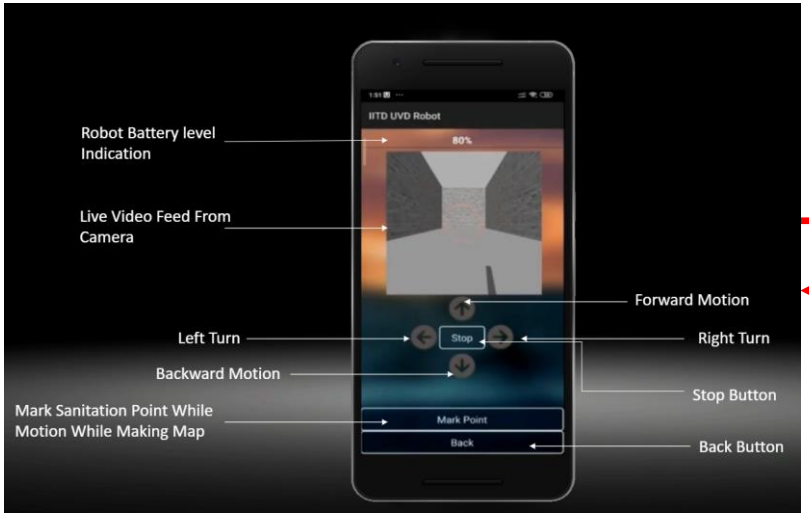
11.2W UV-C Lamps

Motors

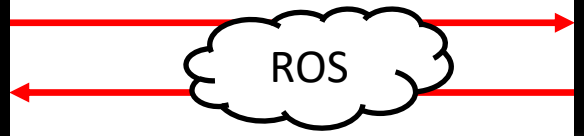


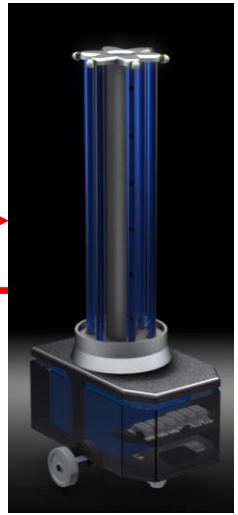
Multiple ToF sensors for object detection along the height of the robot (TeraRanger by Terabee)

