Microsoft Research

Each year Microsoft Research hosts hundreds of influential speakers from around the world including leading scientists, renowned experts in technology, book authors, and leading academics, and makes videos of these lectures freely available.

Microsoft

Sensor Fusion for Learning-based Motion Estimation in VR

Chen Song[1,2] and Shuayb Zarar[2]

[1]University at Buffalo, SUNY

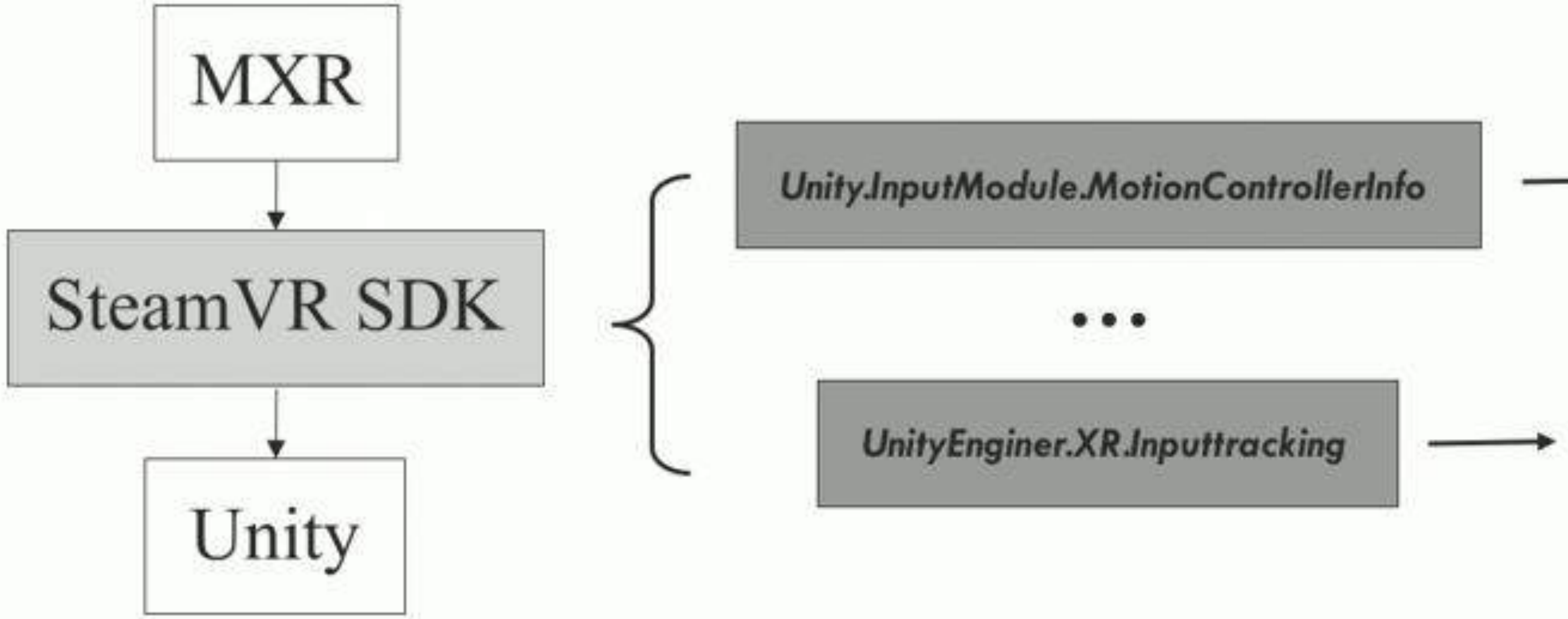
[2]Audio and Acoustics Research Group, Microsoft Research

Motion Tracking in VR

- For controller and head-mounted display (HMD)
- 3D orientation and position
- Continuously tracked over time
- Traditionally with computer vision
- Isotropic accuracy is important

How to Get Position Tracking Data

- For Windows Mixed Reality (MXR)

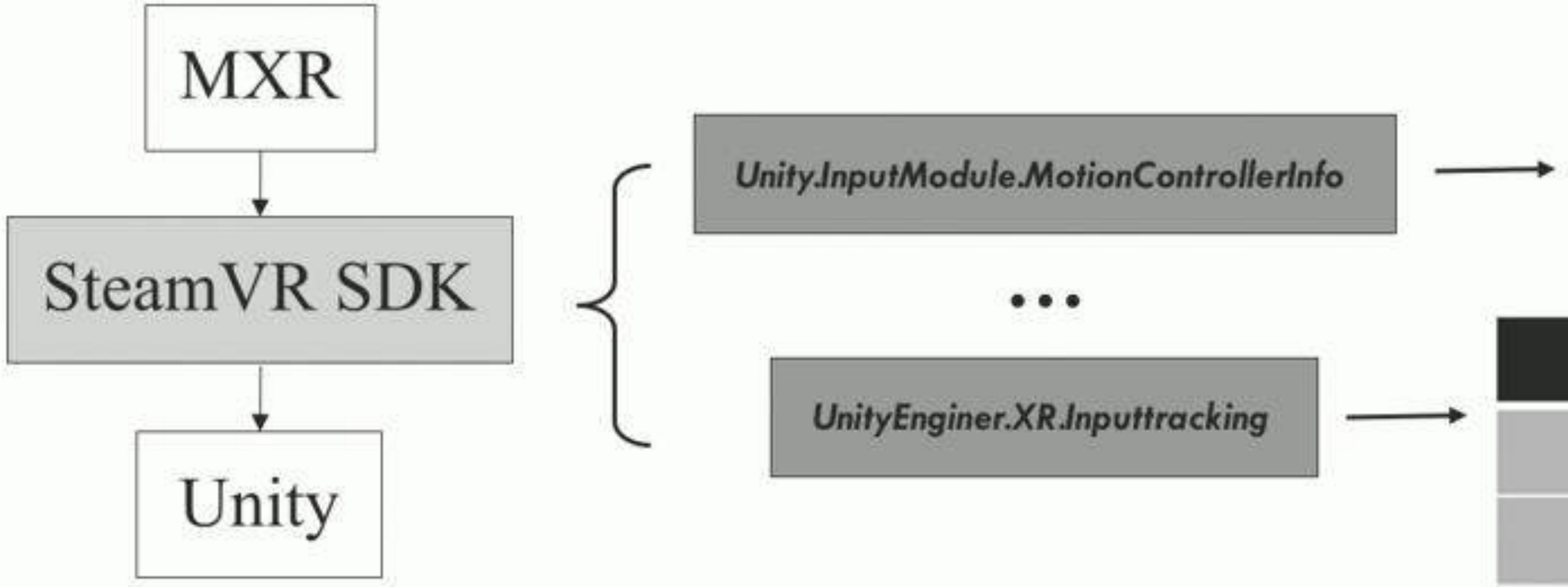


HMD		Controller	
Pos(xyz)	*Rot(xyz)*	*Pos(xyz)*	*Rot(xyz)*

Tracking State	Index	Accuracy (cm)
High	*2*	*< 7*
Approximate	*1*	*20~inf*
None	*0*	*---*

How to Get Position Tracking Data

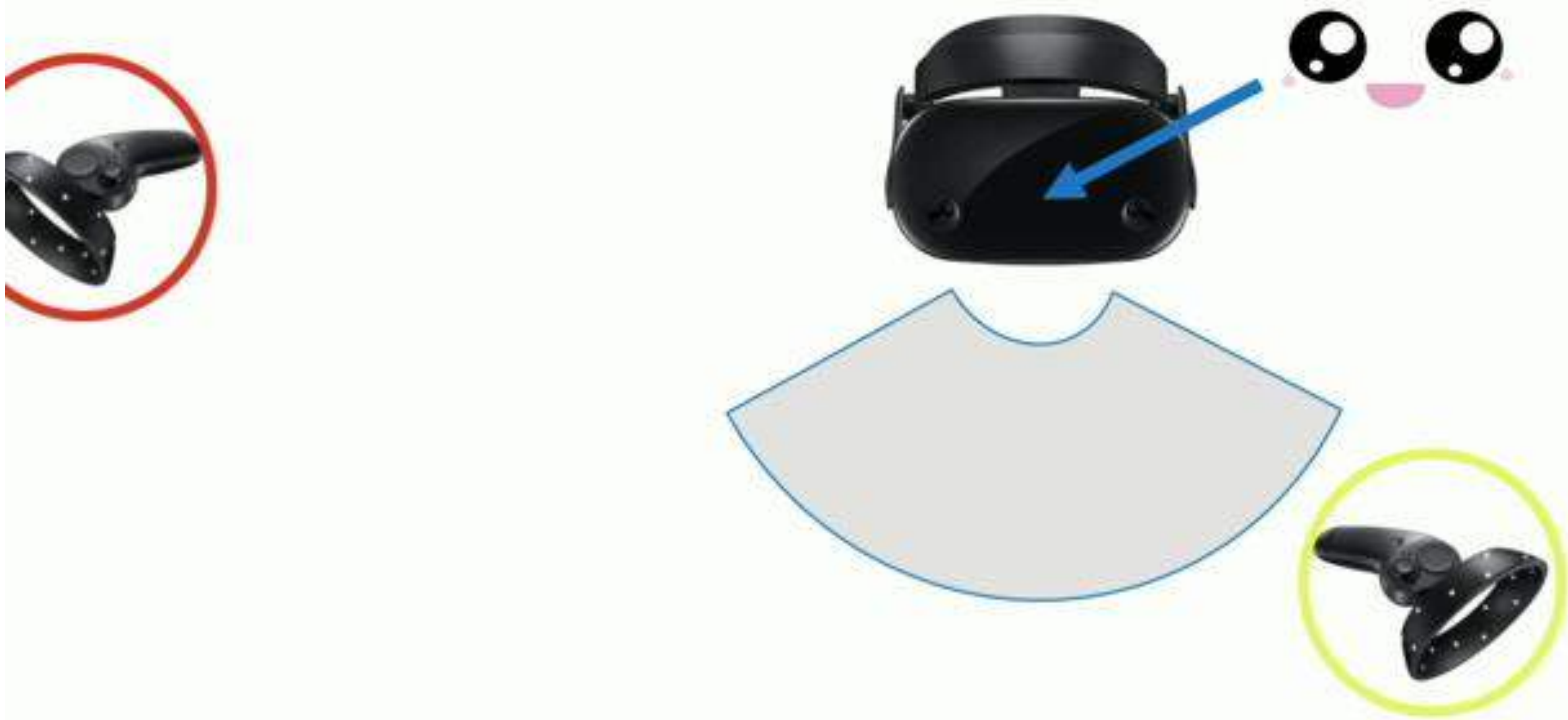
- For Windows Mixed Reality (MXR)

