Dr. Hung La, UNR

- **PROJECT:**
  - **INSPIRE: Nondestructive Data Driven Motion Planning for Inspection Robots – Year 1**
- **ACCOMPLISHMENTS:**
  - **Completed R&D of the climbing robot version 3**
  - **Making good progress to the NDE-based motion planning for automated inspection robot**
- **PLANNED ACTIVITIES:**
  - **Working on autonomous motion planning**

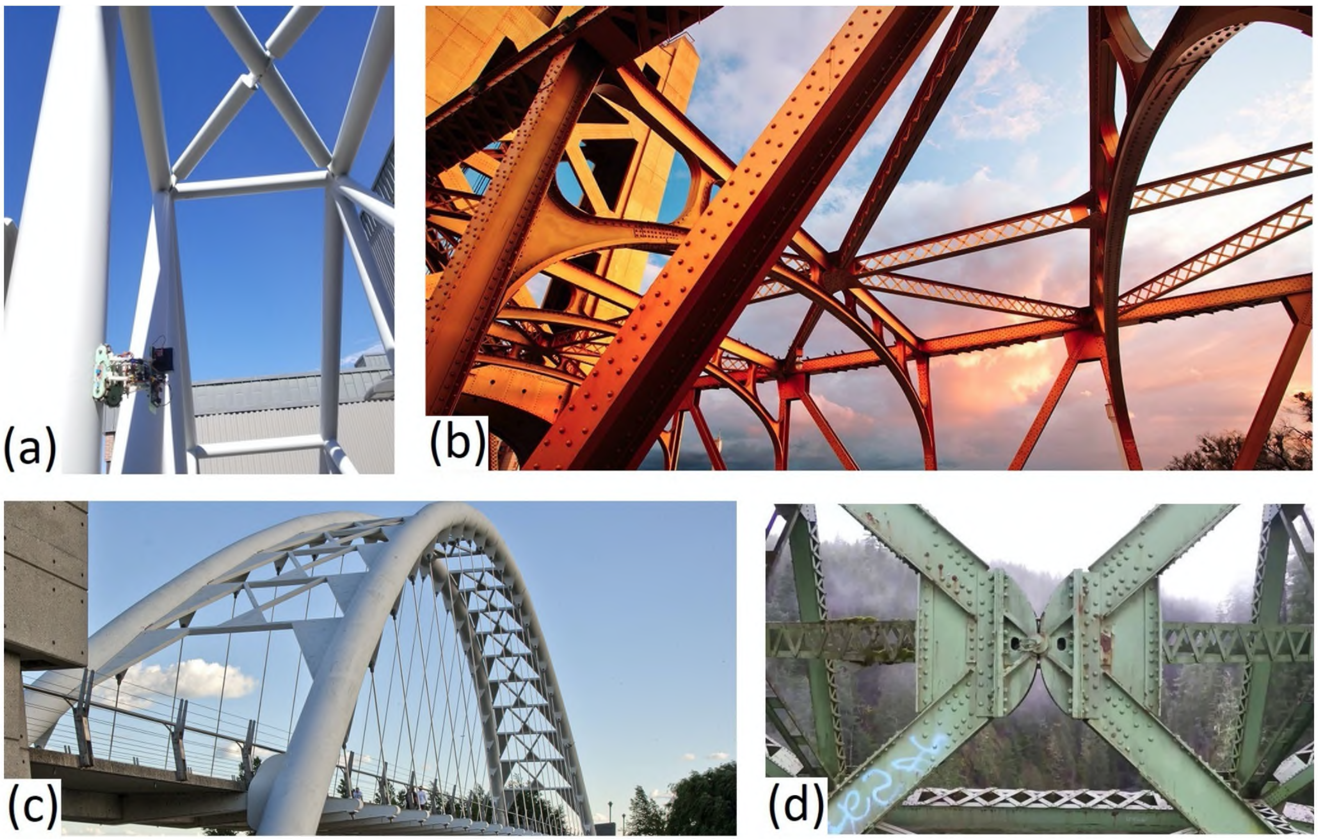
Outline

- Review of Previous Versions 1&2
- Robot Design and Development Version 3

- Robot Navigation Framework

Challenges

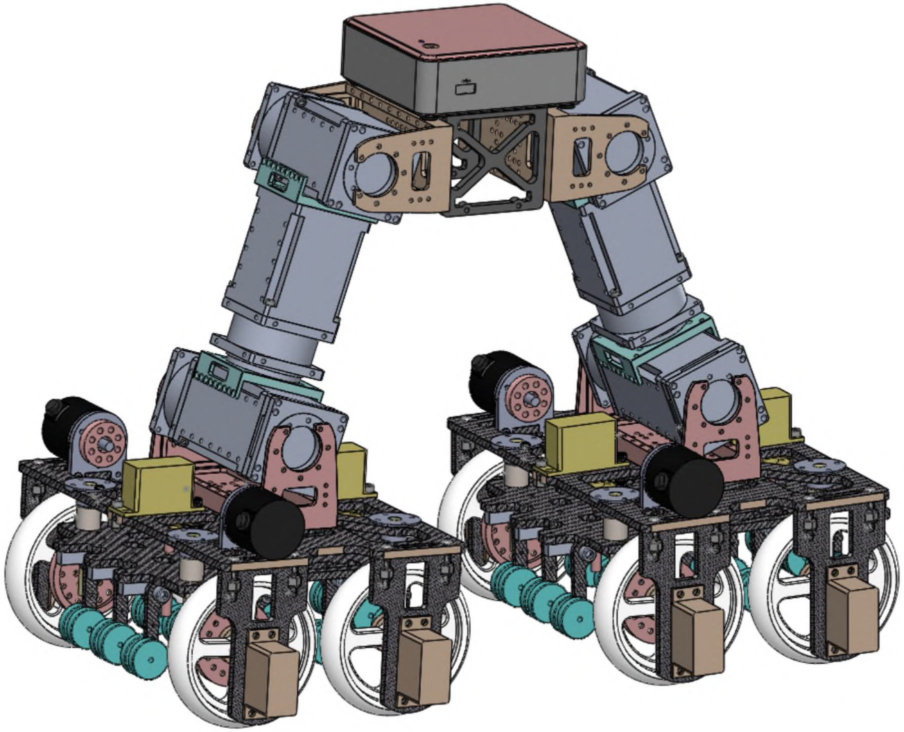
Complex structures: nuts, curving, flat, joints, etc.



