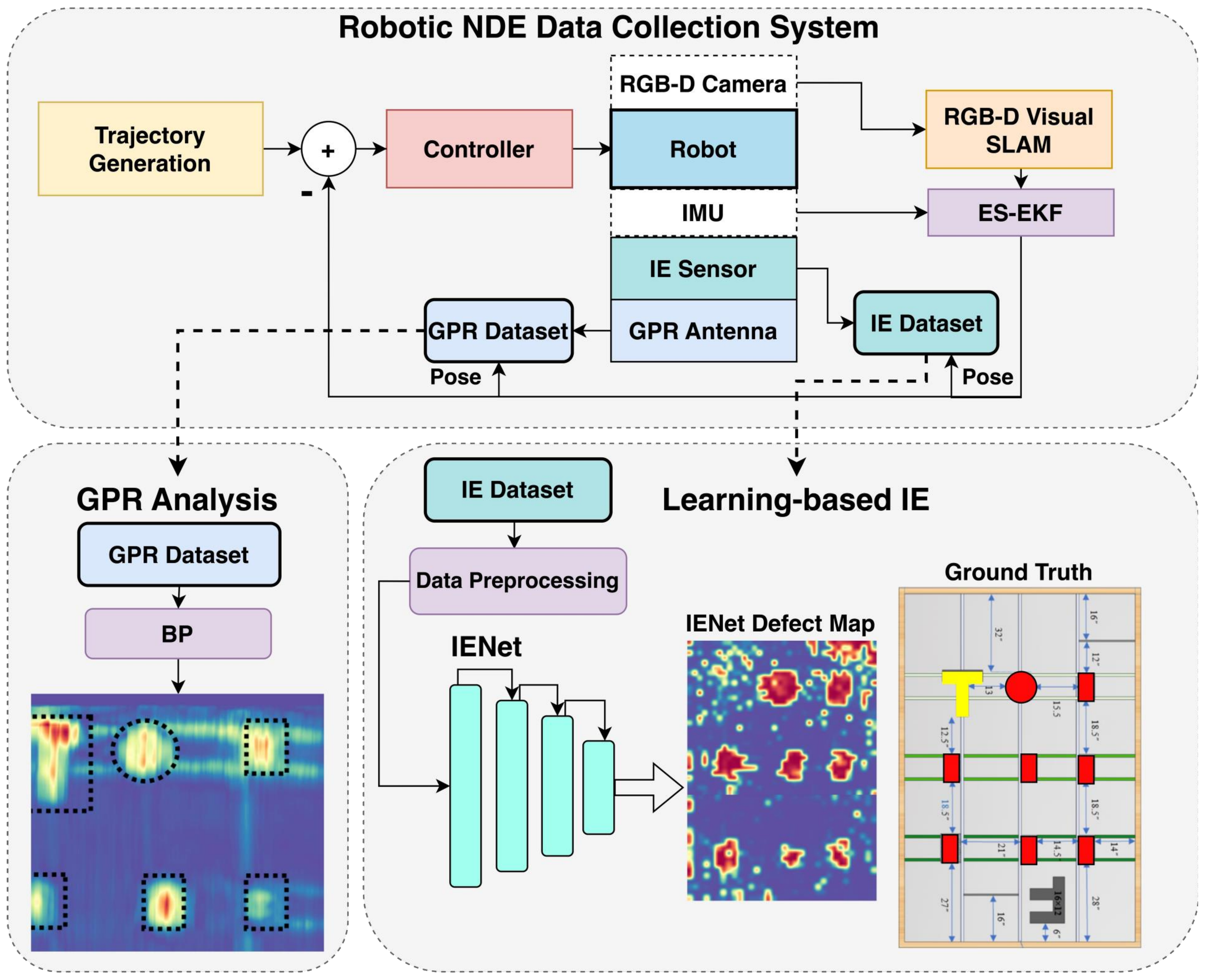
ICRA2024 YOKOHAMA | JAPAN

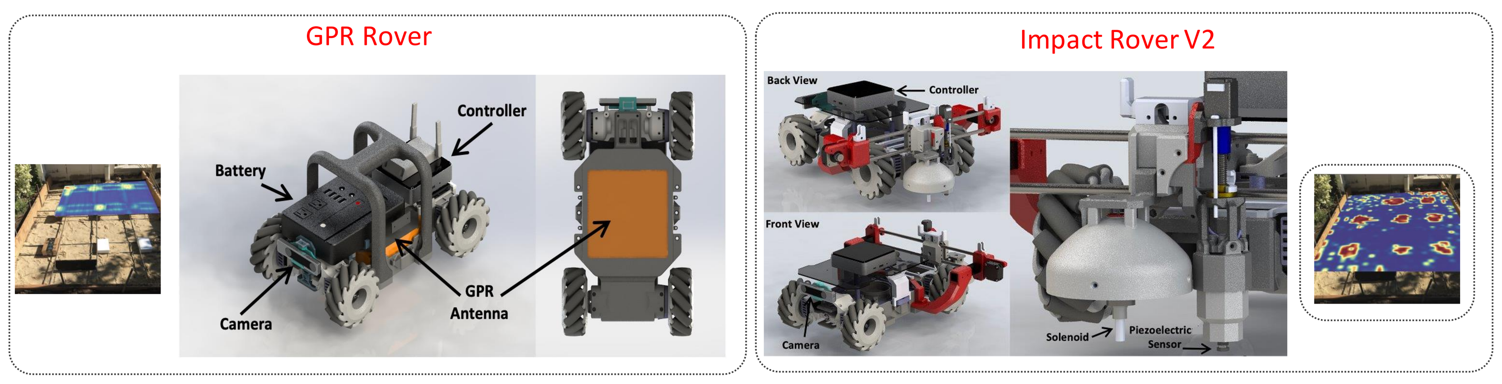
Robotic Inspection and Subsurface Defect Mapping using Impact-Echo and Ground Penetration Radar

Ejup Hoxha[1,2] Dr. Jinglun Feng[1,3] Diar Sanakov[4] Dr. Jizhong Xiao[1]
[1]The City College of New York
[2]Amazon, Inc.
[3]Honda Research Institute USA, Inc.
[4]New York University

Overview

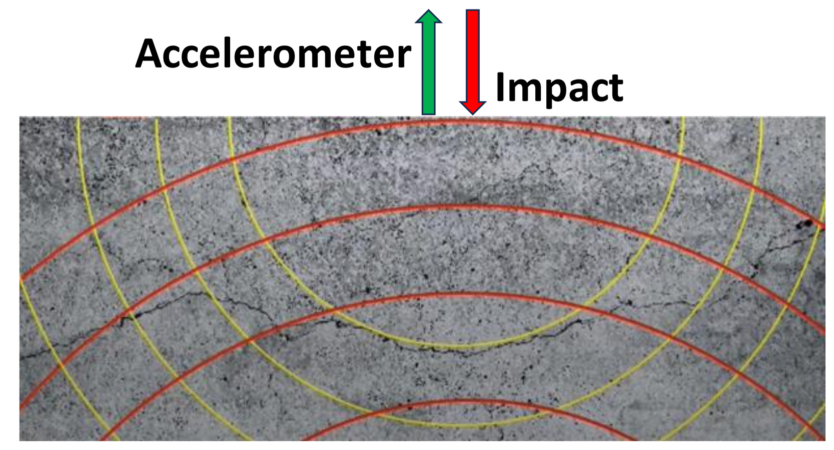
- Introduce two robots for concrete infrastructure inspection:
  - GPR Rover, for ground penetration radar automated data collection.
  - Impact Rover V2, for impact-echo automated data collection.
  - Sensor fusion between IMU and Camera data for localization.
  - Feedback control and coverage path planning module.
- GPR naïve back projection and migration for subsurface mapping.
- Learning-based Impact-Echo subsurface defect mapping.