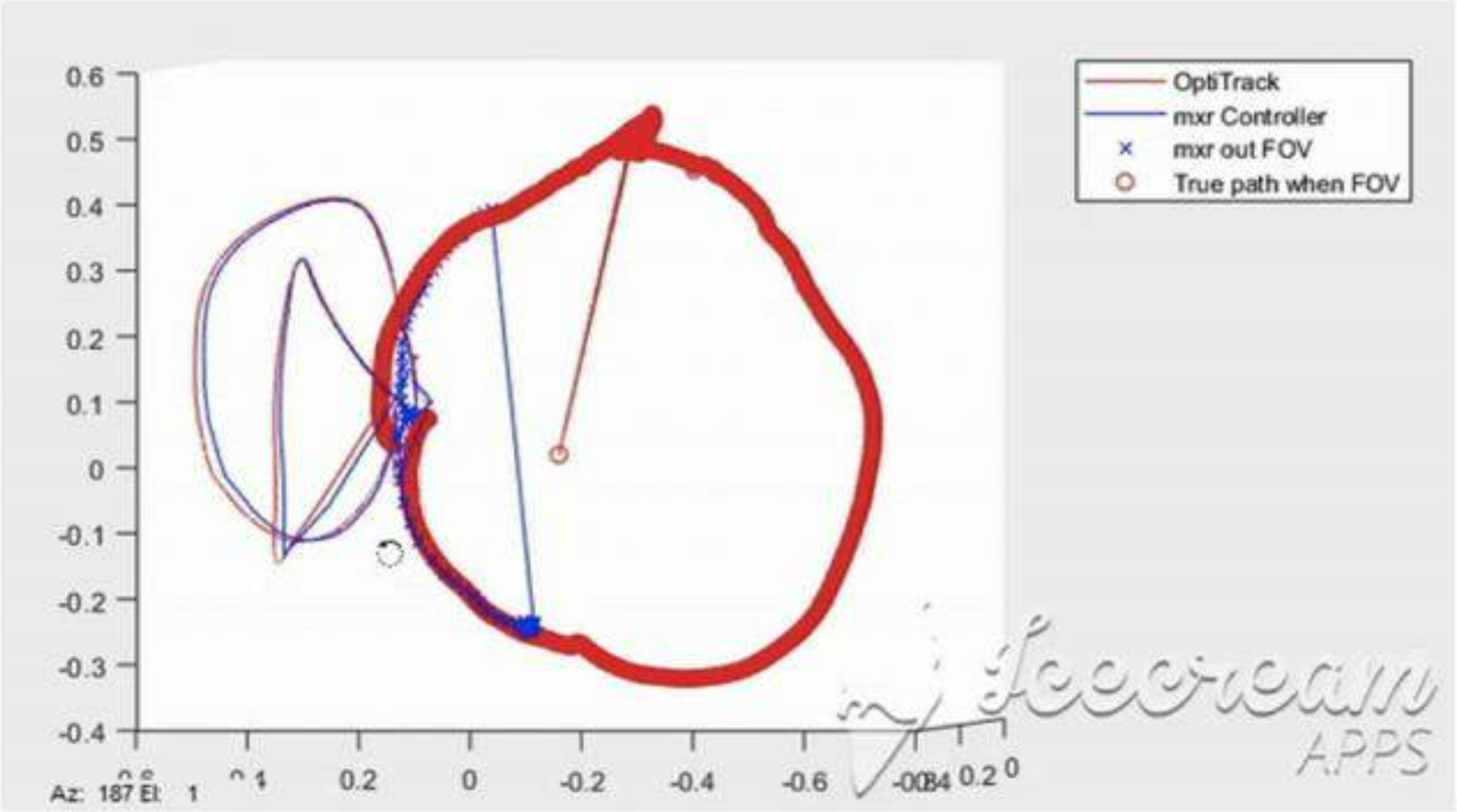


HMD		Controller	
Pos(xyz)	*Rot(xyz)*	*Pos(xyz)*	*Rot(xyz)*

Tracking State	Index
High	*2*
Approximate	*1*
Out of FOV	*4 consecutive 1*

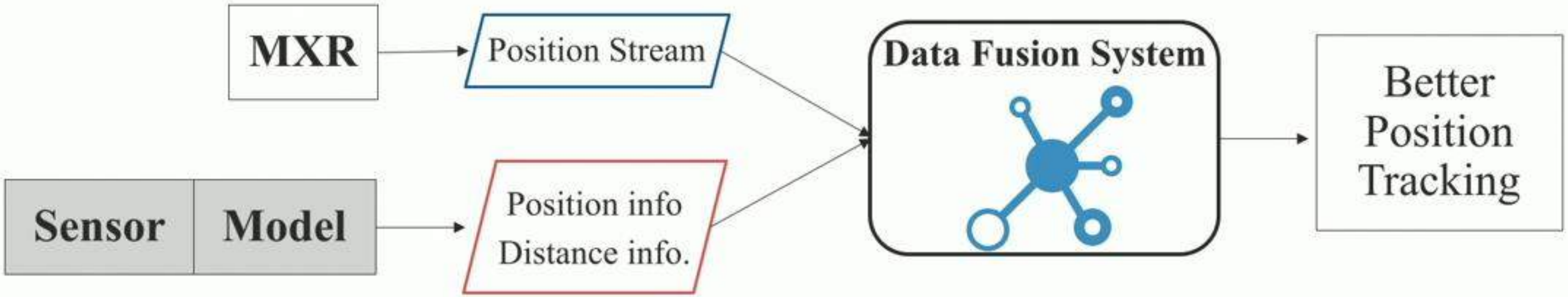
Performance of MXR Tracking





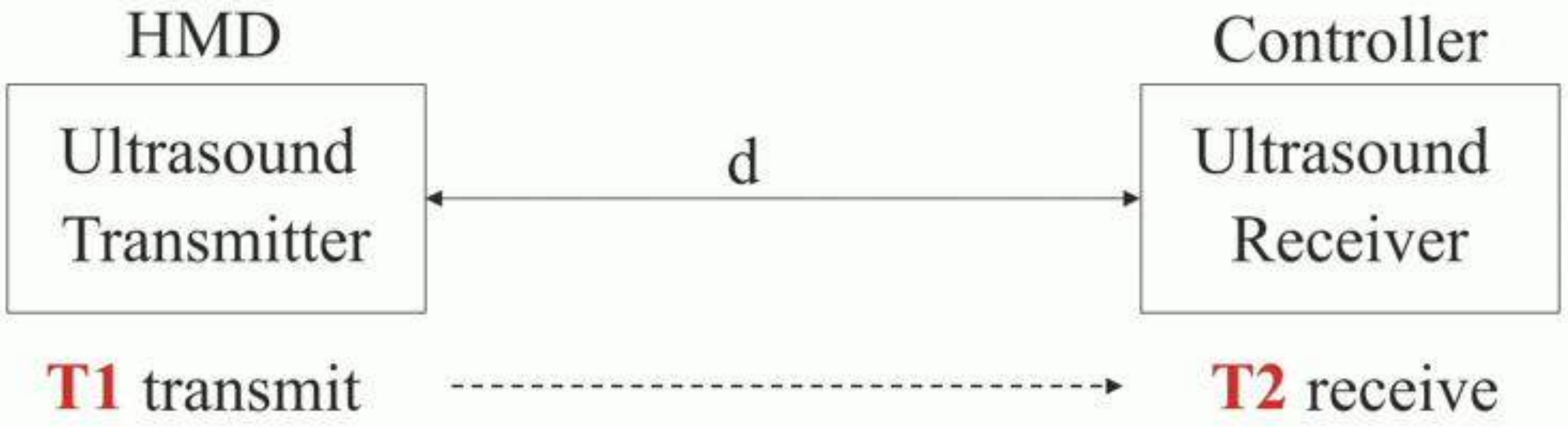
Proposed Approach

- Lightweight Sensor
- Learning-based Model
- Data Fusion System



1. Lightweight Sensor for Ultrasound Ranging

- Principle
  - Calculate the distance via time of flight

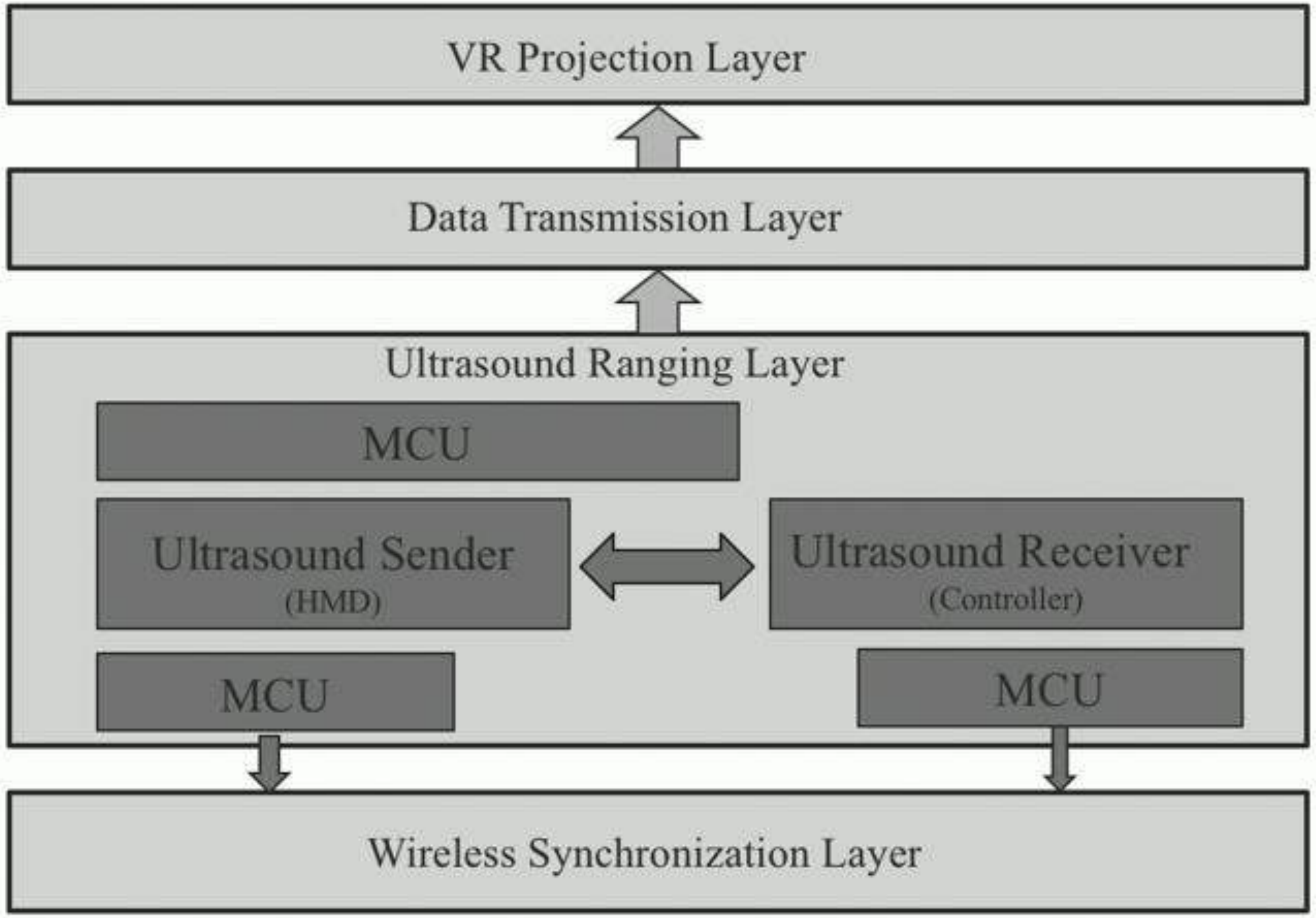


- $d = v_{sound} * (T2 - T1);$
- $v_{sound} = 331 + 0.6 * Temperature(m/s).$

Need time synchronization for accurate measurement