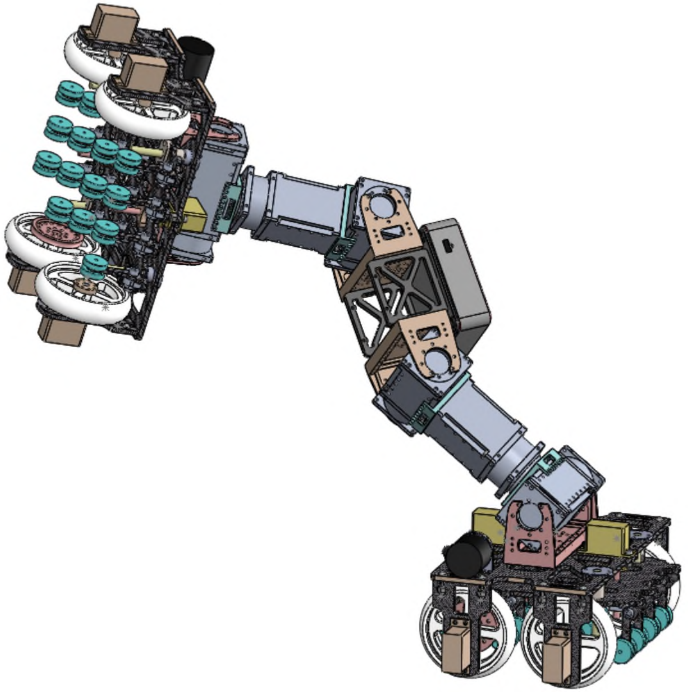
Robot Design Version 1: Mobile Robot

Robot Design Version 2: Tank-Mobile Robot

Robot Design Version 3

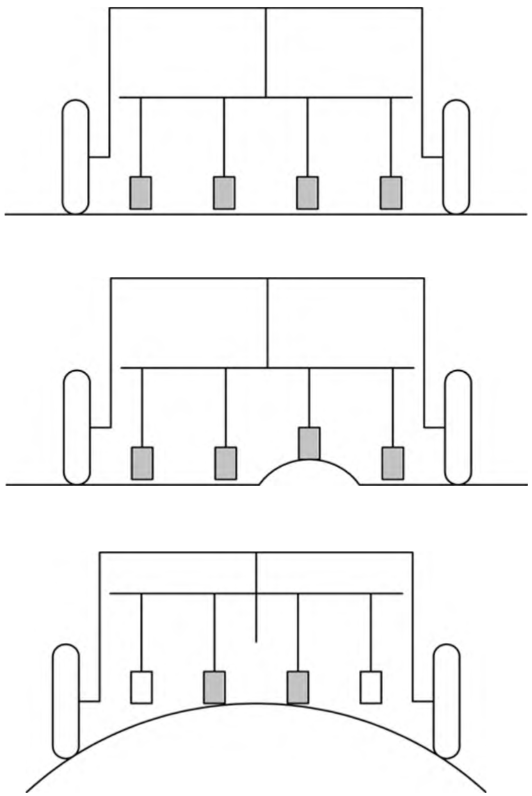


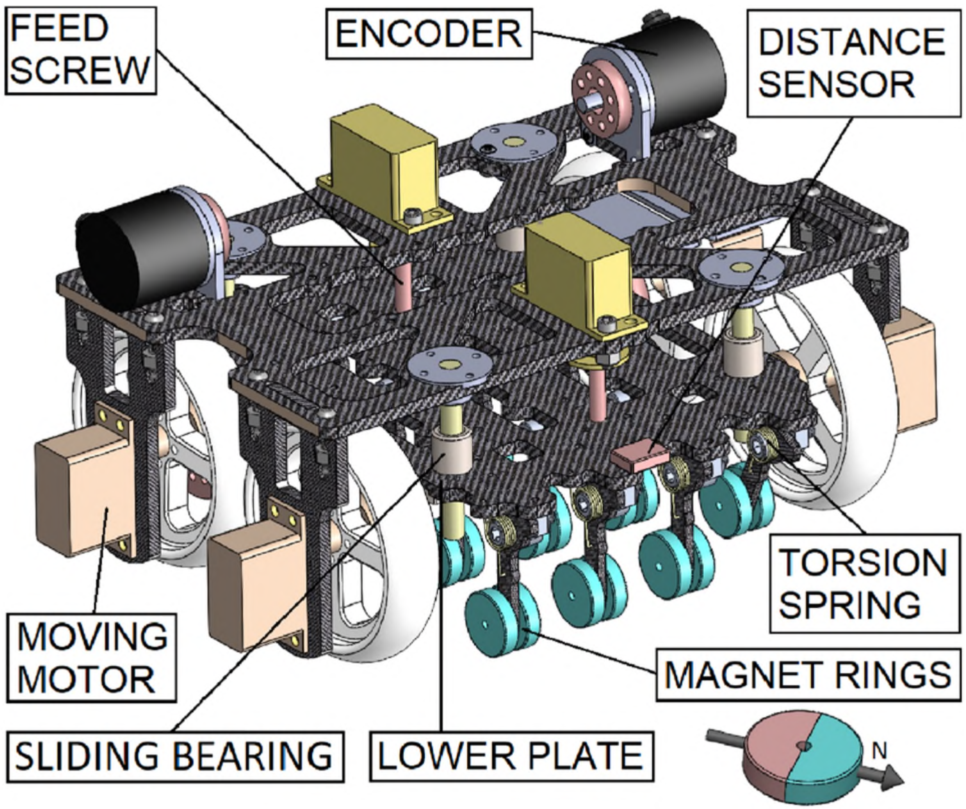
Mobile mode



Worming mode

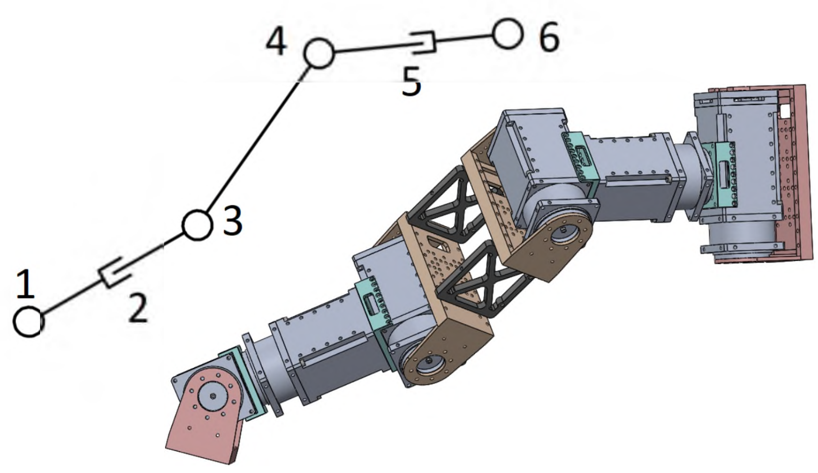
Robot Design

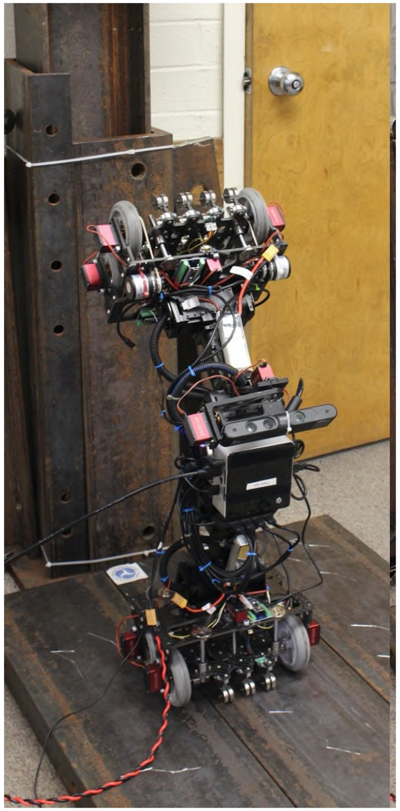


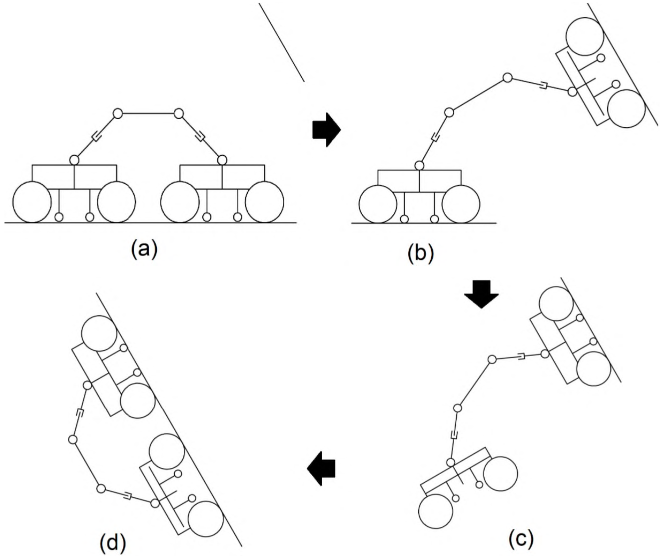


Mobile mode: *Adaptive function of magnet array on different terrains*

Robot Design



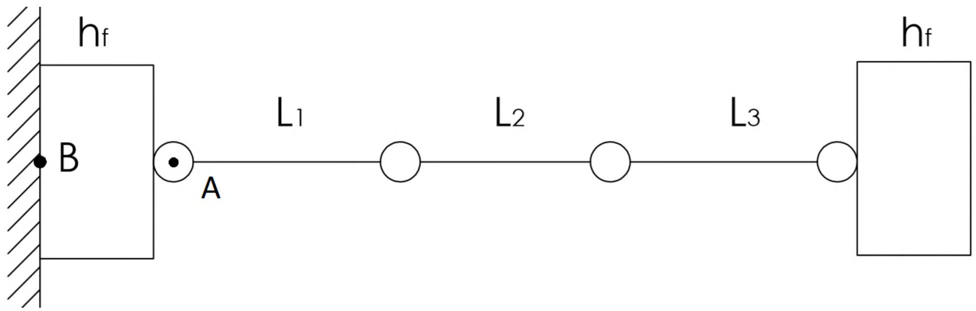