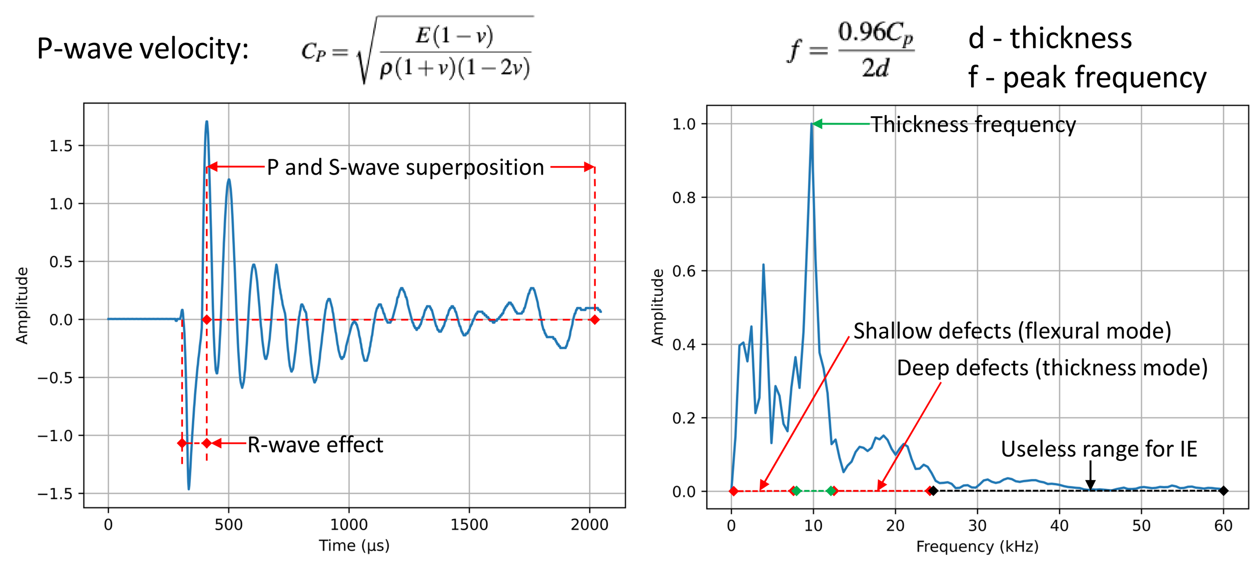




Impact Echo Principle

- A mechanical impact is generated in the surface, causing the impact wave to propagate in the concrete structure.
- P-wave is reflected from boundary conditions.
- The reflected wave is of most importance for impact-echo.
- The time of flight is directly related to the geometry of the structure. It can represent either the depth of the concrete slab, or a defect.