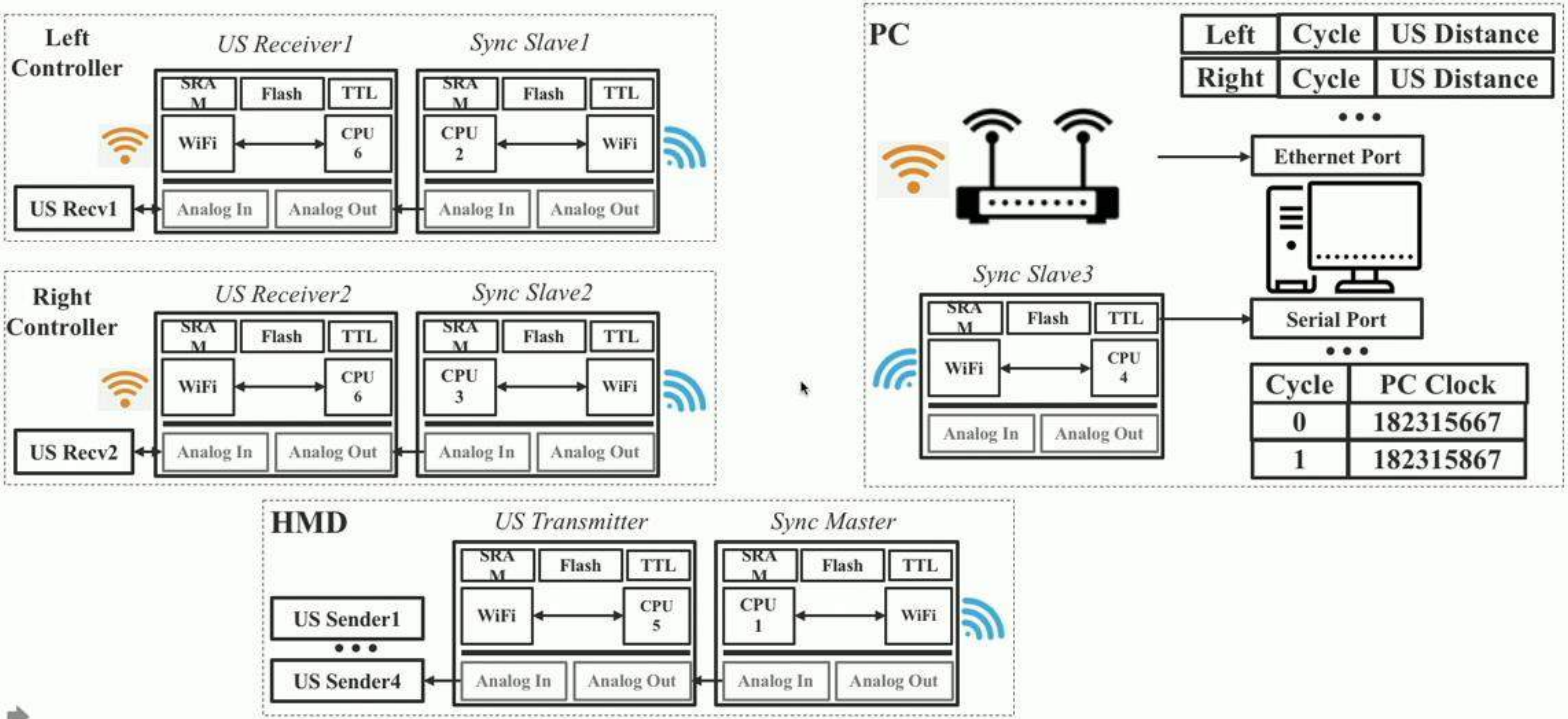
1. Lightweight Sensor for Ultrasound Ranging

- Architecture



1. Lightweight Sensor for Ultrasound Ranging

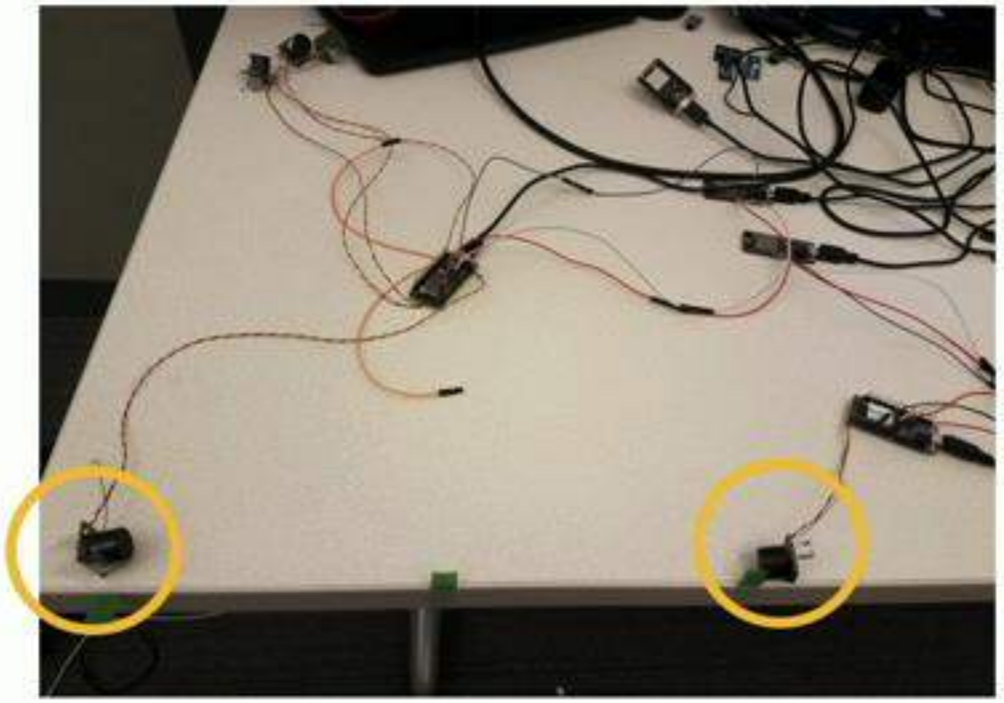
- Diagram

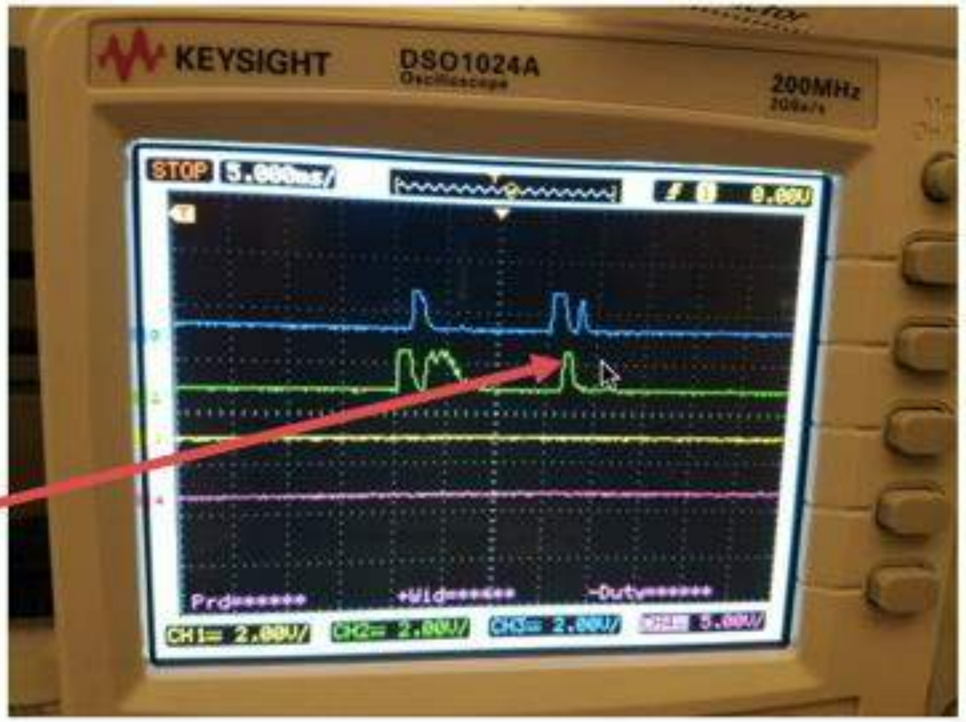


1. Lightweight Sensor for Ultrasound Ranging

- Hardware

MCU (Adafruit ESP8266)		Ultrasound Sensor (MB1330 MaxSonar)	
Frequency	*80 MHz*	***range***	*20-765 cm*
Flash	*4 MB*	***Rate***	*10 Hz*
Vdd	*3.3 V*	***Vdd***	*3.3-5.5 V*
Protocol	*WiFi UDP*	***Resolution***	*1 cm*
Total Power	*478 mW*	***Ultrasound Frequency***	*42 KHz*
Debugging Tool			
KeySight DSO1024A Oscilloscope (200 MHz)			