Worming mode is switched to when path planning returns no solutions

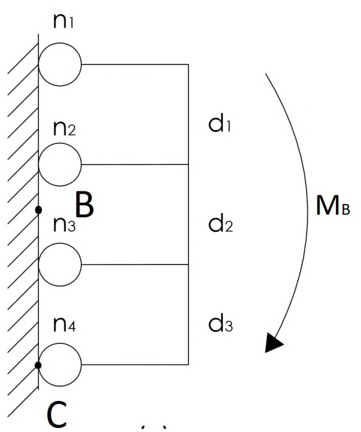
Design Analysis

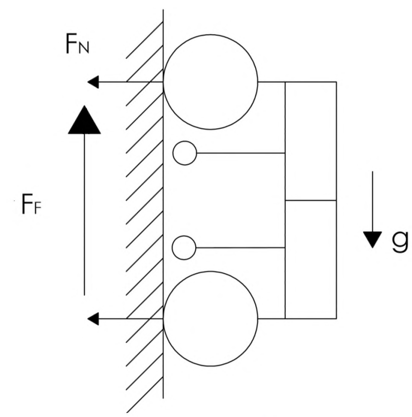
- Transforming analysis: to calculate servo power (moment and speed)



$$T_A = (\frac{1}{2}L_1)m_{L1}g + (L_1 + \frac{1}{2}L_2)(m_1 + m_e)g + (L_1 + L_2 + \frac{1}{2}L_3)m_{L3}g + (L_1 + L_2 + L_3 + \frac{1}{2}h_f)m_f g.$$

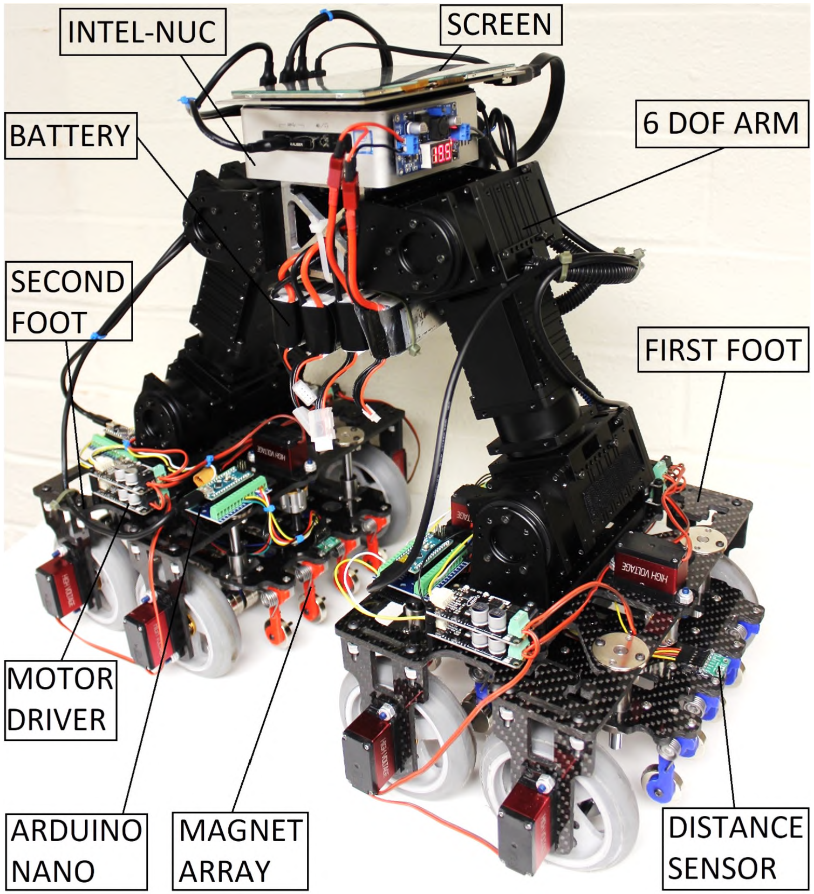
- Turn Over and Sliding Friction Analysis: to calculate magnetic force need





$$M_c = -M_B + F_{mag}n_3d_3 + F_{mag}n_2(d_2 + d_3) + F_{mag}n_1(d_1 + d_2 + d_3).$$