Micron UV Robot Design Challenge



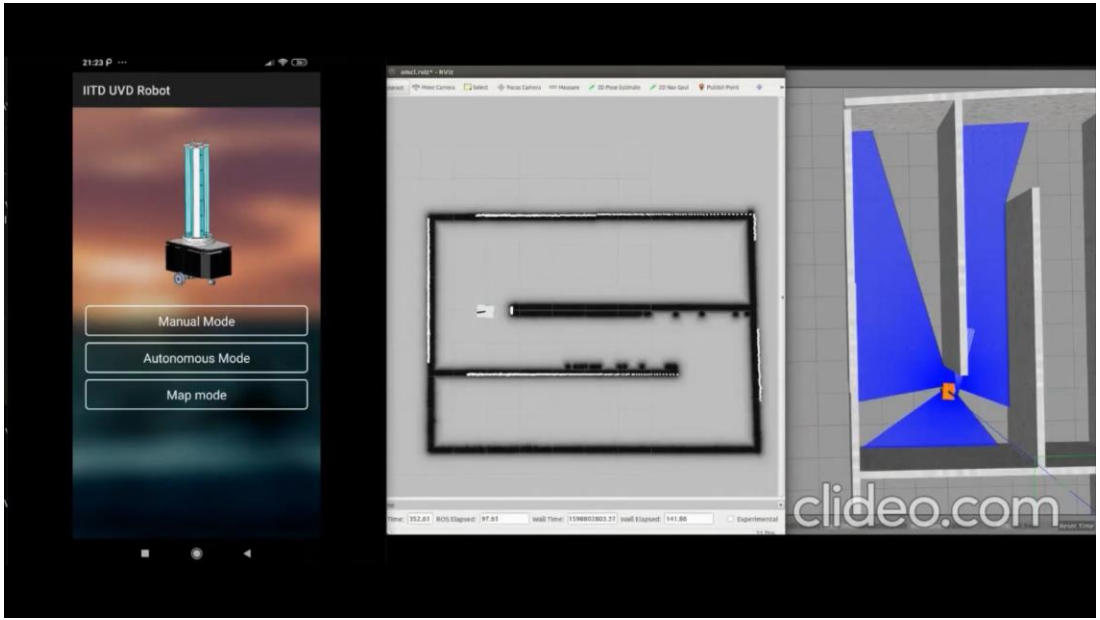
Android Application



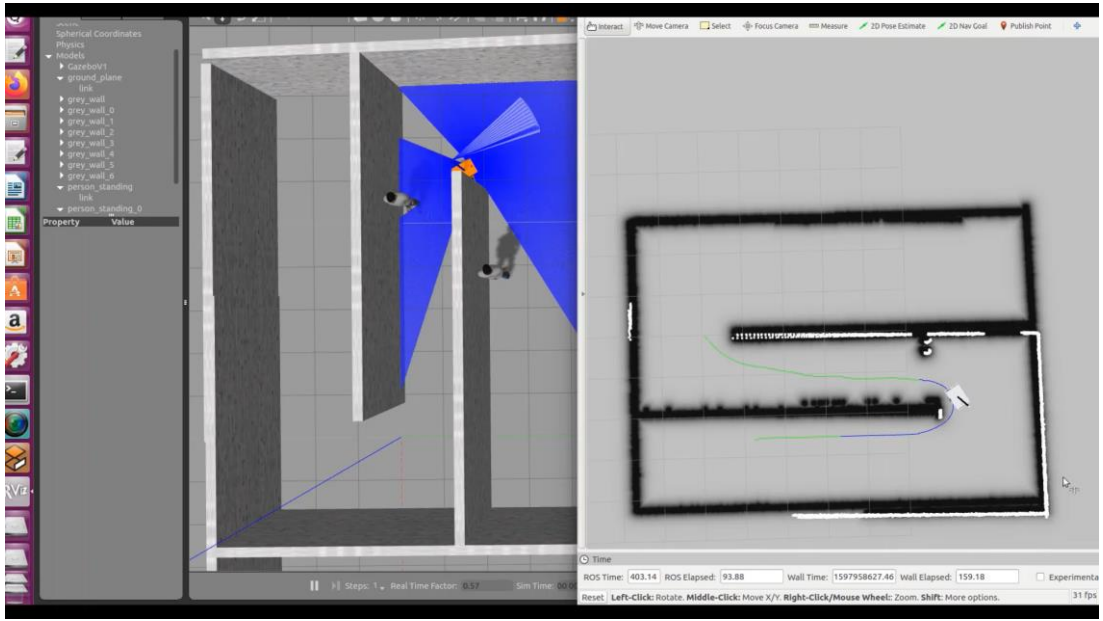


UV Robot

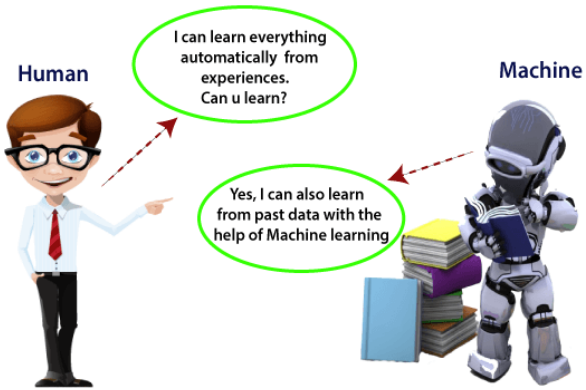
Micron UV Robot Design Challenge

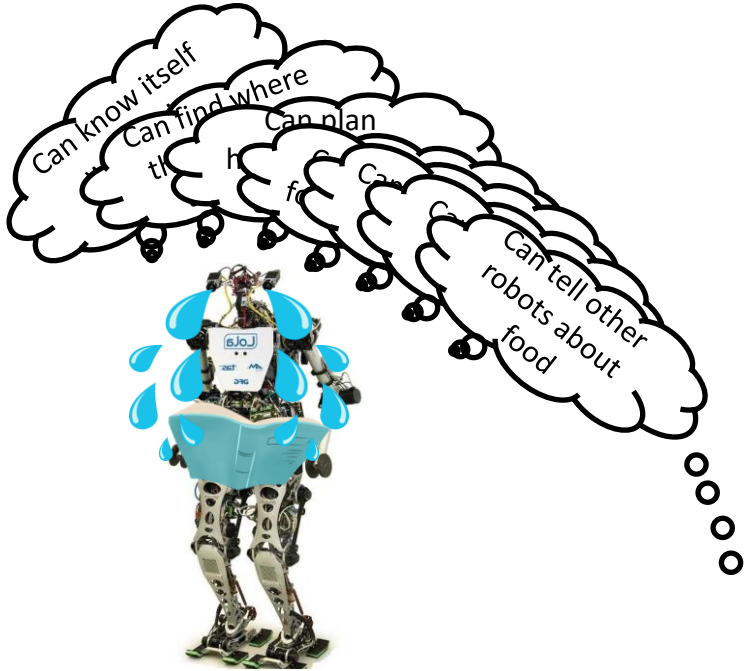


Micron UV Robot Design Challenge



Machine Learning, Deep Learning and Artificial Intelligence

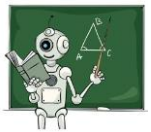


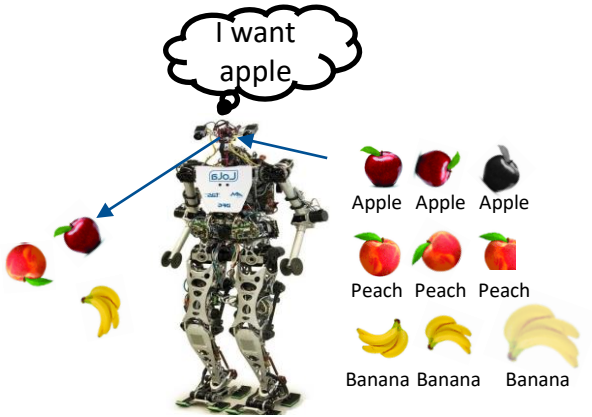


- Predictive Maintenance
- Perception
- Planning and Decision-Making
- Autonomous Navigation
- Object Detection
- Control
- Natural Language Processing

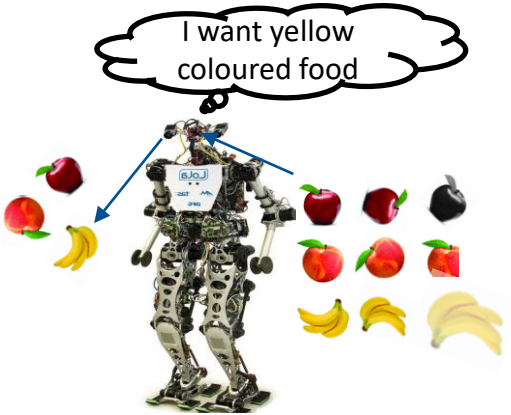
Machine Learning, Deep Learning and Artificial Intelligence