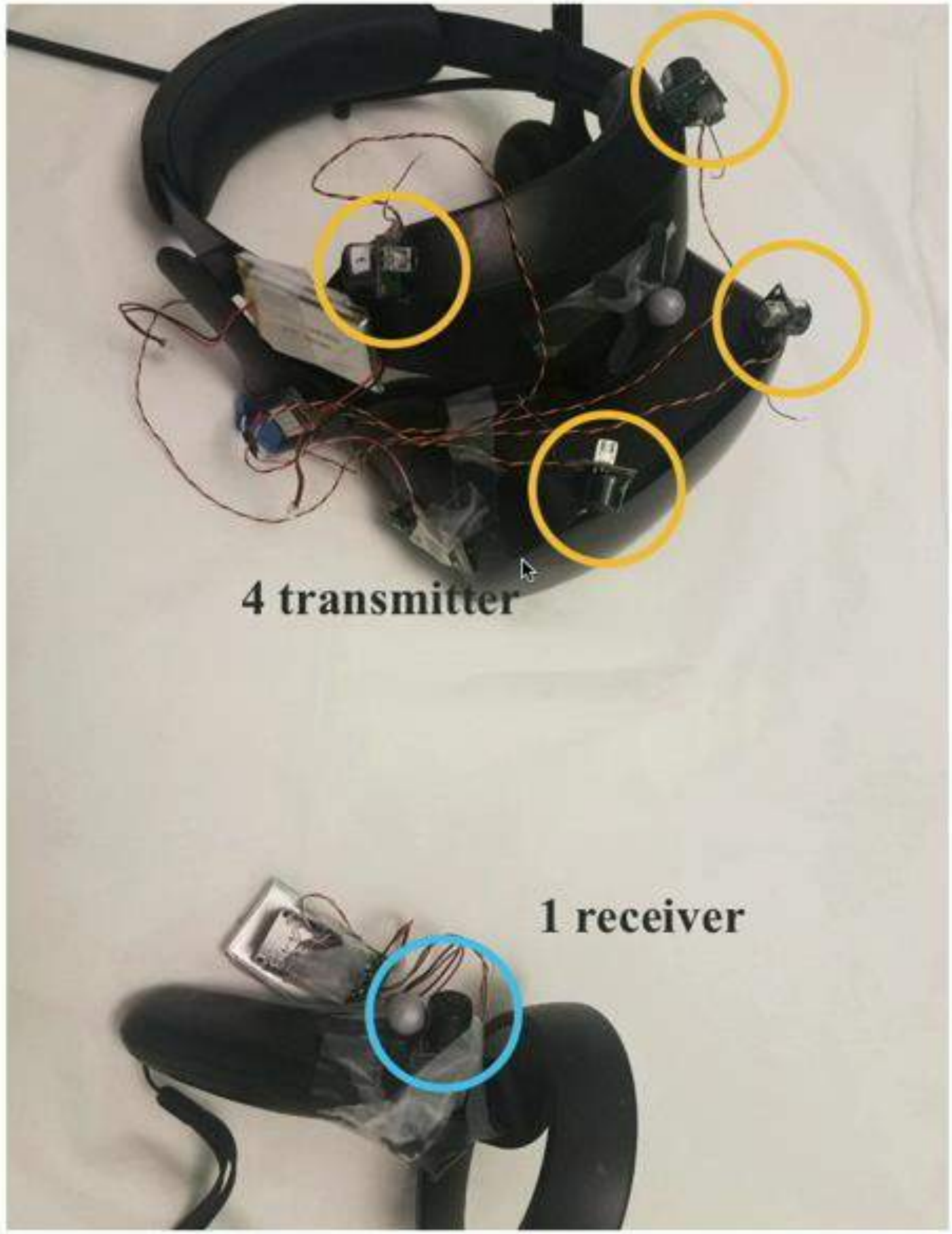


Time difference should match the physical distance.

1. Lightweight Sensor for Ultrasound Ranging

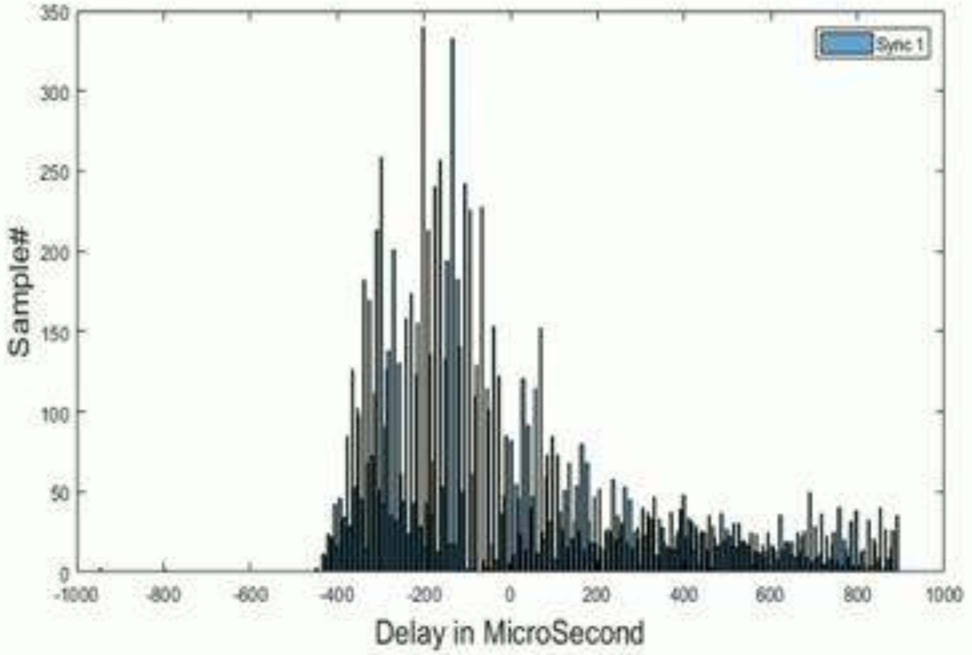
- Hardware

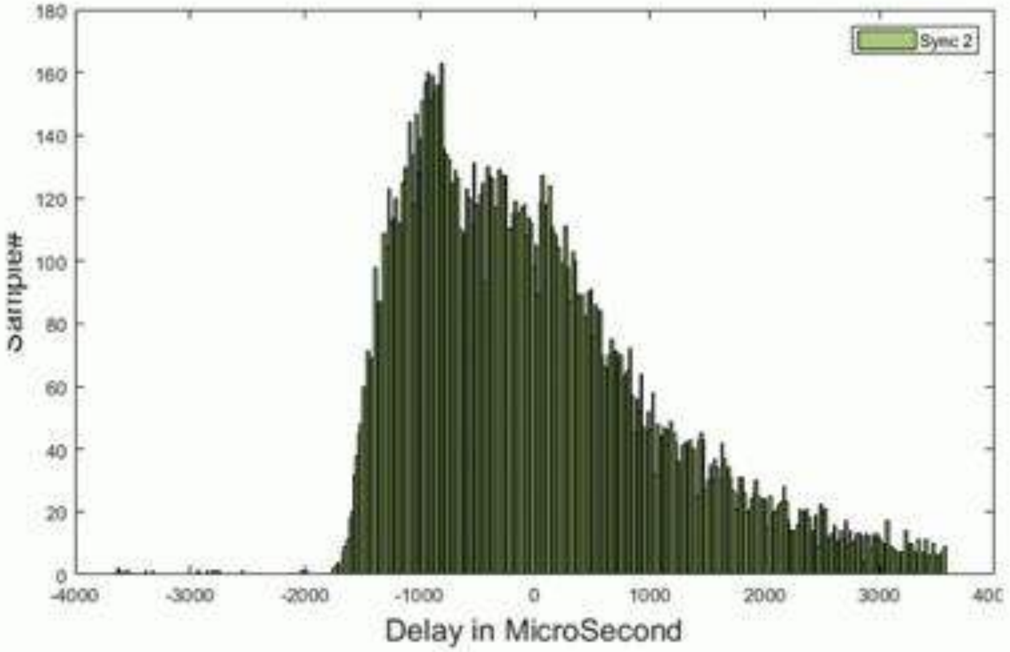
MCU (Adafruit ESP8266)		Ultrasound Sensor (MB1330 MaxSonar)	
Frequency	*80 MHz*	***range***	*20-765 cm*
Flash	*4 MB*	***Rate***	*10 Hz*
Vdd	*3.3 V*	***Vdd***	*3.3-5.5 V*
Protocol	*WiFi UDP*	***Resolution***	*1 cm*
Total Power	*478 mW*	***Ultrasound Frequency***	*42 KHz*
Debugging Tool			
KeySight DSO1024A Oscilloscope (200 MHz)			

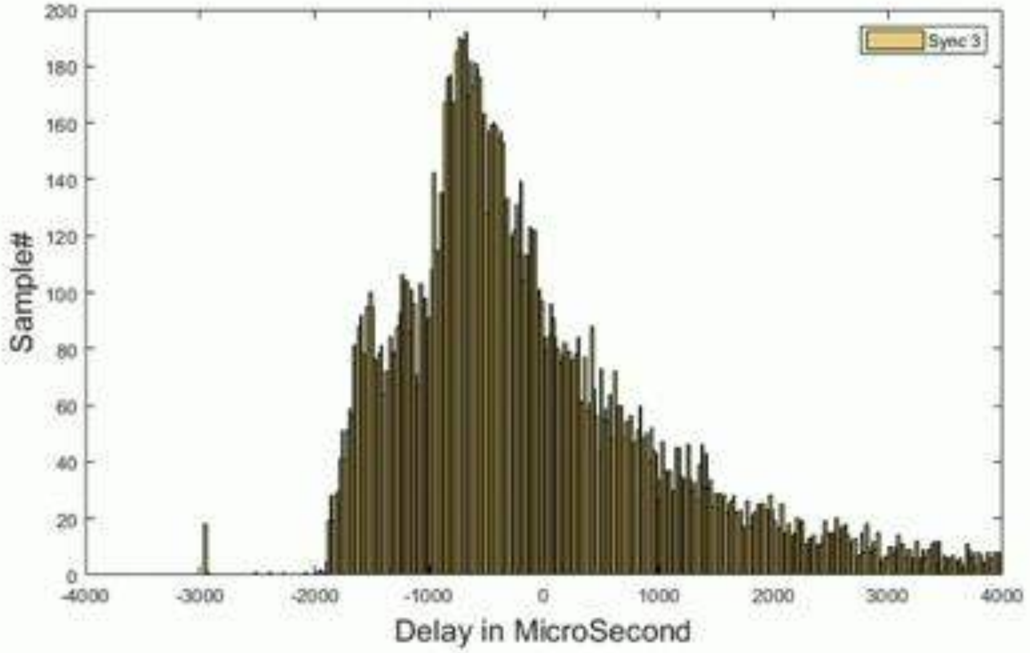


1. Lightweight Sensor for Ultrasound Ranging

- Synchronization performance
  - Bluetooth, Router WiFi have large random delay (upto 15 ms).
  - We use internal WiFi channel.