Robot Implementation

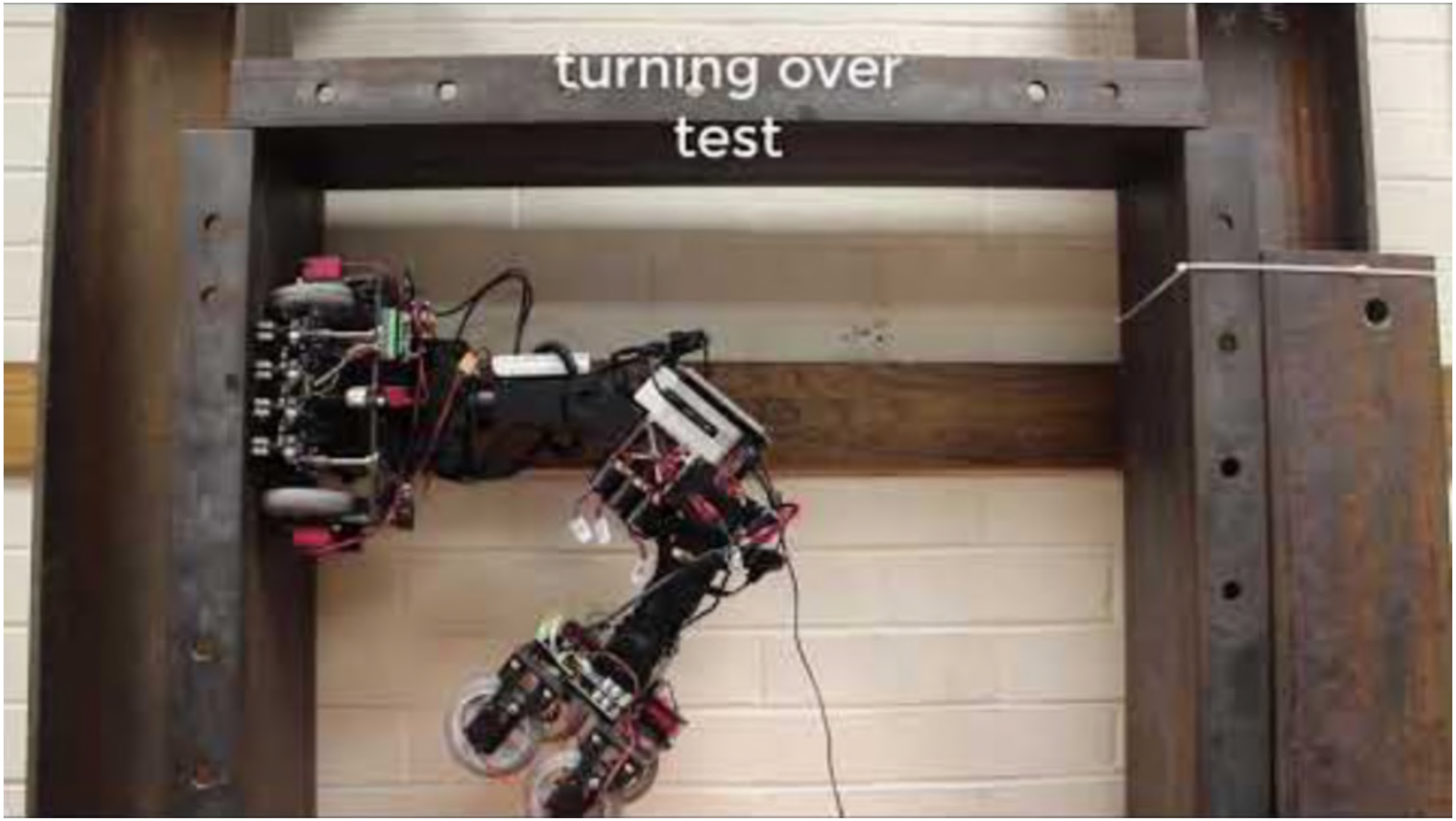


Dimension : 25x50x50cm

Weight : 11.5kg

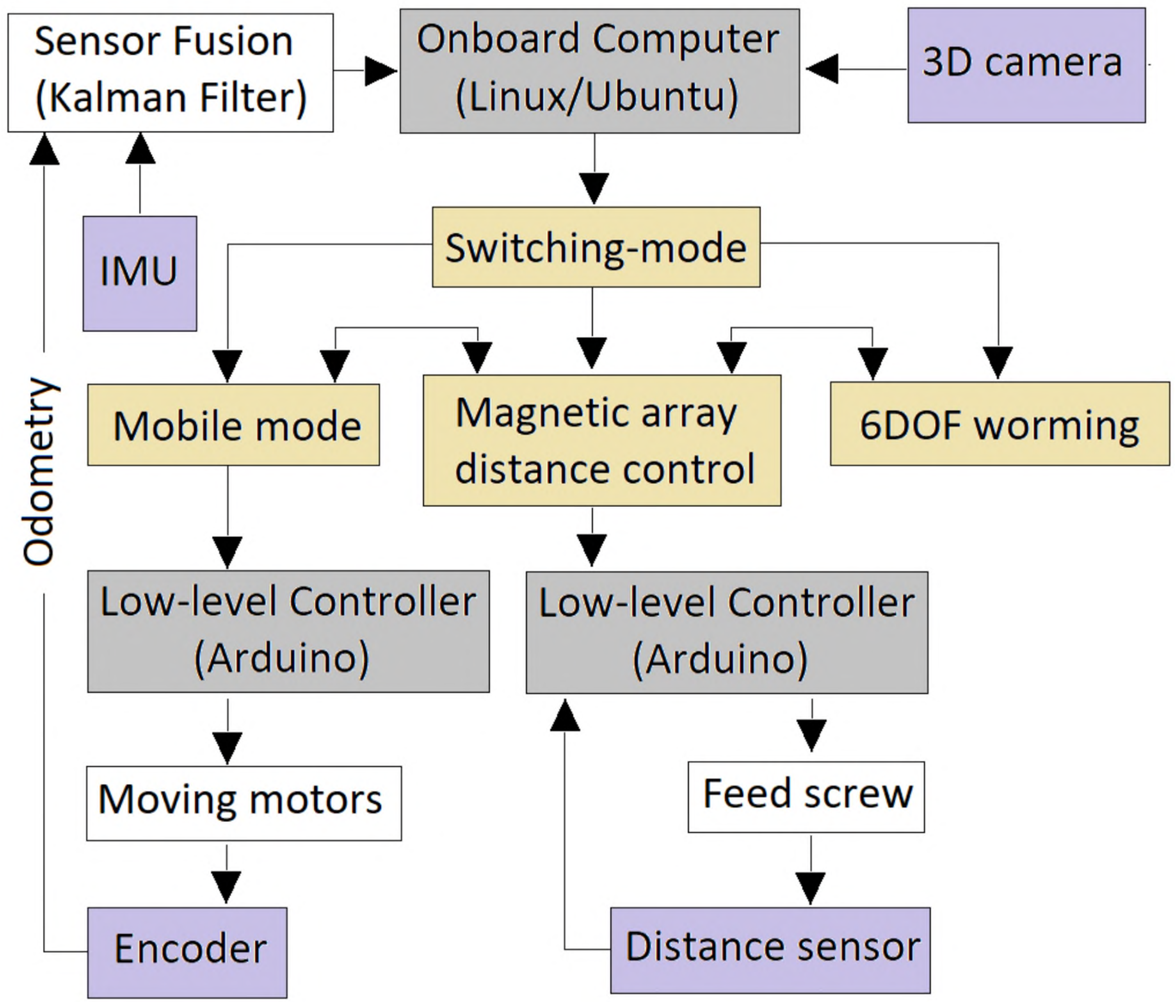
Duration: 30-40 minutes

Robot Test

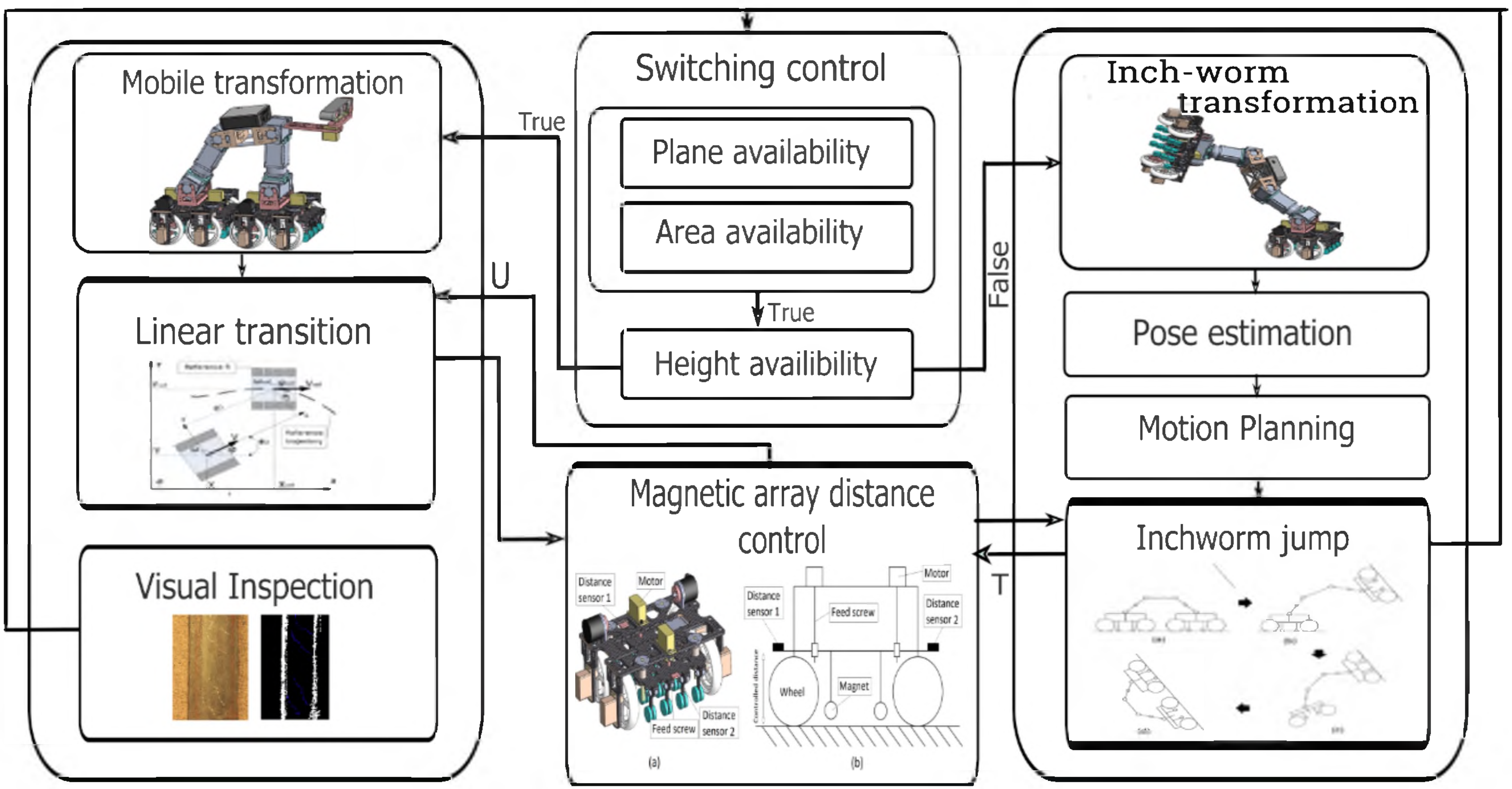


Robot Navigation Framework

Control Architecture



A control framework



Switching Control

- Enable the robot to autonomously configure into two transformations
- The control is deployed by Boolean function S:

$$S = f(S_{pa}, S_{am}, S_{hc}) = S_{pa}S_{am}S_{hc} \quad (1)$$

If S returned True, the robot will activate the mobile transformation.