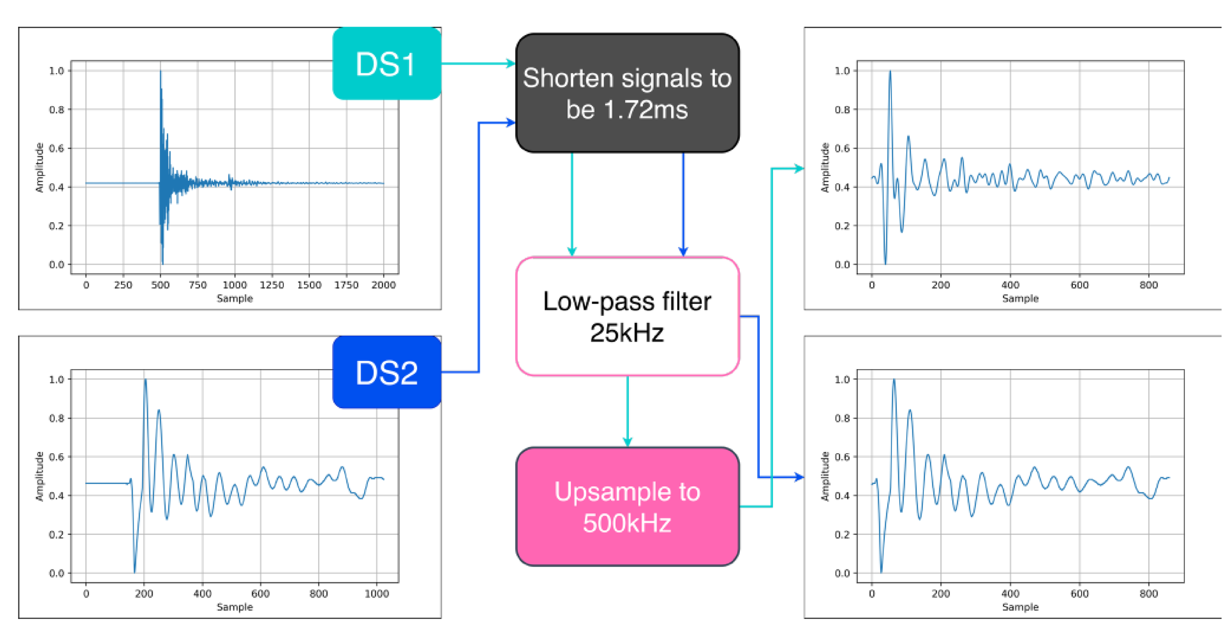




P-wave velocity: $C_P = \sqrt{\frac{E(1-\nu)}{\rho(1+\nu)(1-2\nu)}}$ $f = \frac{0.96C_p}{2d}$

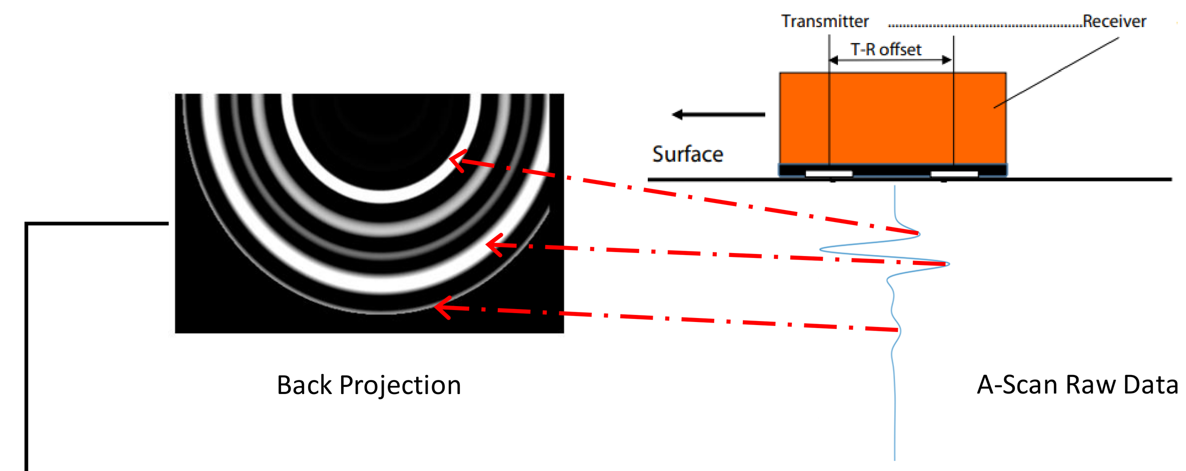
Impact Echo Data Preprocessing

- **Dataset1**
  - With labels.
  - Split for training / validation / testing.
- **Dataset2**
  - Without labels, ground truth schematics only.
  - Used for qualitative validation only.
- **Preprocessing:**
  - 1 - Shorten to 1.72ms
  - 2 - LPF (cutoff=25kHZ).
  - 3 - Upsample to 500kHz.