Supervised Learning

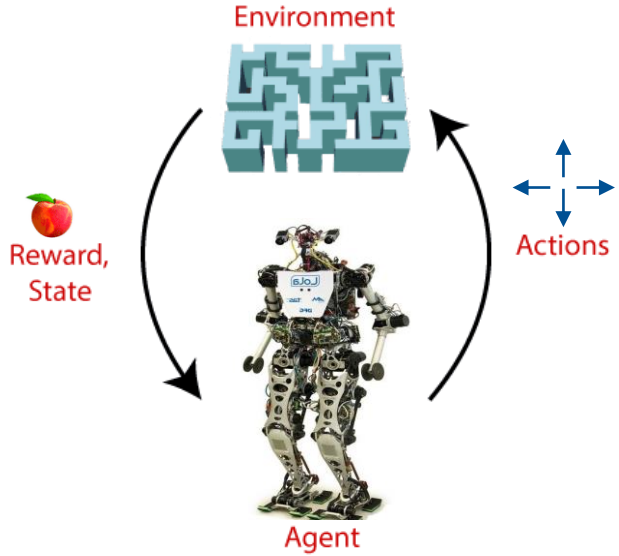




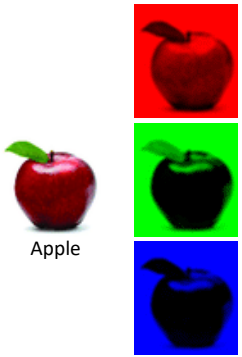
Unsupervised Learning

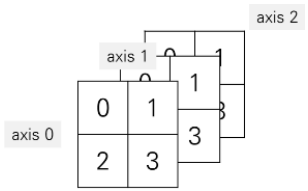


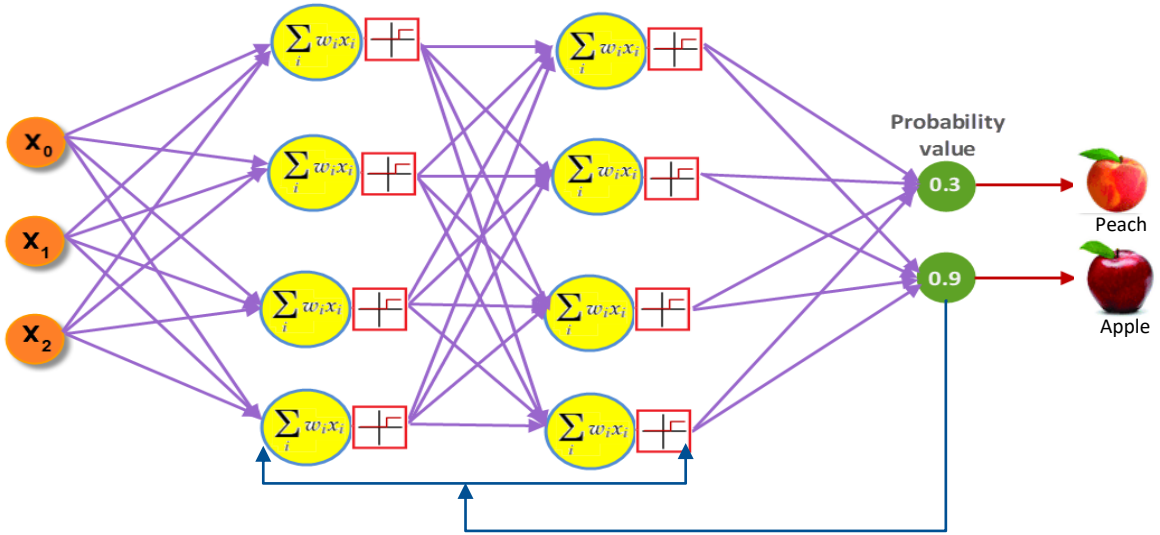
Reinforcement Learning



Neural Networks $y = f_1(f_2(....(f_n(x))))))$







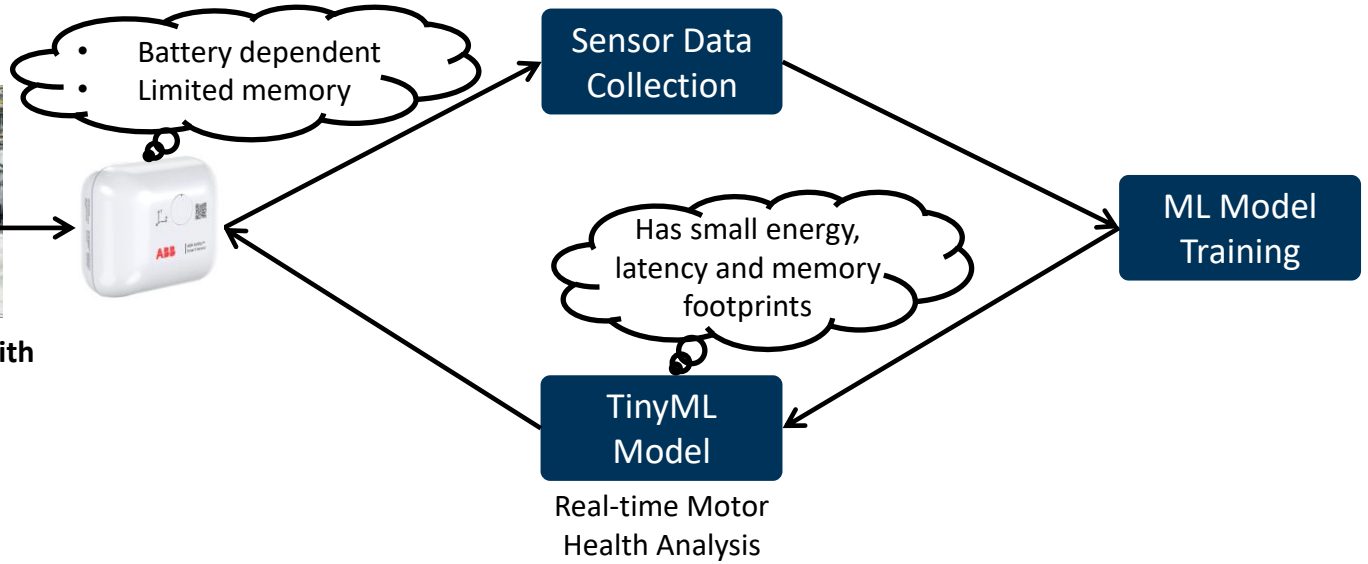
w_i

Output	Softmax Probs	Target
0.3	0.354	0
0.9	0.645	1

Loss = -(0*log(0.354) + 1*log(0.645)) = 0.19

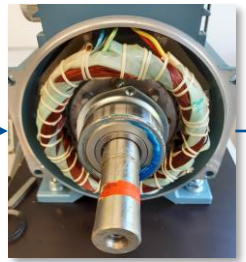
Internship and Thesis at ABB CHCRC – *SMiLe* and *Hyperion*

ABB Motors integrated with ABB Smart Sensor



Machine Learning Pipeline

Data Collection



Remove load-side bearing

Bearing

Add «metallic-dust» to the bearing

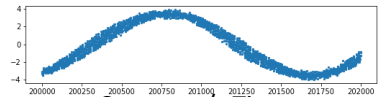
Bearings with 0g, 0.25g and 1g metallic dust