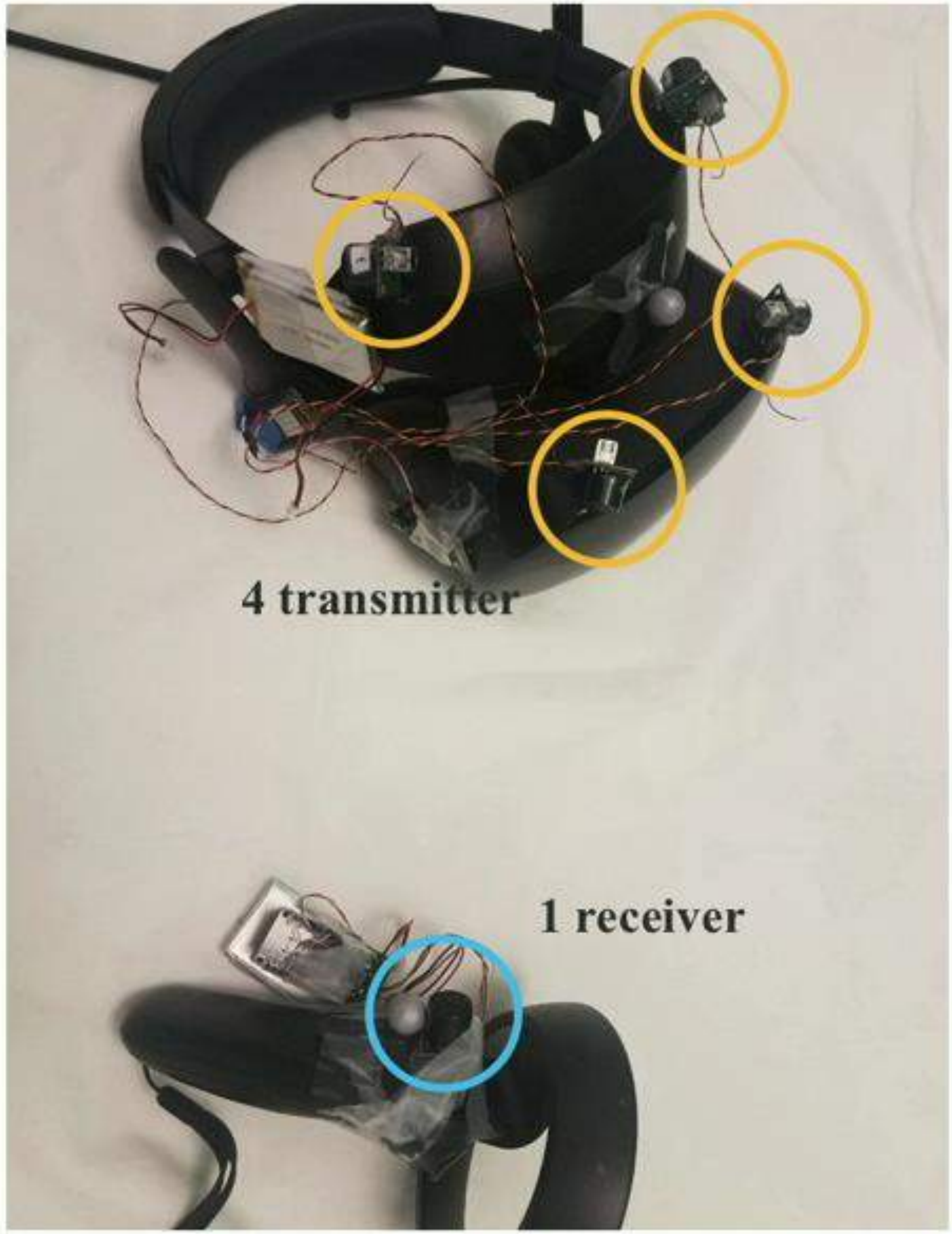




1. Lightweight Sensor for Ultrasound Ranging

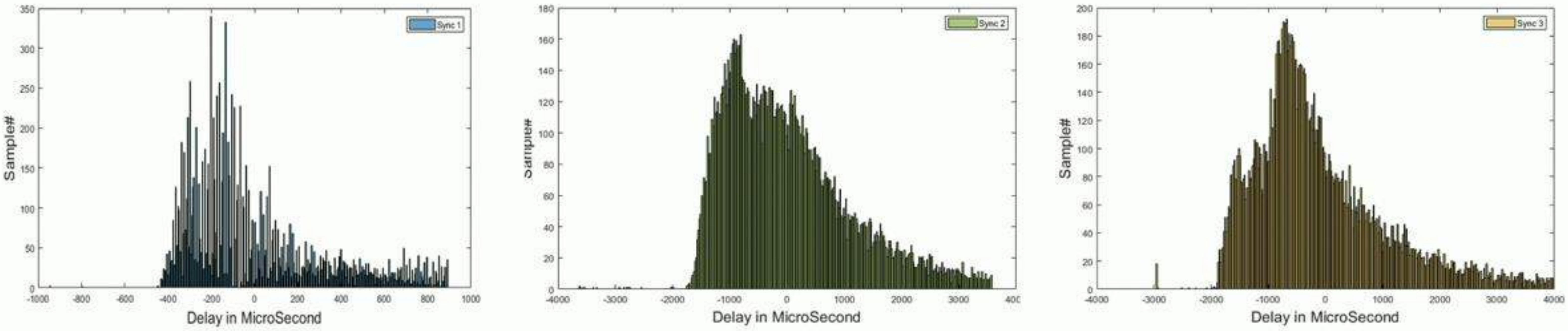
- Hardware

MCU (Adafruit ESP8266)		Ultrasound Sensor (MB1330 MaxSonar)	
Frequency	80 MHz	**range**	20-765 cm
Flash	4 MB	**Rate**	10 Hz
Vdd	3.3 V	**Vdd**	3.3-5.5 V
Protocol	WiFi UDP	**Resolution**	1 cm
Total Power	478 mW	**Ultrasound Frequency**	42 KHz
Debugging Tool			
KeySight DSO1024A Oscilloscope (200 MHz)			



1. Lightweight Sensor for Ultrasound Ranging

- Synchronization performance
  - Bluetooth, Router WiFi have large random delay (upto 15 ms).
  - We use internal WiFi channel.