Otherwise, it will go to inch-worm transformation.

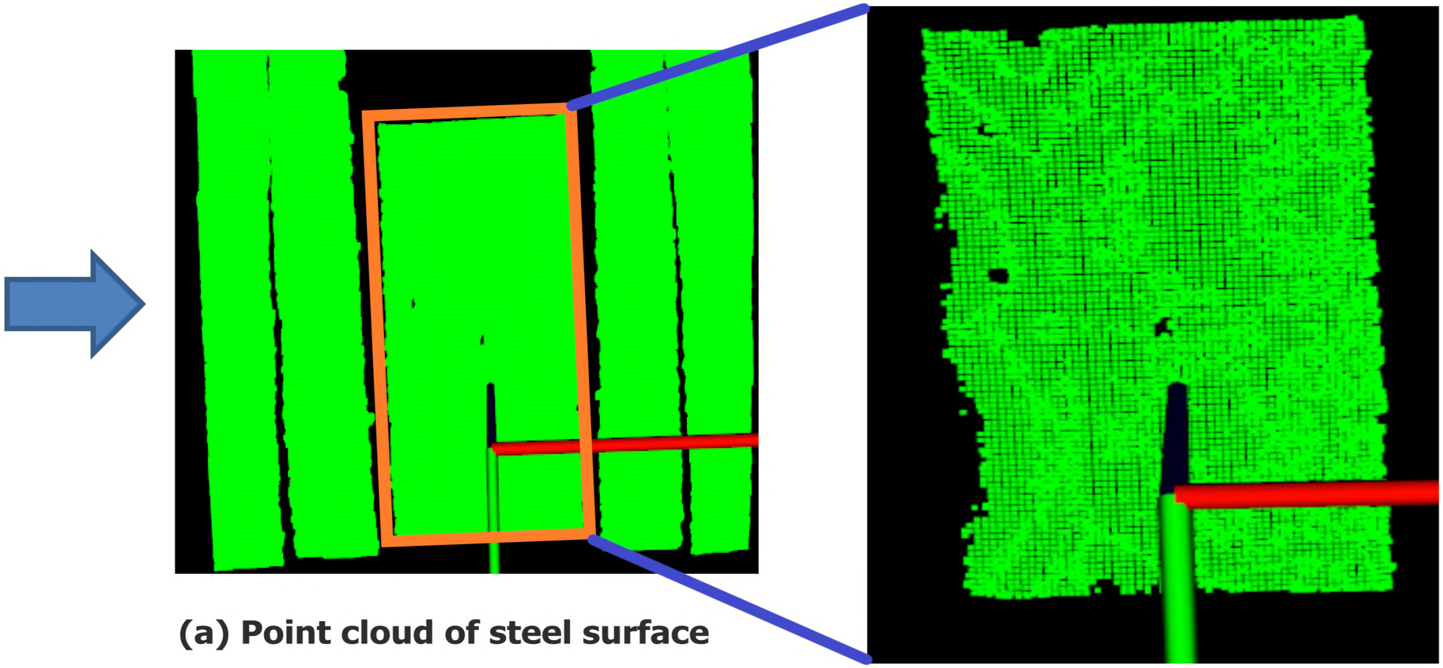
- S_{pa} is Boolean plane availability variable, which determine by a simple function Plane_Detection (based on RANSAC method)

$$\begin{cases} S_{pa} = False, \; if \; P_{cl} = \emptyset \\ S_{pa} = True, \; otherwise. \end{cases} \quad (2)$$

P_{cl} is the set of point cloud, which belong to a plane

Switching Control

- S_{pa} is Boolean plane availability variable



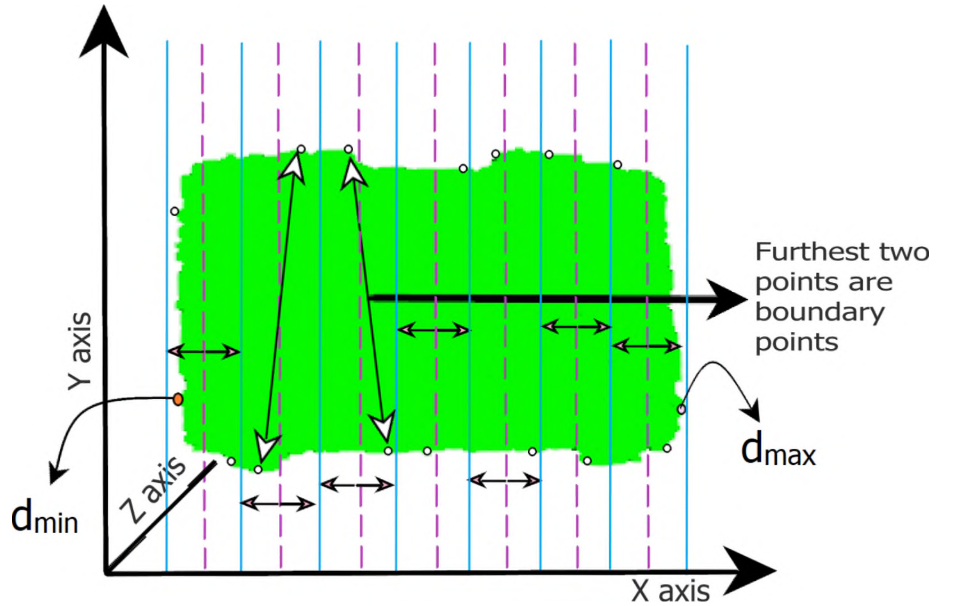
(a) Point cloud of steel surface

(b) Point cloud after extracting planar surface

Switching Control

- S_{am} is boolean area check variable, which is determined by two algorithms (1 and 2). If S_{am} returns True, there is available area for the robot to make a new move.

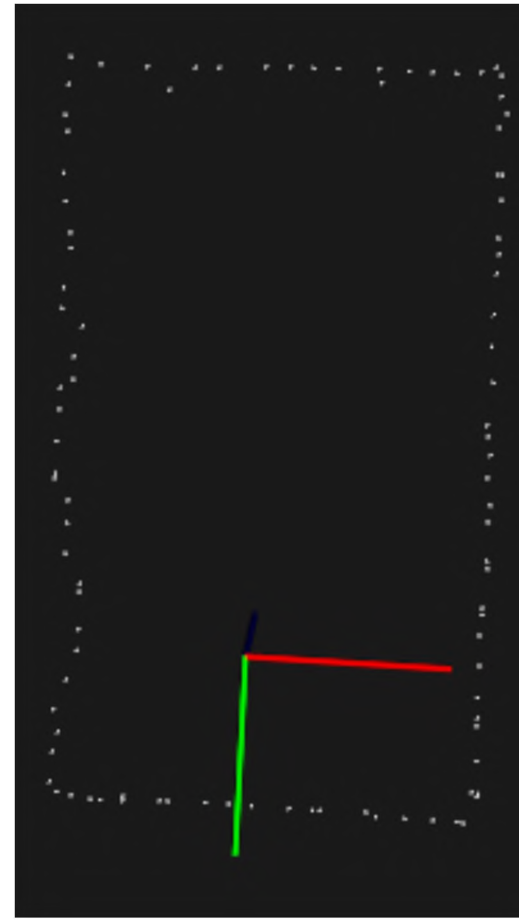
- **Algorithm 1**: estimate the boundary of a non-convex point cloud plane, based on window-based approach.
- Input: plane point cloud data
- Output: set of boundary points