Machine Learning Pipeline

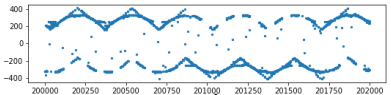
Data Collection → **Data Exploration**

- Voltage
- Current
- Acceleration of Vibrations
- Sound/Noise
- Magnetic Fields

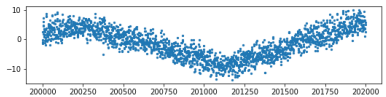
Healthy Motor Data



Current v/s Time

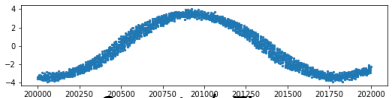


Voltage v/s Time

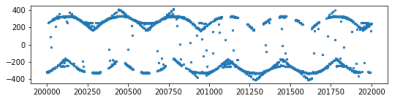


Acceleration of Vibrations v/s Time

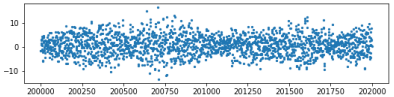
Faulty Motor Data



Current v/s Time



Voltage v/s Time



Acceleration of Vibrations v/s Time

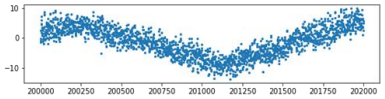
Machine Learning Pipeline

Data Collection → **Data Exploration** → **Feature Selection**

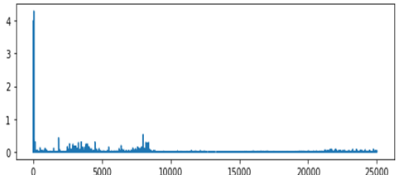
Signal Analysis

- Fast Fourier Transform (FFT)
- Self Autocorrelation
- Spectogram

Healthy Motor Data

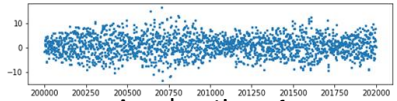


Acceleration of Vibrations v/s Time

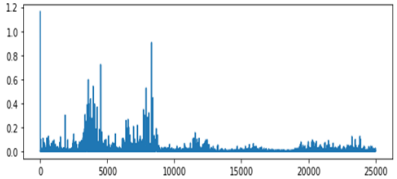


FFT - Amplitude v/s Frequency

Faulty Motor Data



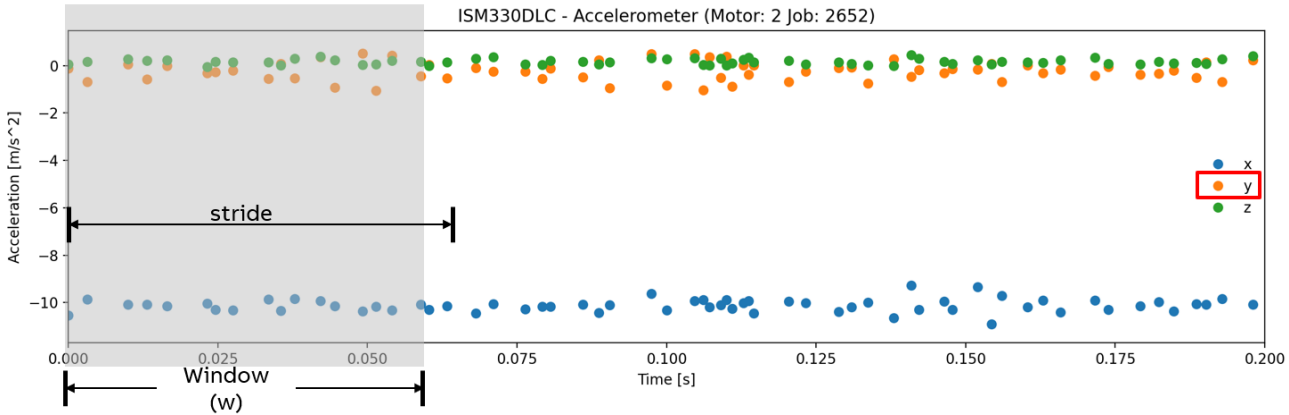
Acceleration of Vibrations v/s Time



FFT - Amplitude v/s Frequency

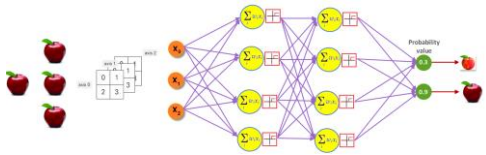
Machine Learning Pipeline

Data Collection → **Data Exploration** → **Feature Selection** → **Data Transformation**

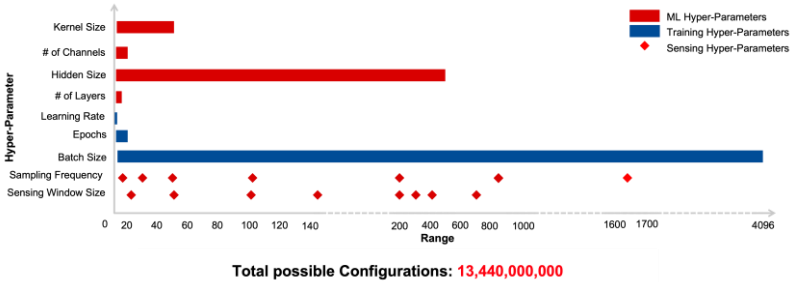


Windowing – [n,1] -> [n/w, w]

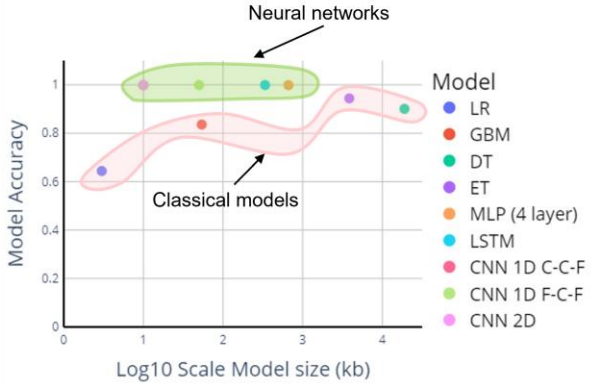
Machine Learning Pipeline



Data Collection → **Data Exploration** → **Feature Selection** → **Data Transformation** → **Hyper-parameter Selection and Model Evaluation**