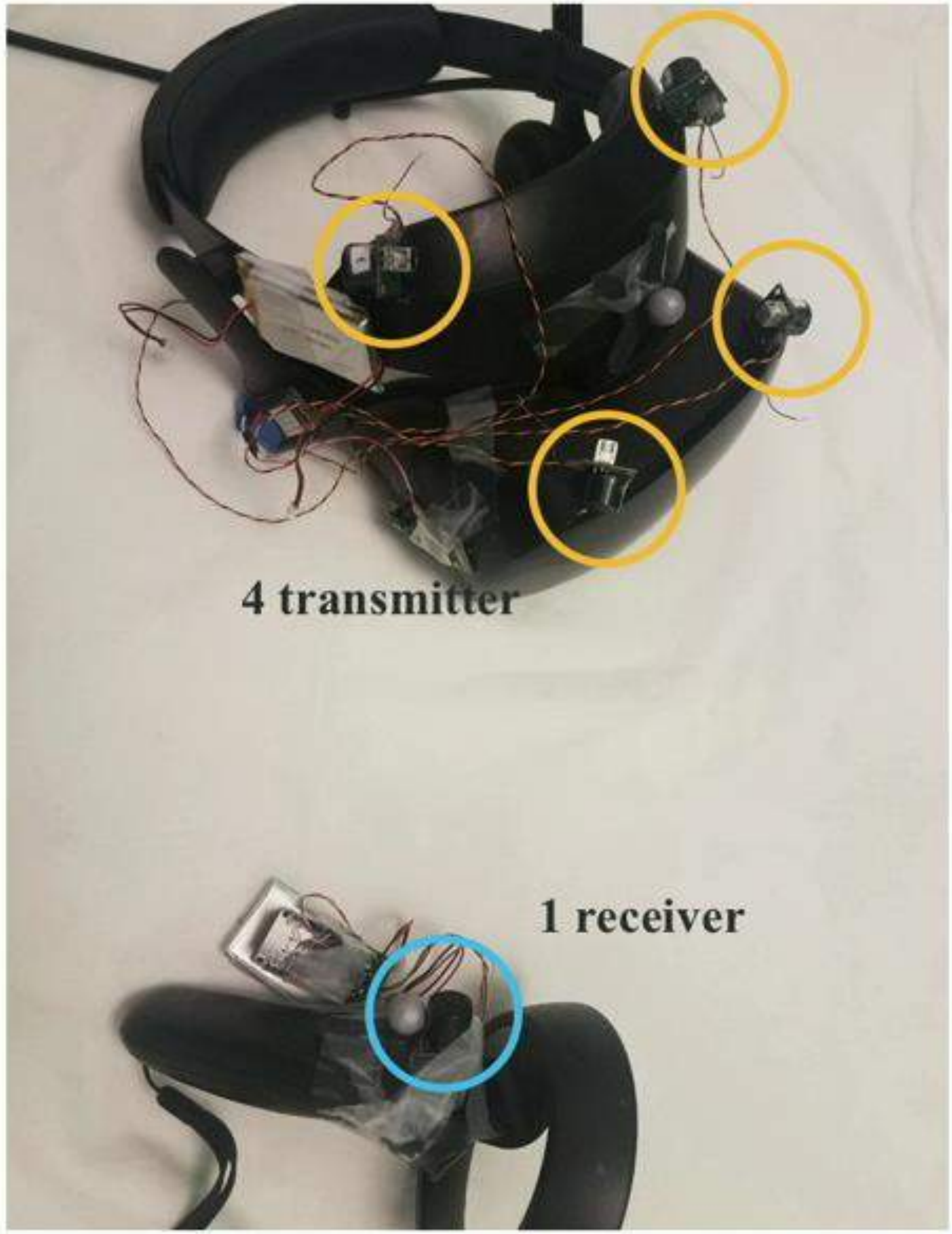


Std1: 0.55ms; Std2: 1.09ms; Std3: 1.40ms

1. Lightweight Sensor for Ultrasound Ranging

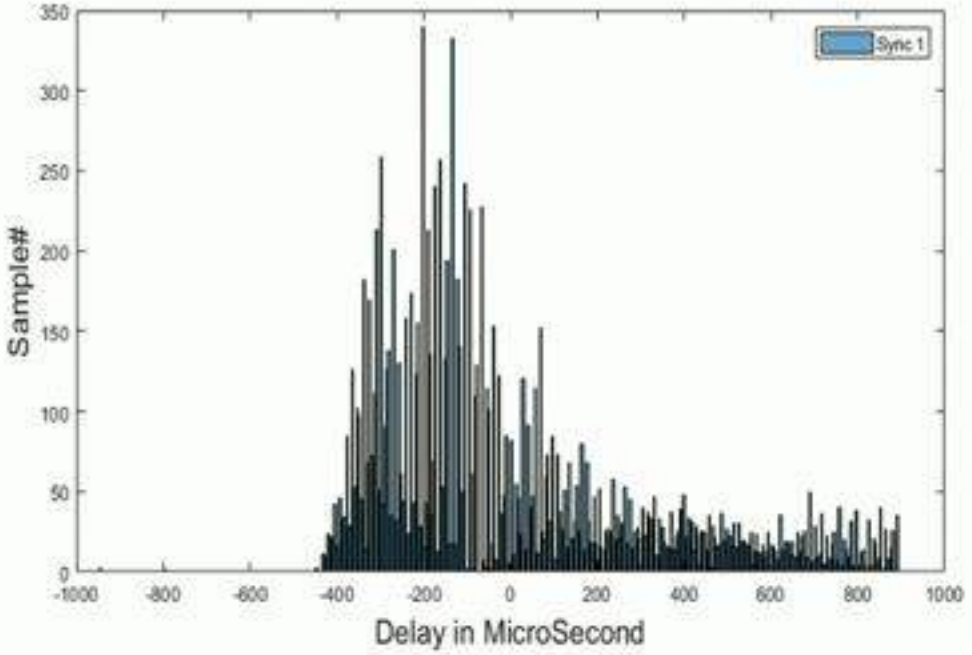
- Hardware

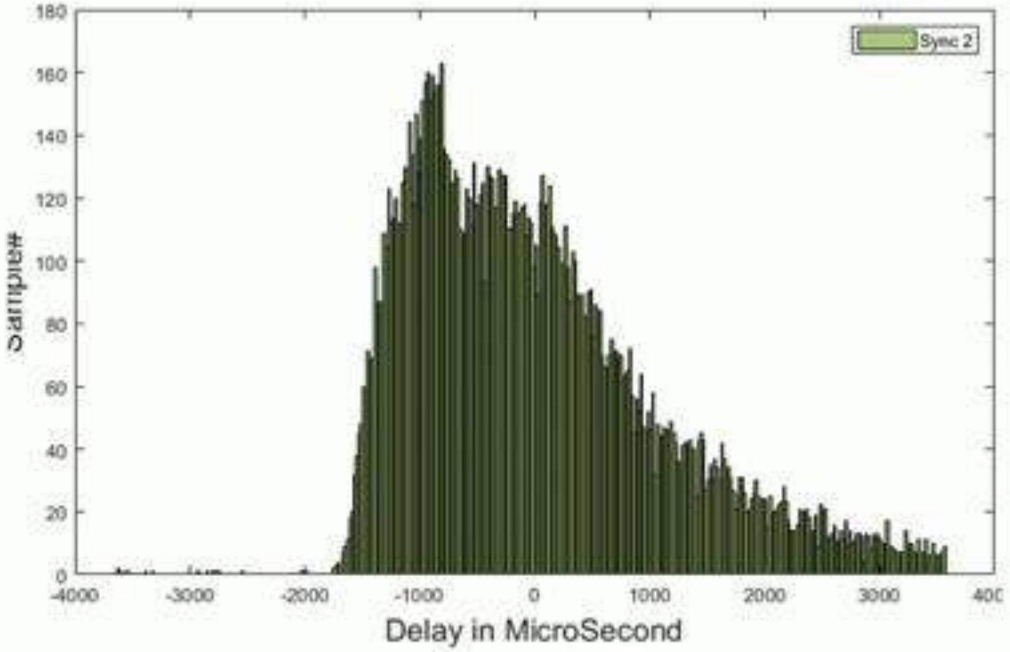
MCU (Adafruit ESP8266)		Ultrasound Sensor (MB1330 MaxSonar)	
Frequency	*80 MHz*	***range***	*20-765 cm*
Flash	*4 MB*	***Rate***	*10 Hz*
Vdd	*3.3 V*	***Vdd***	*3.3-5.5 V*
Protocol	*WiFi UDP*	***Resolution***	*1 cm*
Total Power	*478 mW*	***Ultrasound Frequency***	*42 KHz*
Debugging Tool			
KeySight DSO1024A Oscilloscope (200 MHz)			

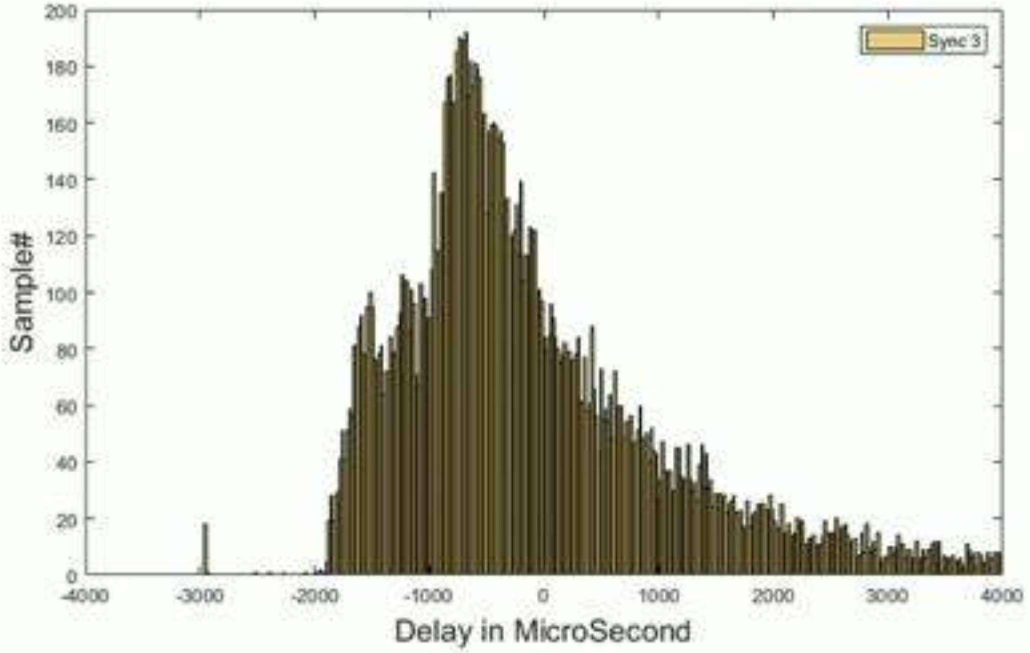


1. Lightweight Sensor for Ultrasound Ranging

- Synchronization performance
  - Bluetooth, Router WiFi have large random delay (upto 15 ms).
  - We use internal WiFi channel.