GPR Back Projection

Back-projection (BP) is a naïve GPR data processing method

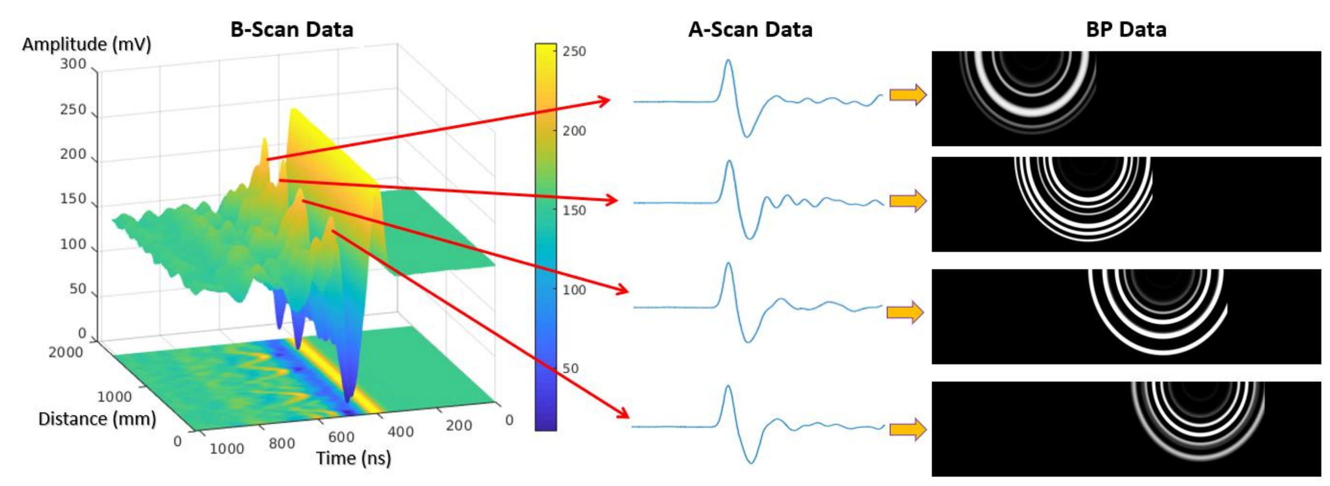


$$\forall A_q^k \in B_k,\ (x-a)^2 + (y-b)^2 = (a_t * t)^2,\ y < 0$$

$A_q^k = \{a_t | t = 1, \dots, n_q\}$ the q-th A-scan measurement in k-th B-scan data t: traveling time