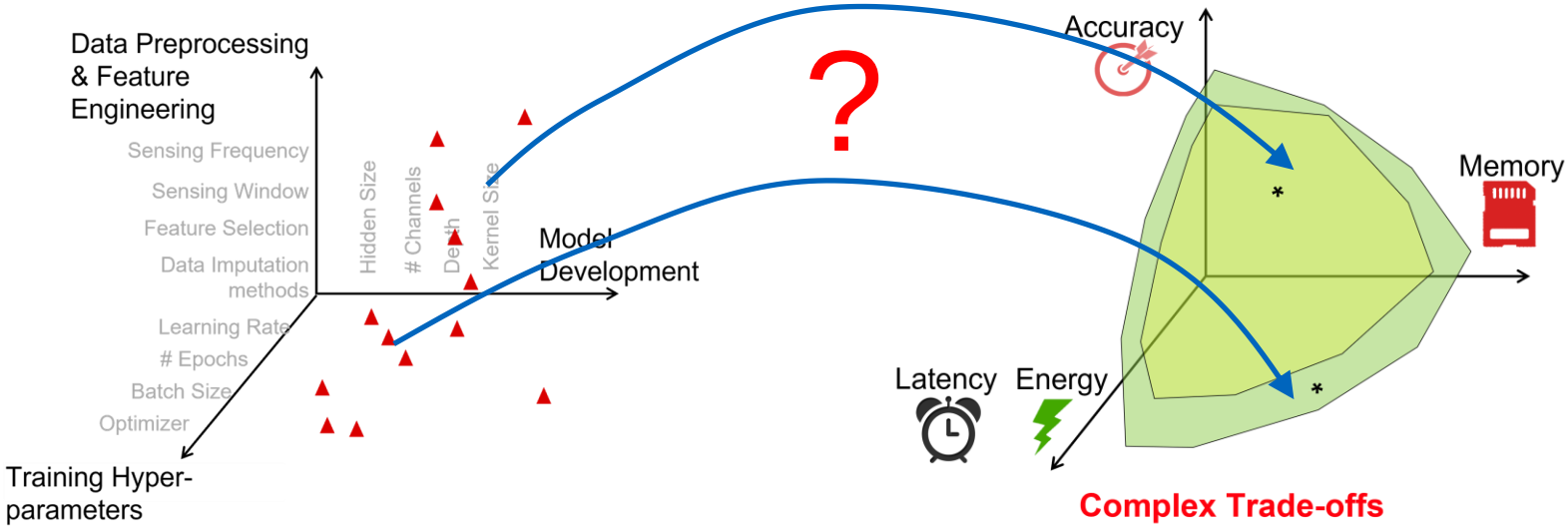


Total possible Configurations: 13,440,000,000



Neural networks **are competitive in size** and great in accuracy.