Switching Control

Algorithm 1 Non-convex boundary point estimation from 3D point cloud data of steel bridges

procedure BOUNDARYESTIMATION(P_{cl}, α_s)
  $Planes = \{xy, yz, zx\}$
  $d_{min} = \forall_{i \in Planes}$ //Point along minimum value of plane i
  $d_{max} = \forall_{i \in Planes}$ //Point along maximum value of plane i
  Initialize $B_s = \{\}$
  for $p \in Planes$ **do**
    $i \rightarrow 1$
    while $sl_{p_i} < d_{max}$ **do**
      $sl_{p_i} = d_{min_p} + i * \alpha_s$
      PS_{p_i} = Set of points in range $sl_{p_i} \pm \alpha_s/2$
      $P_{cl_A}, P_{cl_B} = \underset{\forall \{P_i, P_j\} \in PS_{p_i}}{\text{argmax}} \{\|P_i - P_j\|\}$
      $B_s = B_s \cup \{P_{cl_A}, P_{cl_B}\}$
      $i = i + 1$
    end while
  end for
return B_s
end procedure



Switching Control

Algorithm 2: with plane boundary point set as input (S_{pa}), this algorithm will estimate the plane meet the area requirements for the robot feet. If the area is satisfied, S_{am} is returned True and the algorithm also outputs the plane Pose.

The procedure:

- (1) A rectangle whose area is enough to contain the robot feet is put at the center of the boundary set.
- (2) 8 points of the rectangles (4 corners and 4 middle points) are estimated whether it lies inside the boundary or not.
- (3) The method: iterative compare each point (belong to the rectangle) to the n closest points, which belongs to the boundary set.

Switching Control

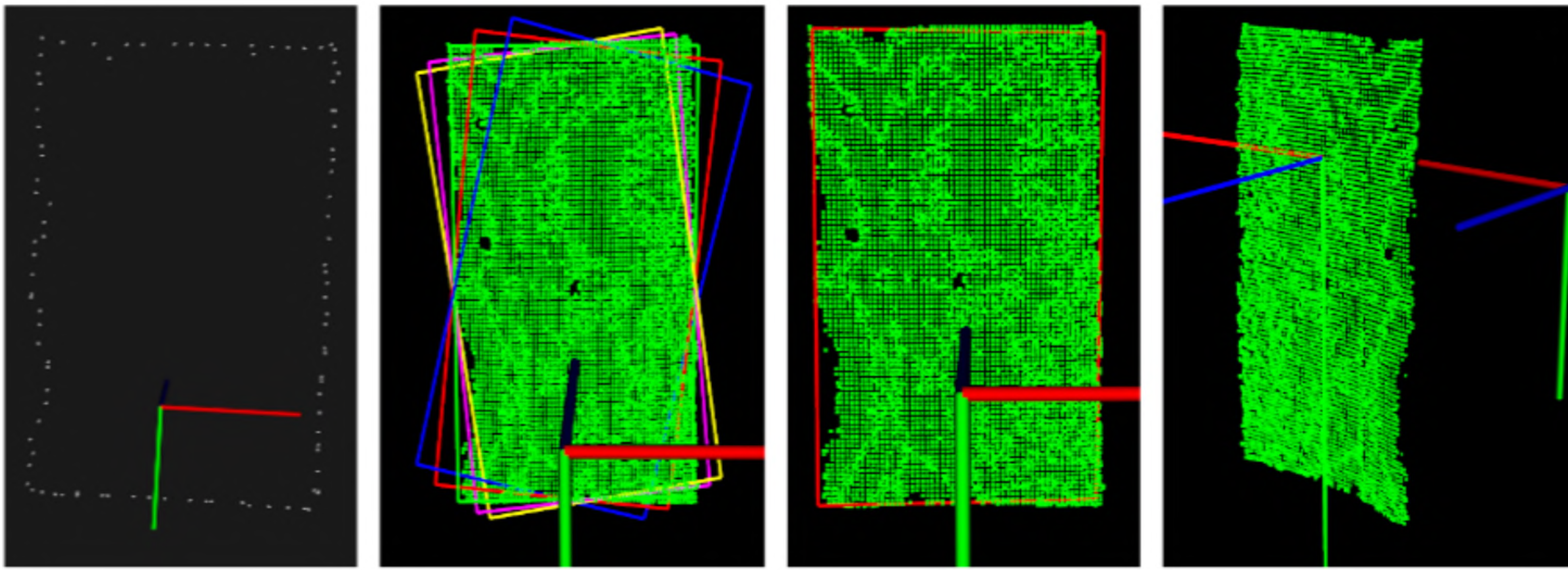
Algorithm 2:

The procedure:

- (4) To determine the side of the rectangle points comparing with the boundary set, the distance from each point to the center point of the boundary set is used.
- (5) Due to the error of point cloud data, a tolerance *t is* considered
- (6) If the 8 points are estimated lying inside the boundary → it means the area is enough for the robot move.
- (7) Calculate the pose of the plane

Switching Control

Algorithm 2:



(a) (b) (c) (d)

Switching Control

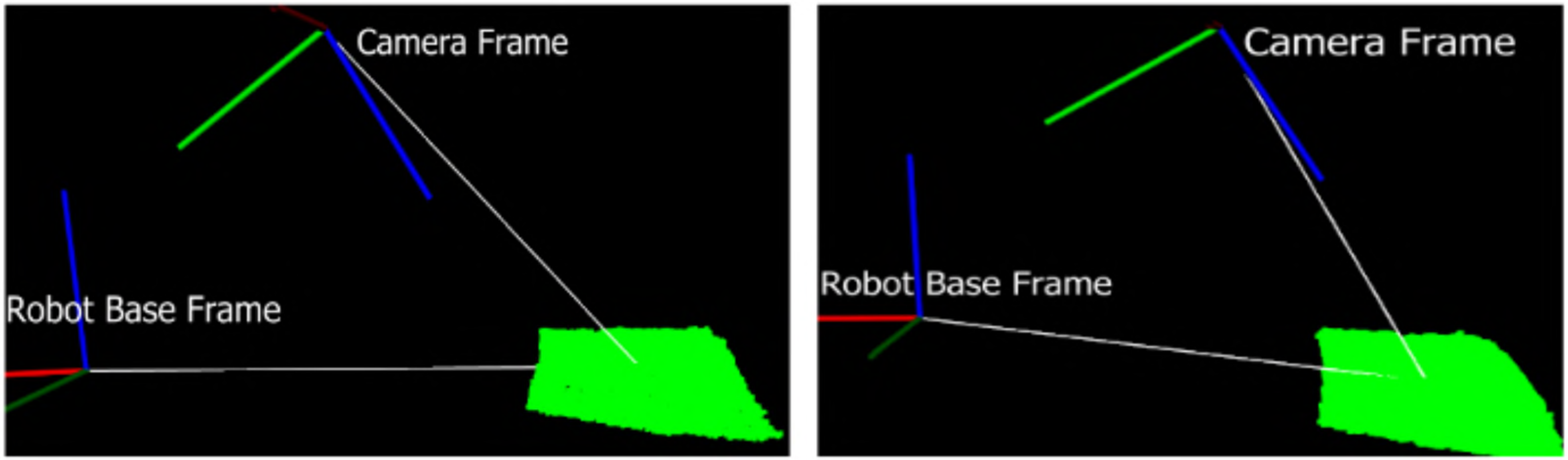
- S_{hc} is boolean height availability variable, which returns True if the plane height is equal to the robot base's height. Otherwise, it returns False
- The plane height is calculated based on the center point of the point cloud respective to the camera, then the transformation matrices multiplication is done to convert to the robot's base.

$$P_{C_{f_{rb}}} = T_{f_{rb} f_c} P_{C_{f_c}},$$

Where ($P_{Cfrb,\ PCfc)}$ is coordinate of the centroid C_c in the camera frame and the robot base frame, respectively. T_{frbfc} is the transformation matrix from the camera frame to the robot base frame