$B_k = \{A_q^k | q = 1, \dots, n_k\}$: the k-th B-scan consisting of n_k A-scans

n_q: the total samples in a A-scan measurement



Results: Impact Echo and GPR

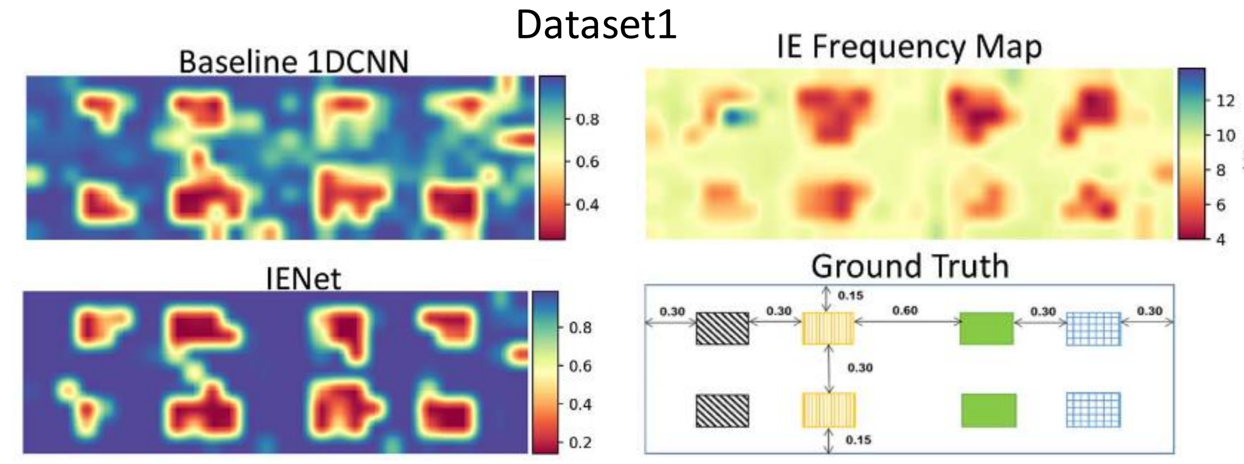
Impact Echo Results

COMPARISON BETWEEN DIFFERENT MODELS IMPLEMENTED ON THIS PAPER

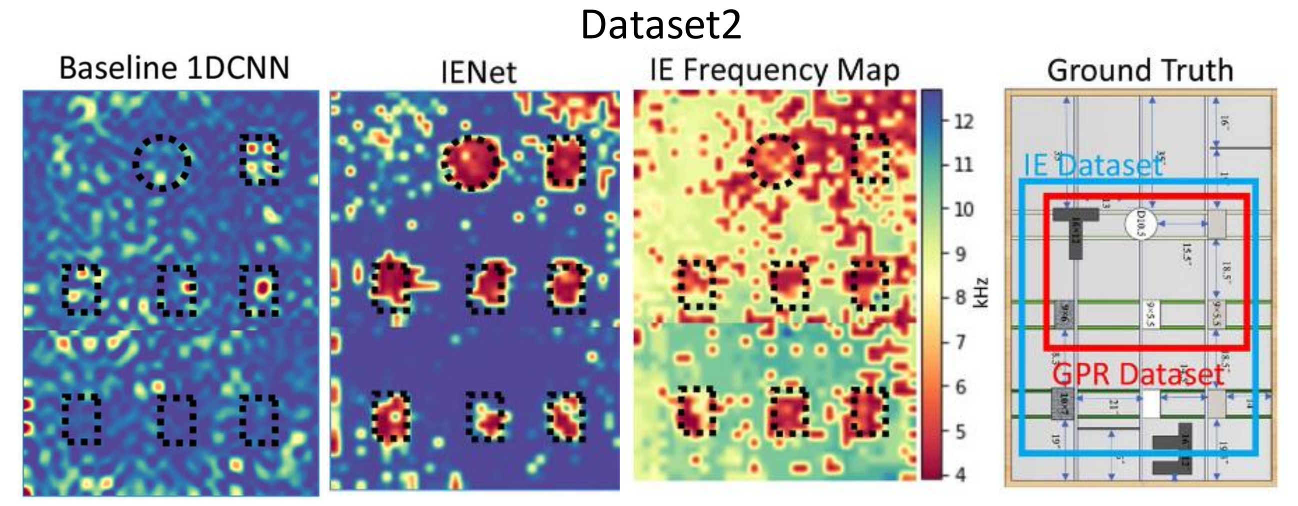
Model	Acc. %	Defect Acc.%	Solid Acc.%	DS2 Map
BL-1DCNN	88.65	70.0	95.0	Fail
BL-BiLSTM	80.03	61.0	87.0	Fail
1DCNN	82.93	75.0	85.64	Fail
Dp-1DCNN	84.13	**84.38**	84.04	Weak
BiLSTM	85.71	60.93	94.15	Weak
CRNN	88.09	68.75	94.68	Good
IENet1024	87.69	60.93	**96.08**	Fail
IENet	**90.48**	75.0	95.74	**Best**

Quantitative results are generated using test data from DS1. "DS2 MAP" column states how well model identifies defects on DS2, shown in fig. 9. Bl means baseline model