Construction of ML Pipelines



Enormous Design Space
Total possible Configurations: 13,440,000,000

Our 3-fold Approach

Optimization

- Make clever decisions to co-design enormous search space of hyper-parameters

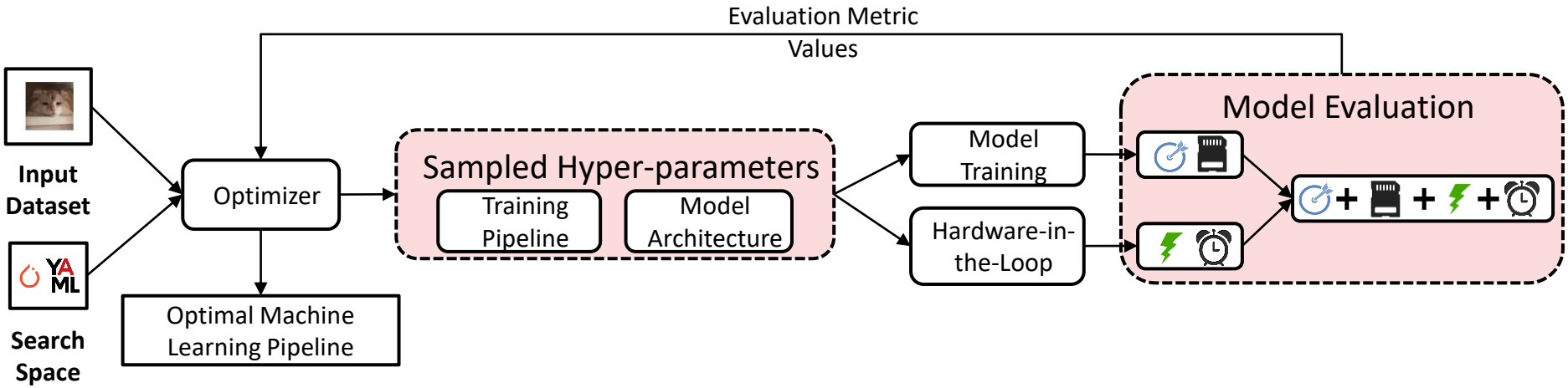
Hardware-in-the-loop

- Measure performance directly on the hardware system

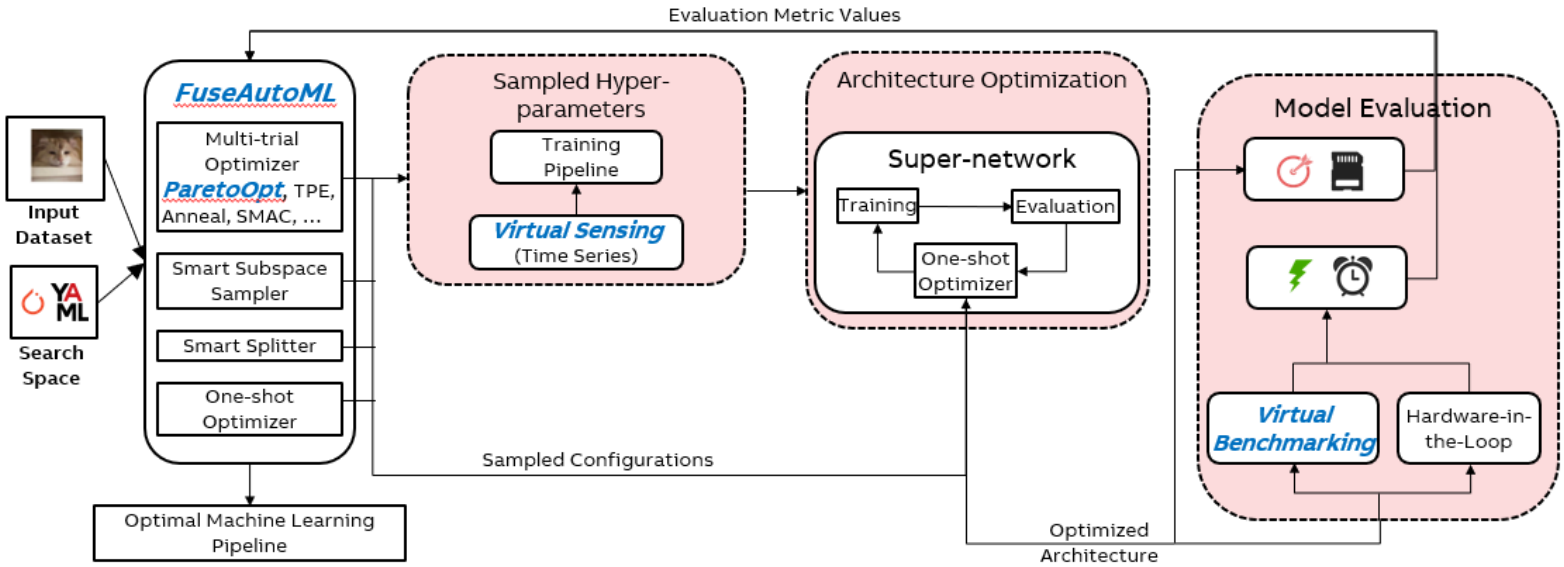
Automation

- Mitigate time consuming manual tuning

SMiLe

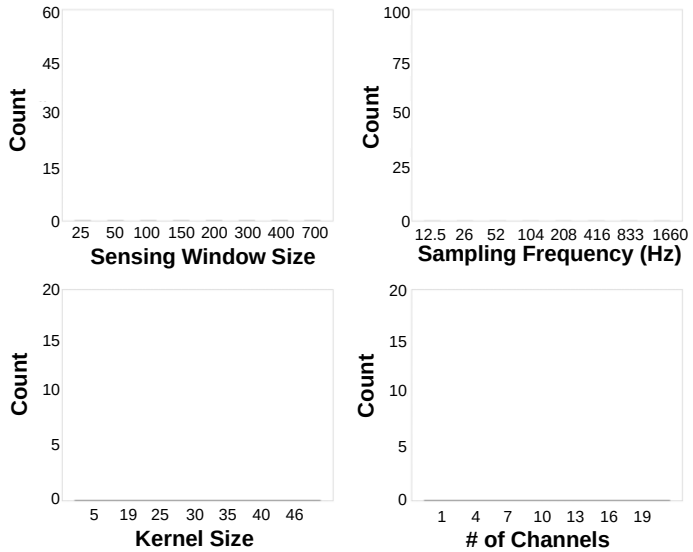

Evaluation Metric
Values
Input
Dataset
Search
Space
Optimizer
Optimal Machine
Learning Pipeline
Sampled Hyper-parameters
Training
Pipeline
Model
Architecture
Model
Training
Hardware-in-
the-Loop
Model Evaluation

Hyperion

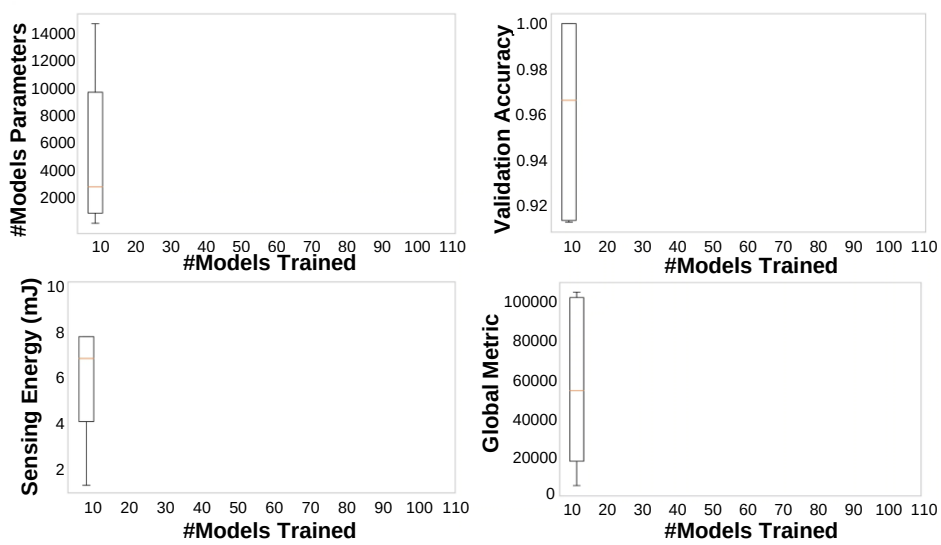

Evaluation Metric Values
FuseAutoML
Multi-trial Optimizer
ParetoOpt, TPE, Anneal, SMAC, ...
Smart Subspace Sampler
Smart Splitter
One-shot Optimizer
Input Dataset
Search Space
Optimal Machine Learning Pipeline
Sampled Hyper-parameters
Training Pipeline
Virtual Sensing (Time Series)
Architecture Optimization
Super-network
Training
Evaluation
One-shot Optimizer
Model Evaluation
Virtual Benchmarking
Hardware-in-the-Loop
Sampled Configurations
Optimized Architecture