Switching Control

-



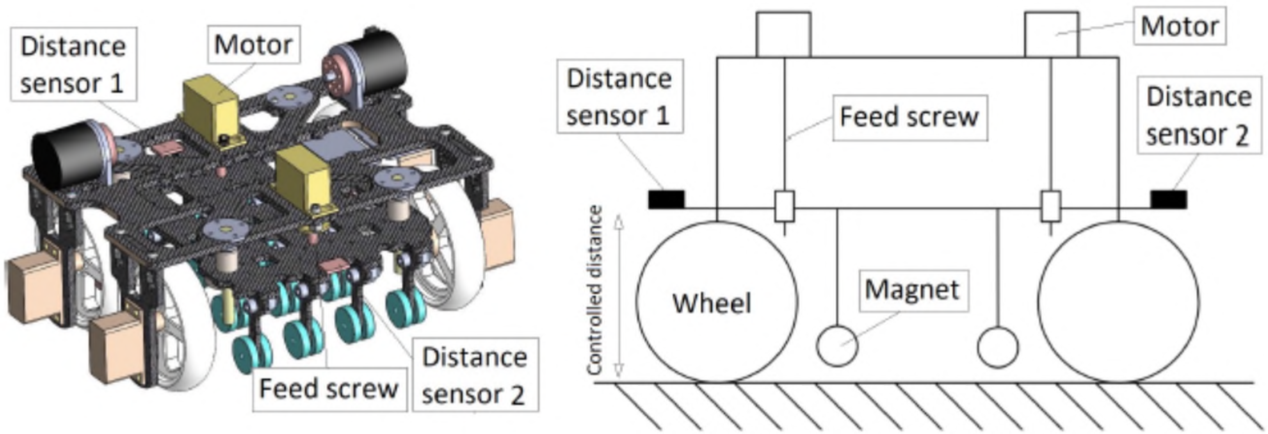
(a) (b)

Algorithm 2 Area Checking from the plane surface boundary points and Pose Calculation

procedure AREA($B_s, C_{P_{cl}}, n\vec{P_{cl}}, w, l, t, S_{am}$)
    N_{clos} = Find n closest points to $C_{P_{cl}}$ from B_s
    for $N_i \in N_{clos}$ **do**
        $R = \{\}$, //Estimated rectangle corner points
        $\vec{e_{x_i}} = N_i - C_{P_{cl}}$
        $\vec{e_{z_i}} = n\vec{P_{cl}}$
        $\vec{e_{y_i}} = \vec{e_{x_i}} \times \vec{e_{y_i}}$
        $k_w = \frac{w}{|\vec{e_{x_i}}|} e_{y_i}$ and $k_l = \frac{b}{|\vec{e_{y_i}}|} e_{x_i}$,
        $\{R_1, R_2\} = \{N_i + k_w, N_i - k_w\}$
        $R = R \cup \{R_1, R_2\}$
        $R = R \cup \{R_1 + k_l, R_2 + k_l\}$
        $M = \forall_{r_i \in R}\{\frac{r_i + r_{i+1}}{2}\}$
        $R = R \cup M$
        $S_{am} = True$
        for $r_i \in R$ **do**
            Q_i = Find m closest points to r_i
            $d_{r_i} = \|d_{r_i}, C_{P_{cl}}\|$ and $d_{Q_i} = \|Q_i, C_{P_{cl}}\|$
            $S_i = (d_{r_i} < d_{Q_i}) \vee (\frac{d_{r_i} - d_{Q_i}}{d_{r_i}} < t$
            $S_{am} = S_{am} \wedge S_i$
        end for
        Pose = {Orientation, Position}
        if $S_{am} ==$ True **then**
            R_c = Centroid of R
            Orientation = $(\vec{e_{x_i}}, \vec{e_{y_i}}, \vec{e_{z_i}})$
            Position = $(x_{R_c}, y_{R_c} - l/4, z_{R_c})$
            return Pose
        end if
    end for
    return False
end procedure

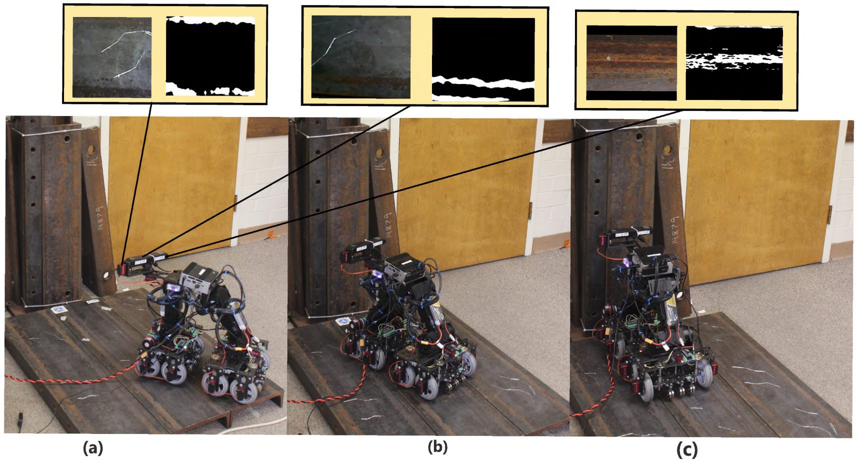
Magnetic Array Distance Control

- Touched mode: generates a huge force for the robot can stand on one foot and make the inch-worm jump.
- Untouched mode: the magnetic array keeps a distance of 1 mm to the steel surface.



Results

- The switching control mechanism with the unique features for two transformation enhances the flexibility and navigation and inspection.
- Two algorithms work effectively to estimate available non-convex surface for navigation using area, plane and height availability.
- Propose a flexibility control for magnetic adherence distance controller which help the robot works smoothly on both mobile and inch-worm transformation.



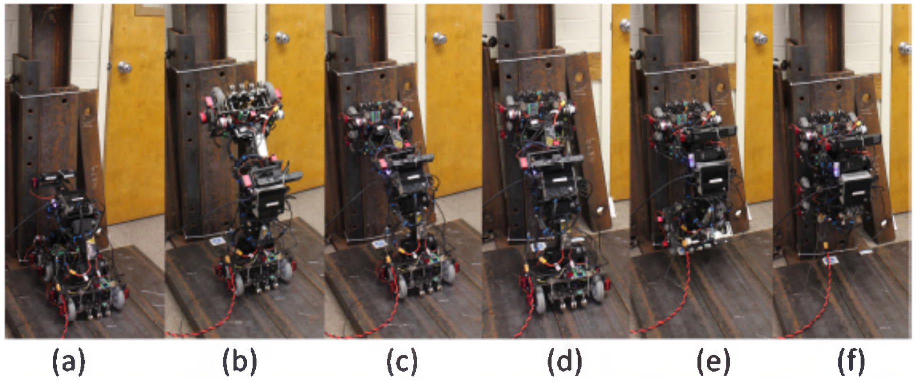
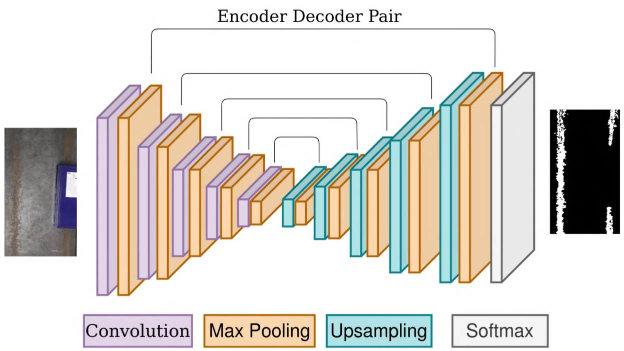
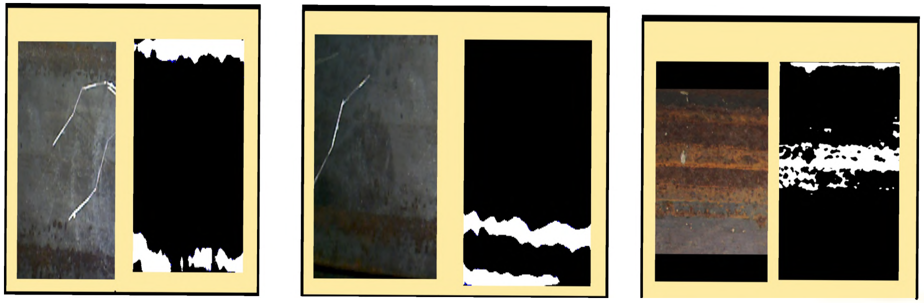