Dataset1

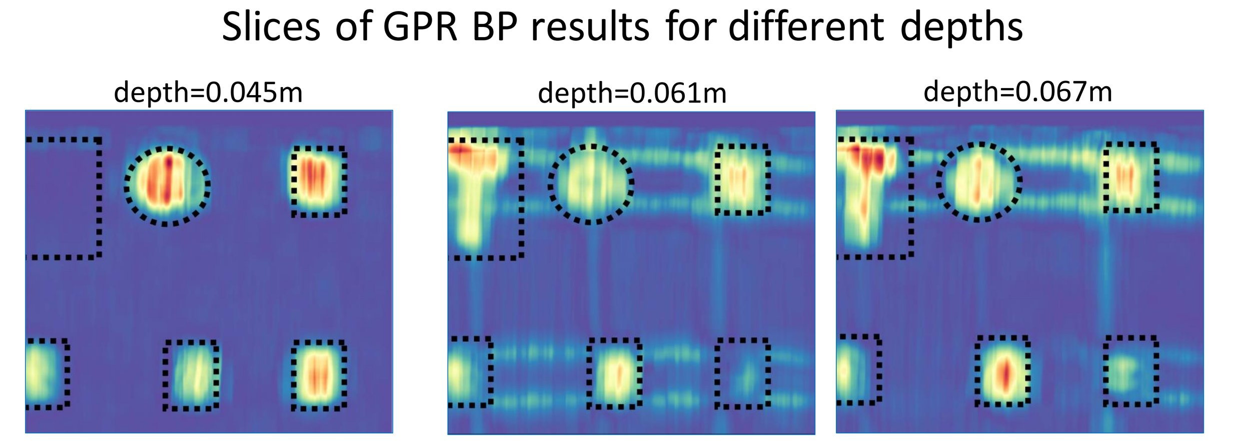


Dataset2



GPR Results

Slices of GPR BP results for different depths



Warped Defect Maps Over Dataset2 Ground Truth

GPR Results Map Ground Truth Image IENet Results Map

ES-EKF Based Visual-Inertial Fusion

Prediction Step (IMU)

1. Propagate nominal and error state:

$$\begin{bmatrix} p_{k+1} = p_k + v_k \cdot \Delta t + \frac{1}{2}\Delta t^2 \cdot (R(a_m - a_b) + g) \\ v_{k+1} = v_k + \Delta t \cdot (R(a_m - a_b) + g) \\ q_{k+1} = q_k \otimes q\{(\omega_m - \omega_b)\Delta t\} \\ a_b = a_b \\ \omega_b = \omega_b \\ g = g \end{bmatrix}$$

$$\begin{bmatrix} \delta p_{k+1} = \delta p_k + \delta v_k \cdot \Delta t \\ \delta v_{k+1} = \delta v_k + \Delta t \cdot (-R[a_m - a_b]_x \cdot \delta\theta + g) + V_i \\ \delta\theta_{k+1} = R^T\{(\omega_m - \omega_b)\Delta t\}\delta\theta - \delta\omega_b \Delta t + \Theta_i \\ \delta a_b = \delta a_b + A_i \\ \delta\omega_b = \delta\omega_b + W_i \\ \delta g = \delta g \end{bmatrix}$$

2. Propagate covariance:

$$\hat{\Sigma}_{k+1} = F \cdot \Sigma_k \cdot F^T + G \cdot Q \cdot G^T$$

Correction Step (Camera Update)

1. Calculate Kalman gain:

$$K_{k+1} = \hat{\Sigma}_{k+1} {H_{k+1}}^T (H_{k+1}\hat{\Sigma}_{k+1}{H_{k+1}}^T + R_{k+1})^{-1}$$