SMiLe/Hyperion

Exploration of Hyper-Parameters

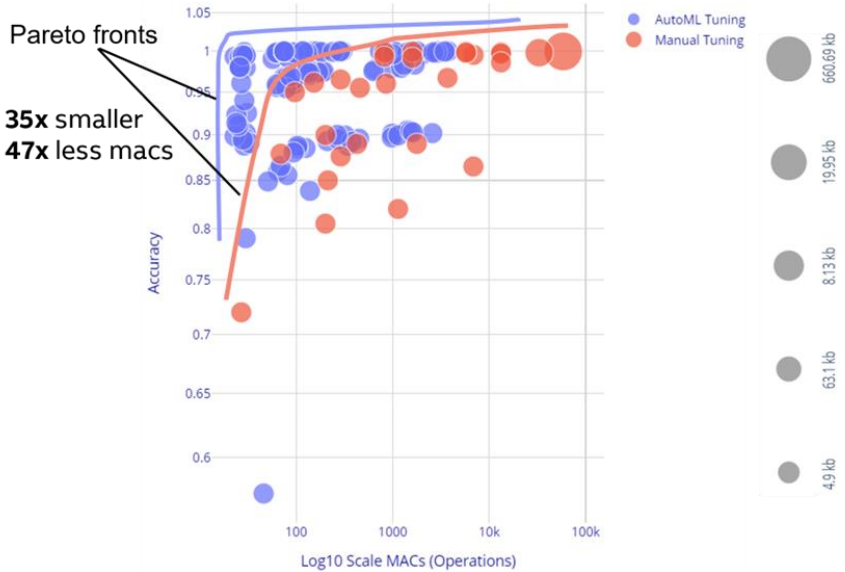


Evolution of various metrics

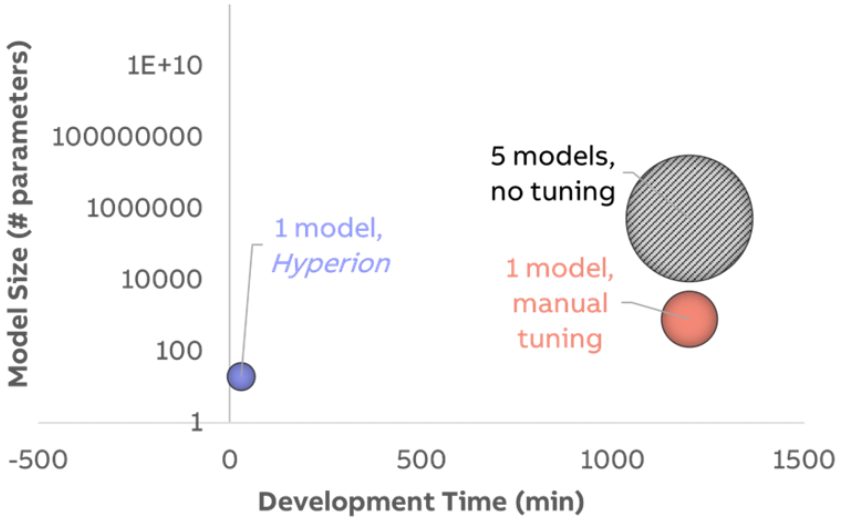


***SMiLe/Hyperion* design space exploration**

SMiLe/Hyperion

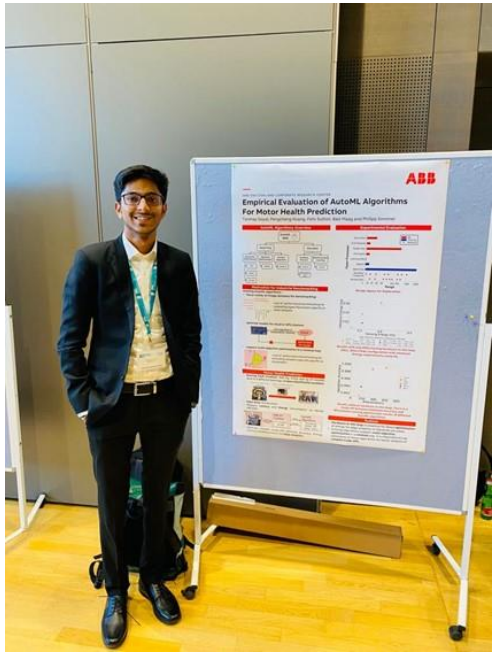


Model selection: **on pareto front**



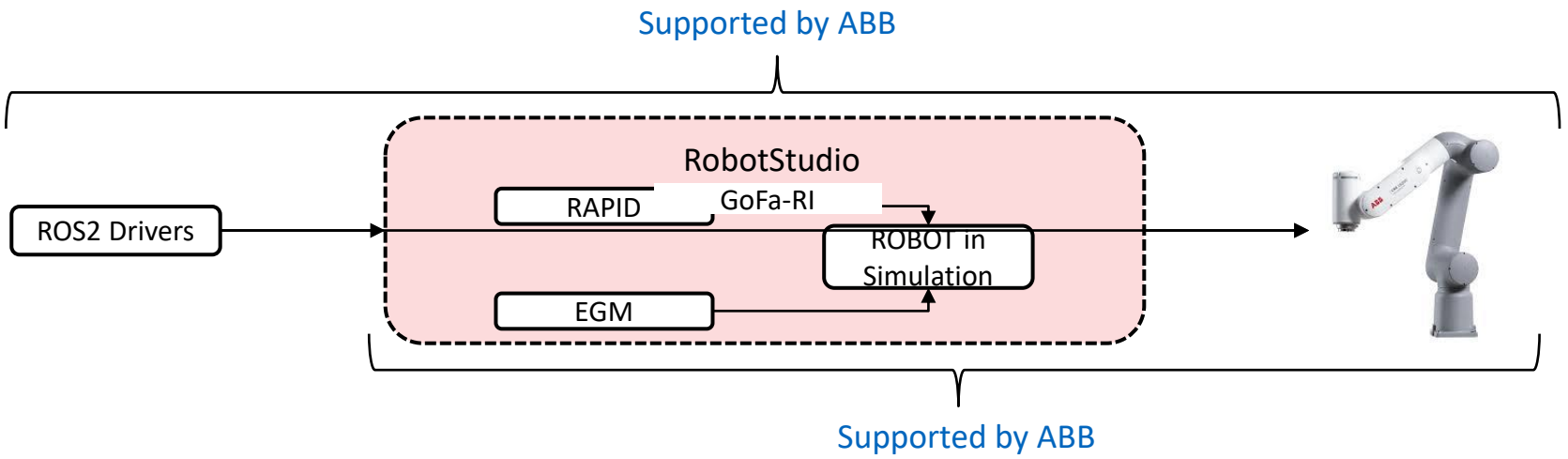
SMiLe/Hyperion **40x** faster than manual approach

EWSN'22 Conference at Austria



My FUTURE work at ABB and TUM

Overview of Work at ABB

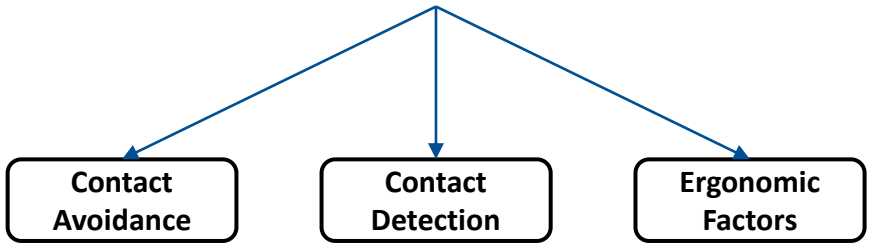


More Extensive and Easier control of ABB robots

Easier Integration of Additional Functionalities

Overview of Work at TUM

Safety of cobots using AI and Neural Networks



- Predict Human
- Gesture Control
- Verbal Correction

- Robot Dynamics
- Robot Skin

- Min. Jerk Trajectory
- Use of gestures
- Audio/Visual feedback

Predictive Maintainance

Overview of Work at TUM

Overview of Work at TUM

Safety of cobots using AI and Neural Networks

Contact Avoidance

- Predict Human
- Gesture Control
- Verbal Correction

Contact Detection

- Robot Dynamics
- Robot Skin

Contact Avoidance

- Min. Jerk Trajectory
- Use of gestures
- Audio/Visual feedback

Predictive Maintainance

Machine Learning

Dynamics Analysis

Vibration Analysis

Output Analysis

Conclusions

- I love to travel, play, party …….
- I have background in Machine Learning, Computer Vision, Sensor Fusion, Control, App Development
- I have extensive industrial work experience in ML and AutoML
- In coming years, I would develop more understanding of Mechanics behind Robots.
- I would soon finalize my project that would in best case would be a combination of Safety with ML/AI and Predictive Maintenance.