Fig. 18: *Worming transformation*: a) magnetic array of second foot touches the base surface b) first foot moves to convenient point c) first foot reach target pose and touches the second surface d) magnetic array of second foot is released e) and f) second foot moves to target pose

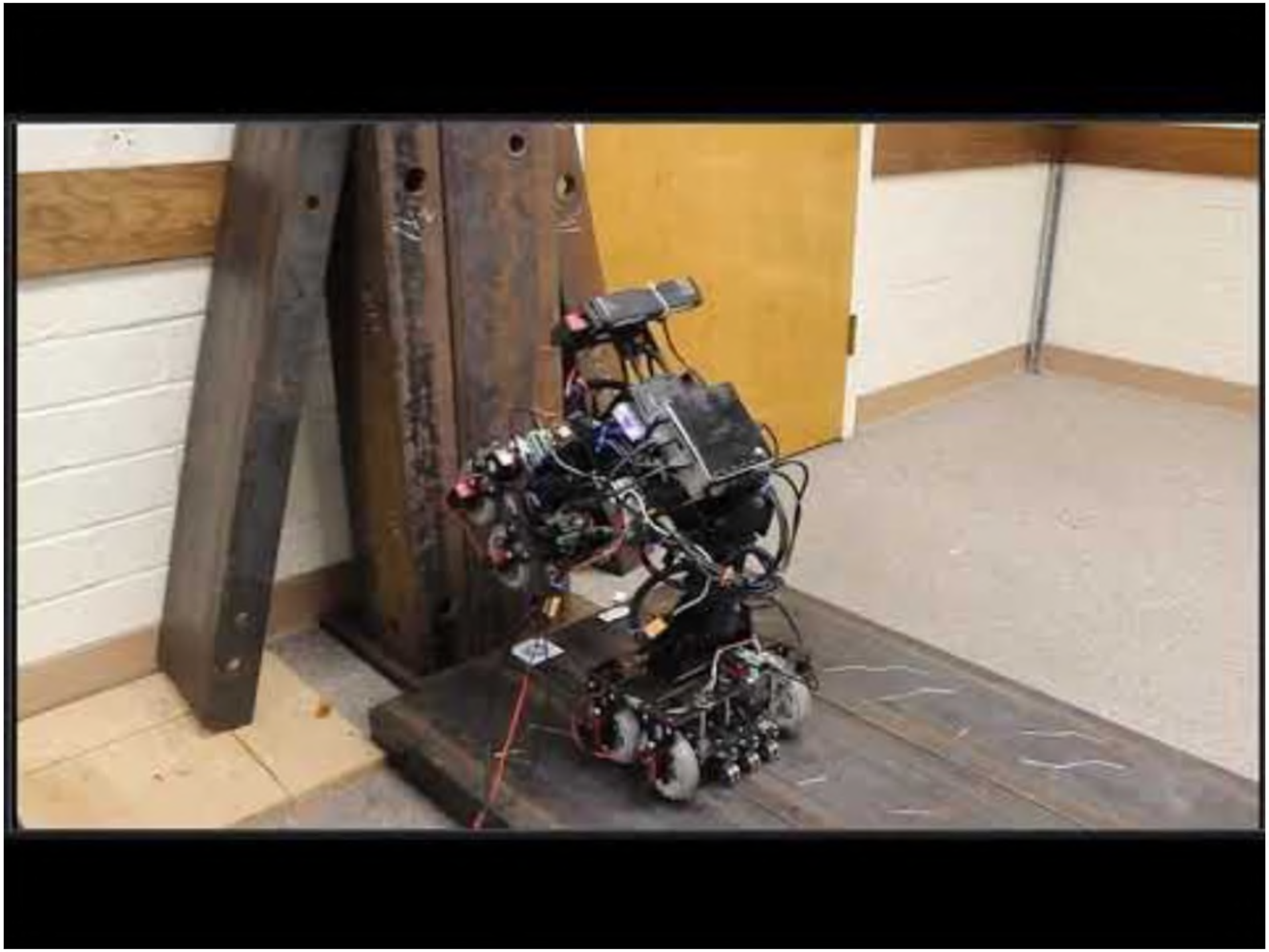
Inspection Results

- The CNN architecture segments images into defect and healthy regions.
- The input images are passed through five encoder layers followed by five decoder layers. Each encoder performs a 7x7 convolution operation to extract defect feature maps.
- The convolution operation is followed by a batch normalization operation and ReLU activation.
- Five encoder layers are utilized for feature extraction. Once the last encoder layer is reached, the resultant feature maps enter the decoder portion of the network, where they are up-sampled using bi-linear interpolation in each decoder layer.
- The CNN architecture is pre-trained on 3000 steel images containing severe defects (e.g. corrosion). For the hyperparameter optimization, we used the ADAM optimizer with a learning rate of 0.0001
- The network was trained for 150 epochs in this experiment.





Video Demonstration



Future Work

- The motion planner RRTConnect (Moveit) was not robust in calculating the inverse kinematics and generating the trajectory for the robot → it needs a robust motion planner for this.
- Optimization for inch-worm transformation is necessary to ease the inch-worm jump.
- Navigation for mobile transformation in varied steel bridge structures

PRODUCTS: Conferences

1) H-D. Bui, S. Nguyen, U-H. Billah, C. Le, A. Tavakkoli, and H. M. La. Control Framework for a Hybrid-steel Bridge Inspection Robot**. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),** October 25-29, 2020, Las Vegas, NV, USA. (Accepted)

1) S. Nguyen, A. Pham, C. Motley, H. M. La. A Practical Climbing Robot for Steel Bridge Inspection. **IEEE International Conference on Robotics and Automation (ICRA)**, May 31 - June 4, 2020, Paris, France.

2) A. Singandhupe, H. M. La, T. D. Ngo, and V. Ho. Correntropy Iterative Closest Point Algorithm for Registering Point Clouds in Noisy Environment. **IEEE/SICE International Symposium on System Integration (SII)**, January 14-16, 2021, Iwaki, Japan. (Submitted)

PRODUCTS: Journals

1) U-H. Billah, H. M. La, and A. Tavvakolli. Deep Learning Based Feature Silencing for Accurate Concrete Crack Detection. **Sensors**, August 2020.

2) H. Ahmed, H. M. La, and N. Gucunski. Review of Non-Destructive Civil Infrastructure Evaluation for Bridges: State-of-the-art Platforms, Sensors and Algorithms. **Sensors**, July 2020.

3) R. Konda, H. M. La, J. Zhang. Decentralized Function Approximated Q-Learning in Multi-Robot Systems For Predator Avoidance. **IEEE Robotics and Automation Letter**, July 2020.

4) S. Nguyen and H. M. La*. A Climbing Robot for Steel Bridge Inspection. Submitted to **Journal of Intelligent & Robotic Systems** (Under review).

5) N. Harris, S. Louis, S. Liu, H. M. La. Genetic Algorithm Based Route Optimization for Robotic Bridge Inspection. Submitted to **Journal of Engineering Applications of Artificial Intelligence** (Under review).

6) H. Ahmed, H. M. La, and K. Tran. Rebar Detection and Localization for Bridge Deck Inspection and Evaluation using Deep Residual Network. Submitted to **Journal of Automation in Construction** (Under review).

Project Members

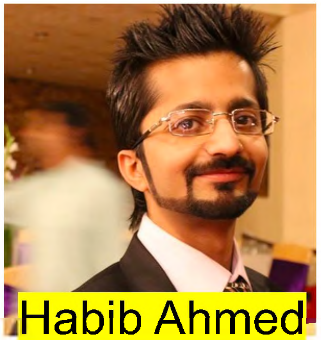
Principal Investigator:


Dr. Hung La

PhD Students


Son Nguyen


Hoang Bui


Habib Ahmed

Master Students


Hafsa Billah


Cadence Motley