2. Update state and covariance and Inject error state:

$$\delta x_{k+1} = K_{k+1}(Z_{k+1} - H_{k+1}\hat{x}_{k+1})$$

$$x_{k+1} = \hat{x}_{k+1} + \delta x_{k+1}$$

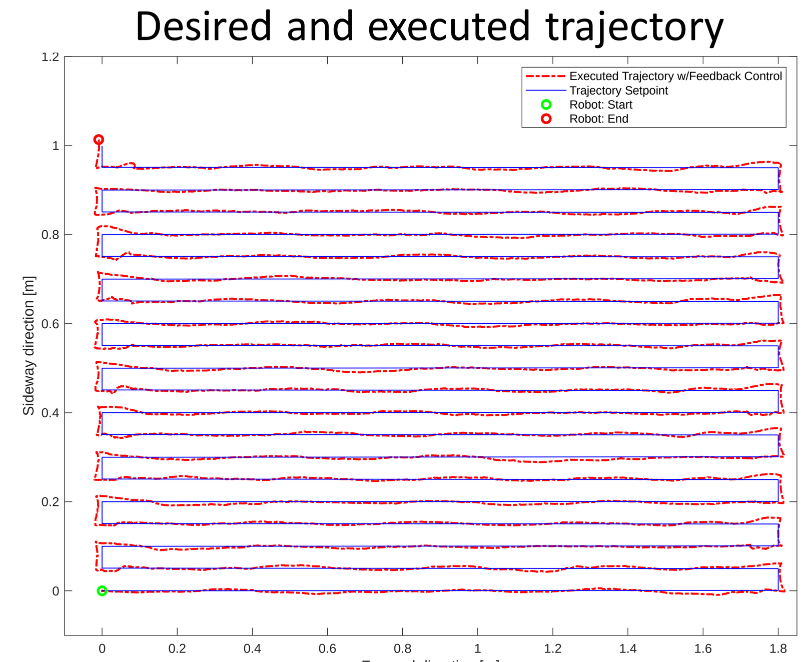
$$\Sigma_{k+1} = \hat{\Sigma}_{k+1} - K_{k+1} H_{k+1} \hat{\Sigma}_{k+1}$$

4. Restart error state ($\delta x = 0$)

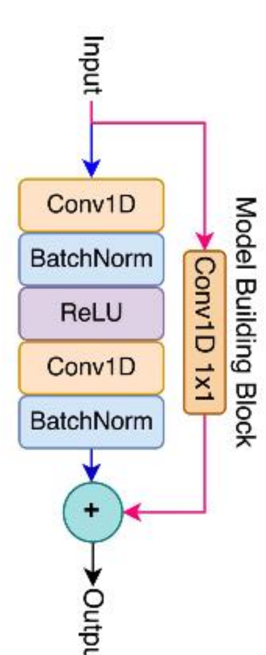
Feedback Control for Trajectory Execution

- Decoupled velocity control for each wheel.
- Platform position control.

Desired and executed trajectory



Proposed IENet Model for Impact Echo



IENET MODEL STRUCTURE

Layer	Configurations	Output Shape
Input	IE Signal	(860×1)
residual_block_1	(kernel=1x200, filters=8, stride=1)	$(860 \times 1 \times 8)$
activation_1	ReLU	$(860 \times 1 \times 8)$
maxpool1d_1	(pool_size=2, stride=2)	$(430 \times 1 \times 8)$
residual_block_2	(kernel=1x100, filters=16, stride=1)	$(430 \times 1 \times 16)$
activation_2	ReLU	$(430 \times 1 \times 16)$
maxpool1d_2	(pool_size=2, stride=2)	$(215 \times 1 \times 16)$
residual_block_3	(kernel=1x50, filters=16, stride=1)	$(215 \times 1 \times 16)$
activation_3	ReLU	$(215 \times 1 \times 16)$
maxpool1d_3	(pool_size=2, stride=2)	$(107 \times 1 \times 16)$
residual_block_4	(kernel=1x25, filters=32, stride=1)	$(107 \times 1 \times 32)$
activation_4	ReLU	$(107 \times 1 \times 32)$
maxpool1d_4	(pool_size=2, stride=2)	$(53 \times 1 \times 32)$
residual_block_5	(kernel=1x13, filters=64, stride=1)	$(53 \times 1 \times 64)$
activation_5	ReLU	$(53 \times 1 \times 64)$
maxpool1d_5	(pool_size=2, stride=2)	$(26 \times 1 \times 64)$
residual_block_6	(kernel=1x7, filters=64, stride=1)	$(26 \times 1 \times 64)$
activation_6	ReLU	$(26 \times 1 \times 64)$
maxpool1d_6	(pool_size=2, stride=2)	$(13 \times 1 \times 64)$
reshape_1	$(13 \cdot 64 \times 1)$	(832×1)
bilstm_1	BiLSTM(units=32)	(64×1)
bilstm_2	BiLSTM(units=32)	(64×1)
bilstm_3	BiLSTM(units=32)	(64×1)
Output	Dense(2, Softmax)	2×1
No# of parameters	534498	

ICRA2024 YOKOHAMA | JAPAN