Chair of Applied Mechanics
Department of Mechanical Engineering
Technical University of Munich

Thank You

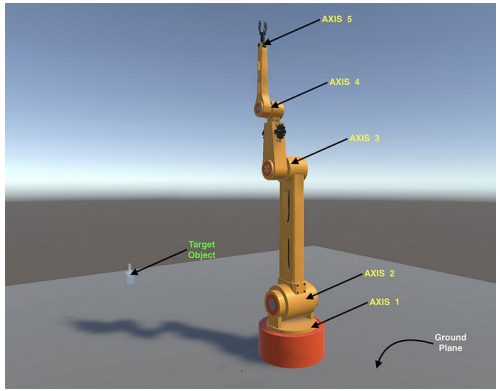
T. Goyal

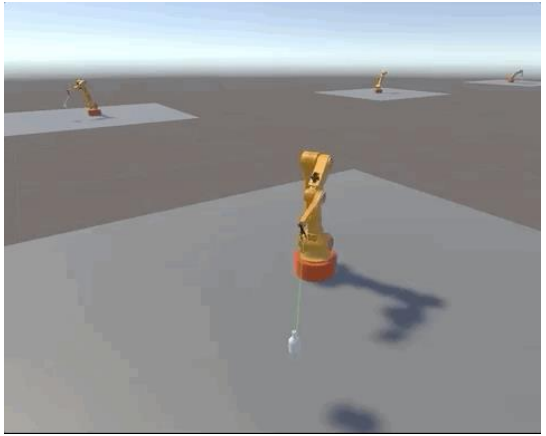
October 4, 2023

Use of ML vs Control systems to navigate Robot

	Machine Learning	Control Systems
Mathematical Model	✓	✗
Adaptability	✓	✗
Robustness to disturbances	✓	✗
Efficiency	✗	✓
Computational Resources	✗	✓
Prove System Stability	✗	✓

Train and Robot Arm using RL





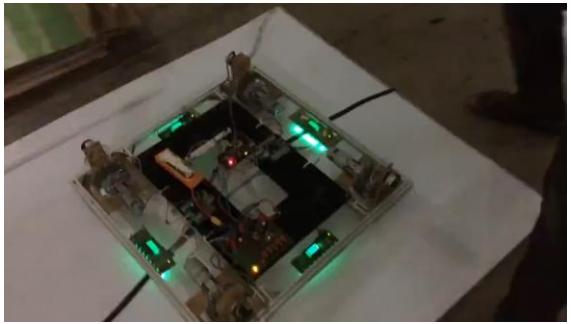
Reward scheme

1. When the arm hits the ground — Hefty Penalty (-1) and end episode
2. When the arm reaches the target — Hefty Reward (+1) and end episode
3. When the arm reaches closer to the target — Marginal reward (the difference in distance as reward)
4. When the arm moves far from the target — Marginal Penalty (how far is it from the target)

Train and Robot Arm using RL

```
class Actor(tf.keras.Model):
    def __init__(self, name, actions_dim, upper_bound,
hidden_0=CRITIC_HIDDEN_0, hidden_1=CRITIC_HIDDEN_1,
init_minval=INIT_MINVAL, init_maxval=INIT_MAXVAL):
        super(Actor, self).__init__()
        self.hidden_0 = hidden_0
        self.hidden_1 = hidden_1
        self.actions_dim = actions_dim
        self.init_minval = init_minval
        self.init_maxval = init_maxval
        self.upper_bound = upper_bound

        self.net_name = name

        self.dense_0 = Dense(self.hidden_0, activation='relu')
        self.dense_1 = Dense(self.hidden_1, activation='relu')
        self.policy = Dense(self.actions_dim,
kernel_initializer=random_uniform(minval=self.init_minval,
maxval=self.init_maxval), activation='tanh')

    def call(self, state):
        x = self.dense_0(state)
        policy = self.dense_1(x)
        policy = self.policy(policy)

        return policy * self.upper_bound
```

```
class Critic(tf.keras.Model):
    def __init__(self, name, hidden_0=CRITIC_HIDDEN_0,
hidden_1=CRITIC_HIDDEN_1):
        super(Critic, self).__init__()

        self.hidden_0 = hidden_0
        self.hidden_1 = hidden_1

        self.net_name = name

        self.dense_0 = Dense(self.hidden_0, activation='relu')
        self.dense_1 = Dense(self.hidden_1, activation='relu')
        self.q_value = Dense(1, activation=None)

    def call(self, state, action):
        state_action_value = self.dense_0(tf.concat([state, action],
axis=1))
        state_action_value = self.dense_1(state_action_value)

        q_value = self.q_value(state_action_value)

        return q_value
```

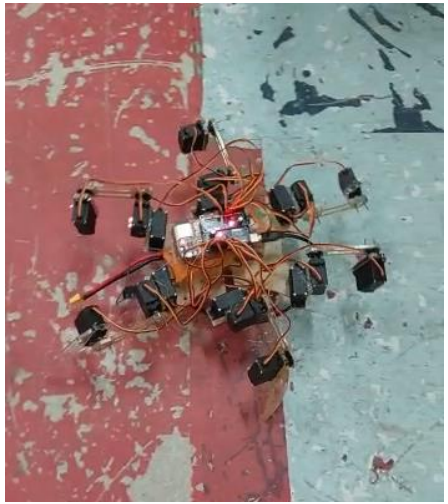


4-wheeled Autonomous Drive

- Mecanum wheels
- LSA08 line sensor
- PID control implemented

Gripper mounted on a Drive

- 4 wheeled drive
- PS4 controller
- Gripper for light weight and soft items



Hexapod Robot

- Tripod gait method
- 18 Servo Motors 3 for each leg
- Adafruit servo controller
- Maximum height is 35-37 cm

Chair of Applied Mechanics
Department of Mechanical Engineering
Technical University of Munich