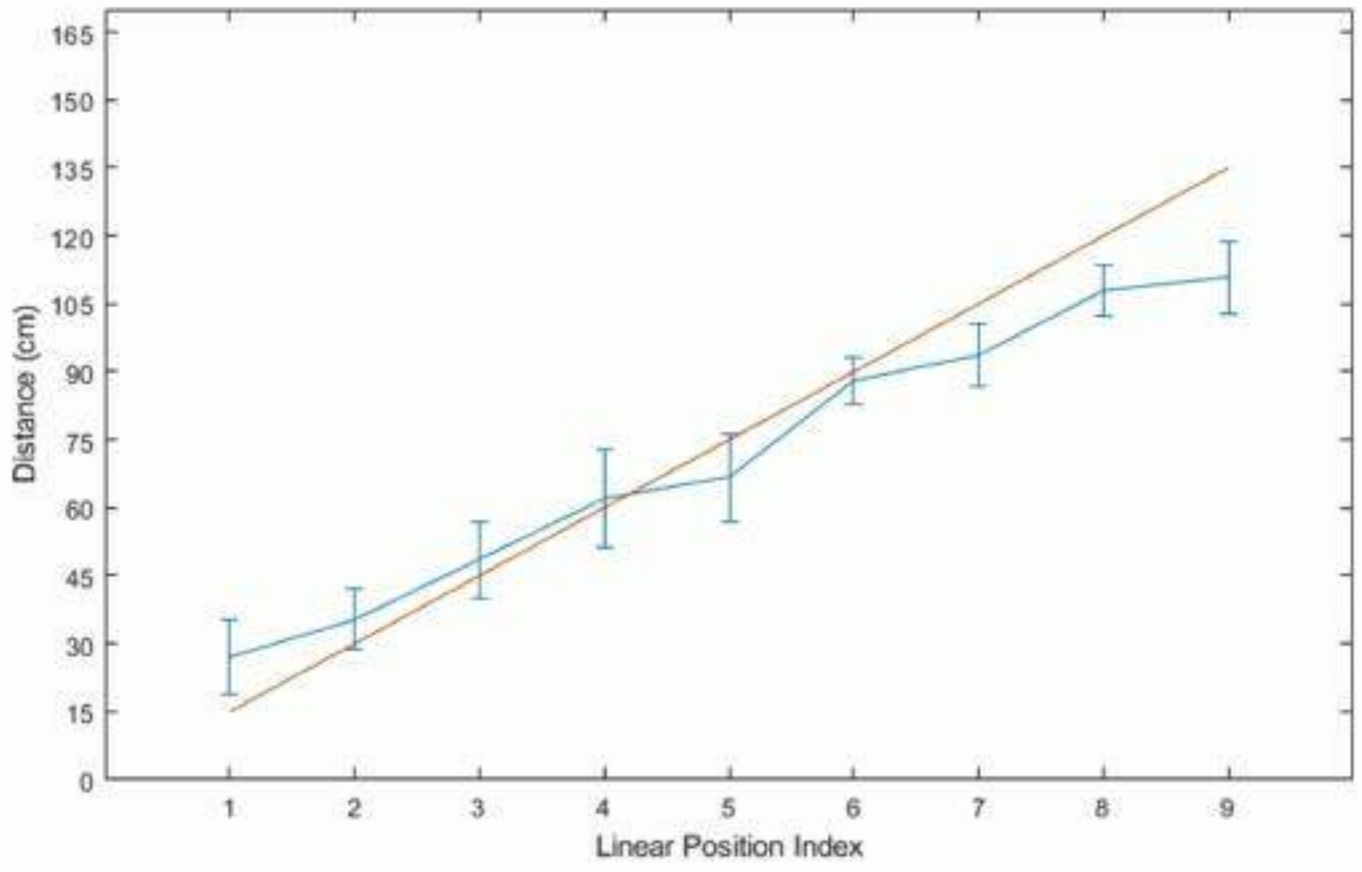


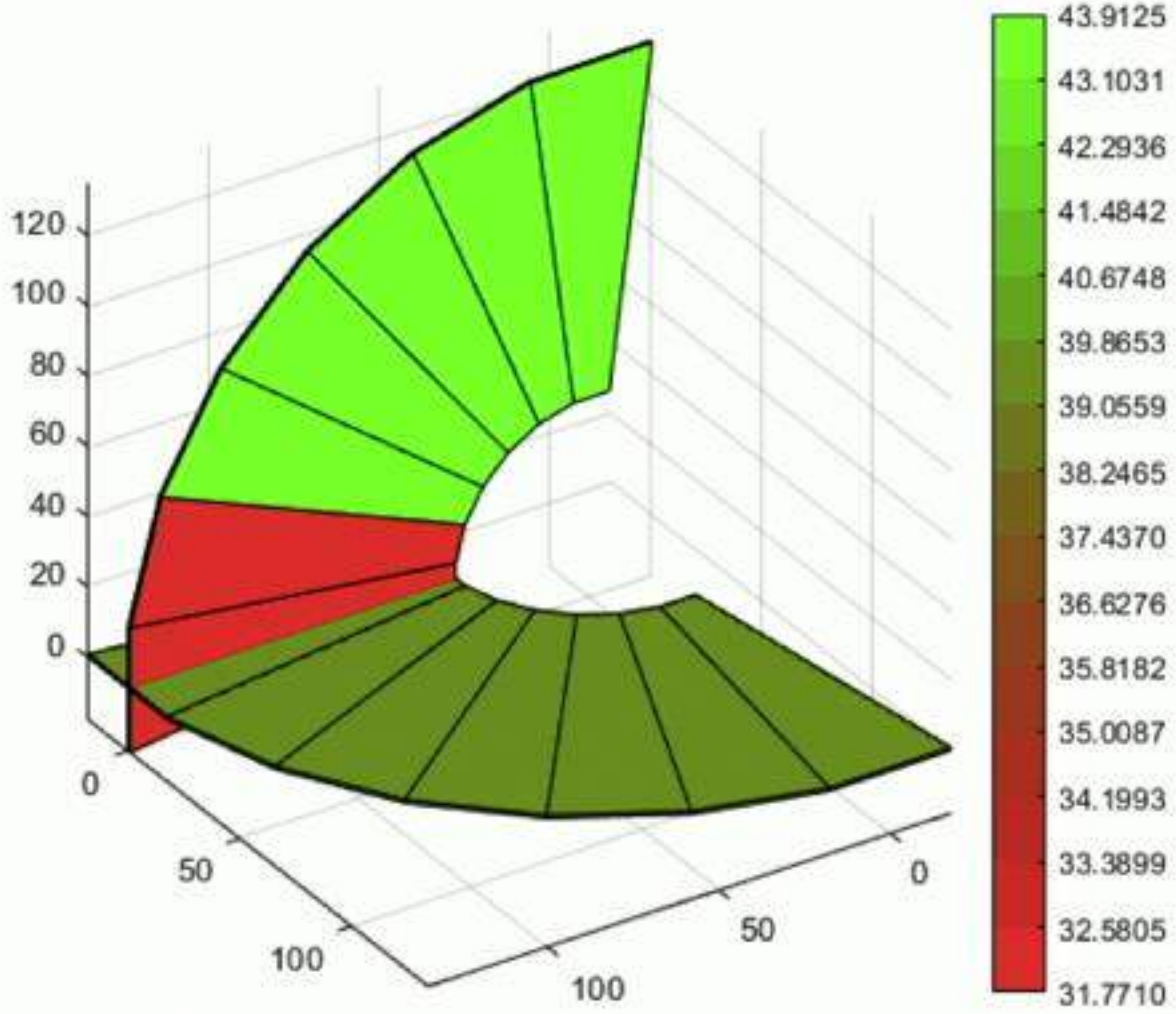


Std1: 0.55ms; Std2: 1.09ms; Std3: 1.40ms

1. Lightweight Sensor for Ultrasound Ranging

- Tracking accuracy (distance and angle)





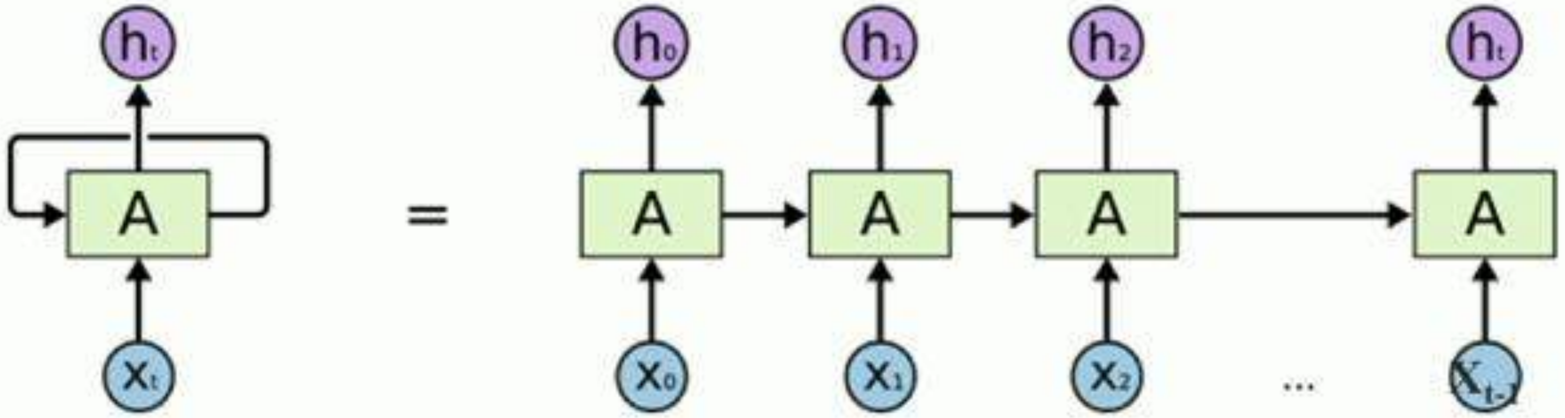
Average Distance Error: 9.74cm; Std:10.31cm

2. Learning-based Model for Tracking

- Autoregressive time-series forecasting model

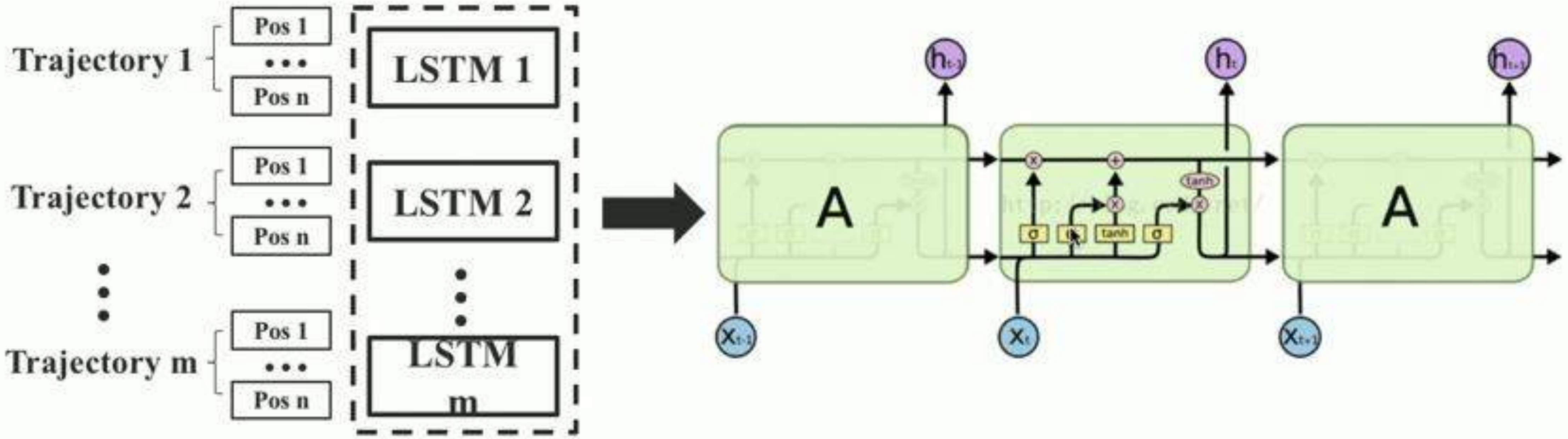
$$X_t = a_0 + a_1 X_{t-1} + a_2 X_{t-2} + \cdots + a_p X_{t-p} + \varepsilon_t, t \in \mathbb{Z} \qquad \{\varepsilon_t\} \in N(0,\sigma^2)$$

- Can also be learned using machine learning



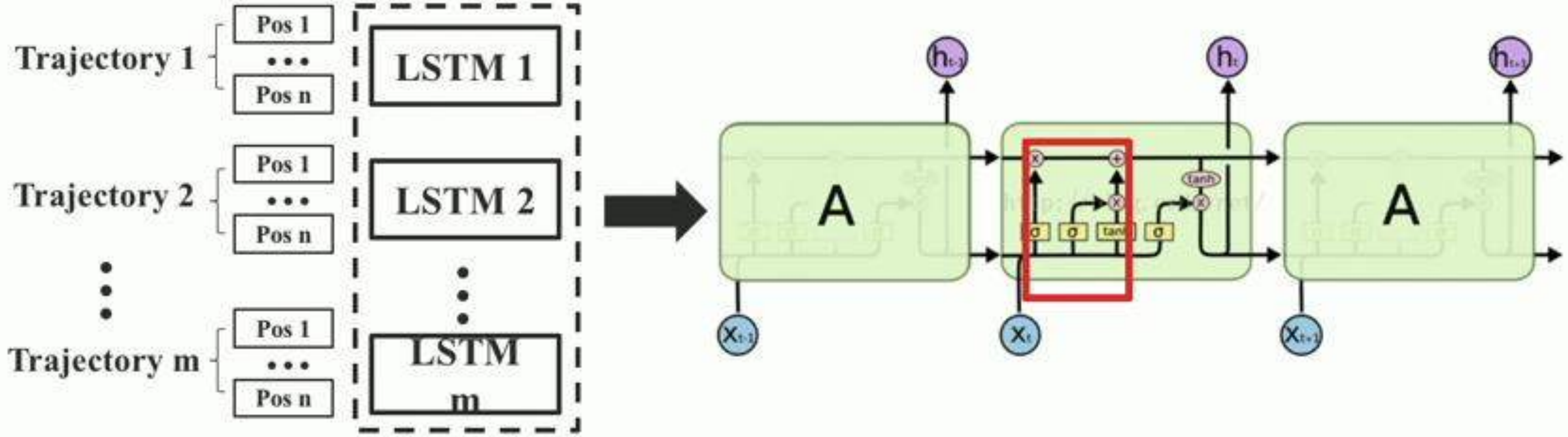
2. Learning-based Model for Tracking

- RNNs: Natural model for tracking time-series data
- We use LSTMs to track time-series data



2. Learning-based Model for Tracking

- RNNs: Natural model for tracking time-series data
- We use LSTMs to track time-series data



Forget Gate:

$$f_t = \sigma(W_f * [h_{t-1}, x_t] + b_f)$$

Input Gate:

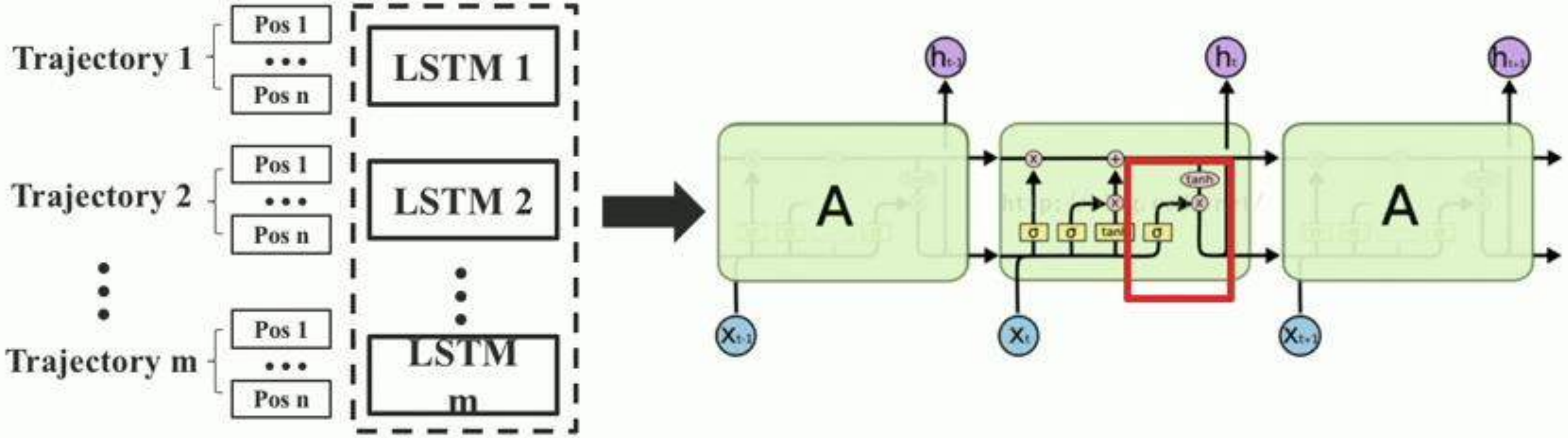
$$i_t = \sigma(W_i * [h_{t-1}, x_t] + b_i)$$

$$\widetilde{C}_t = tanh(W_C * [h_{t-1}, x_t] + b_C)$$

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t$$

2. Learning-based Model for Tracking

- RNNs: Natural model for tracking time-series data
- We use LSTMs to track time-series data



Forget Gate:

$$f_t = \sigma(W_f * [h_{t-1}, x_t] + b_f)$$

Input Gate:

$$i_t = \sigma(W_i * [h_{t-1}, x_t] + b_i)$$
$$\widetilde{C}_t = tanh(W_C * [h_{t-1}, x_t] + b_C)$$
$$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t$$

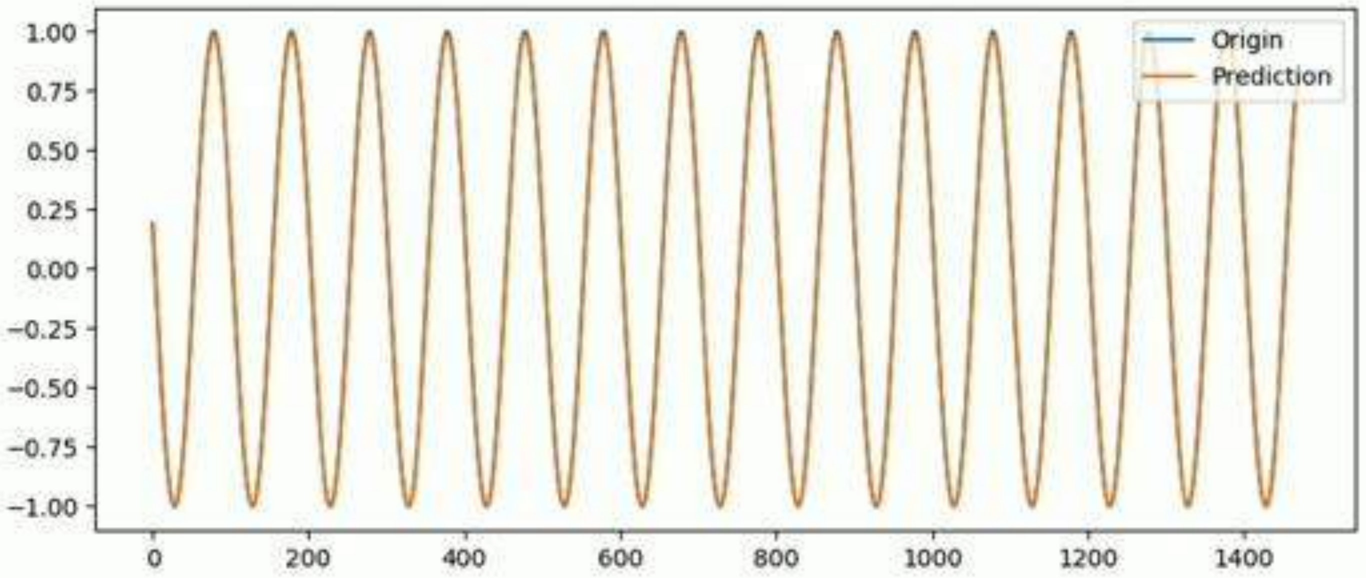
Output Gate:

$$o_t = \sigma(W_o * [h_{t-1}, x_t] + b_o)$$
$$h_t = o_t * \tanh(C_t)$$

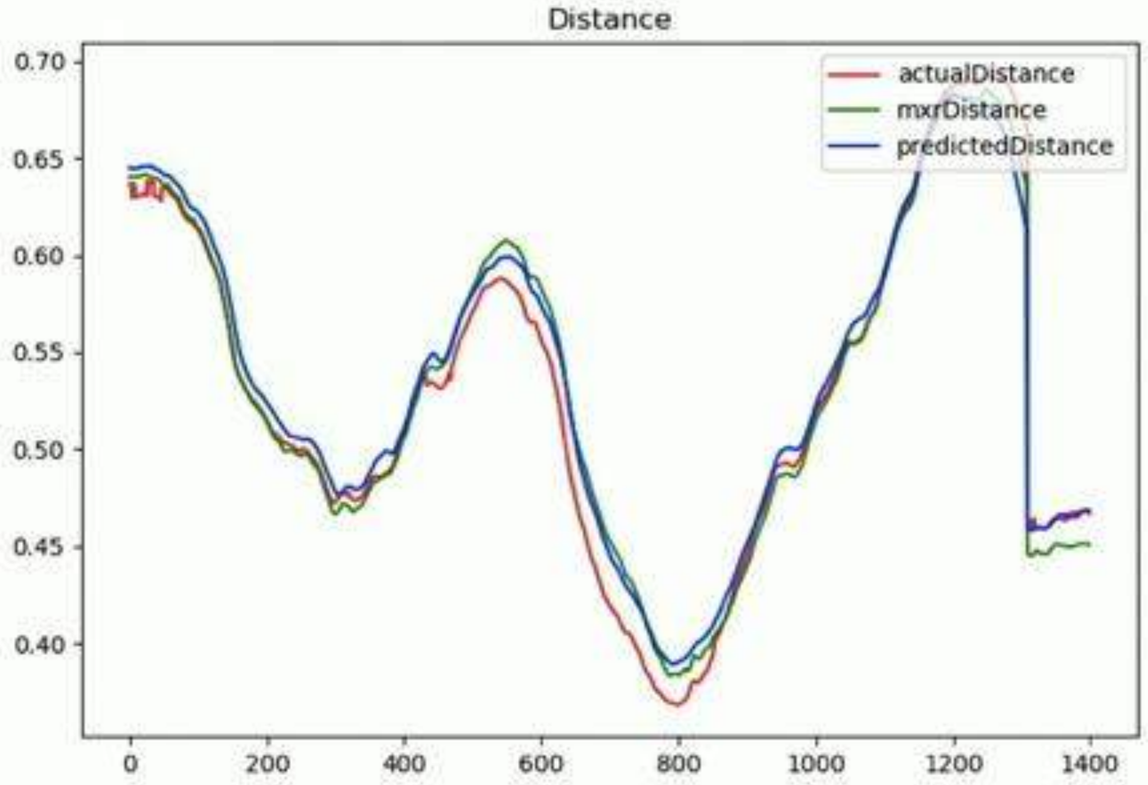
2. Learning-based Model for Tracking

- Tracking performance for Sinusoid and MXR data

Autoregression on sinusoid

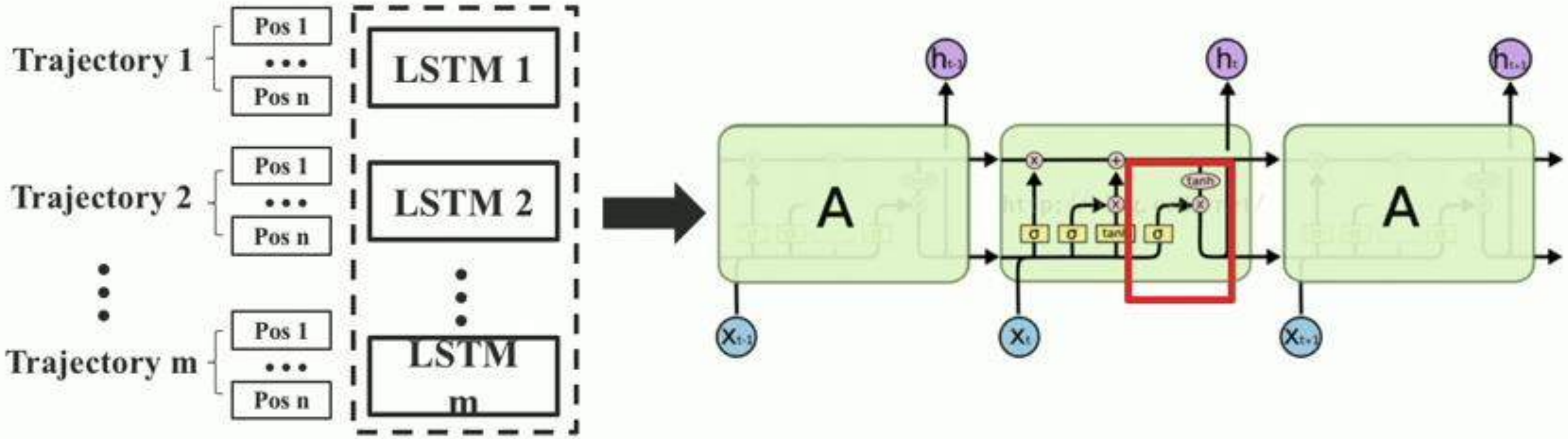


Autoregression on MXR Data



2. Learning-based Model for Tracking

- RNNs: Natural model for tracking time-series data
- We use LSTMs to track time-series data



Forget Gate:

$$f_t = \sigma(W_f * [h_{t-1}, x_t] + b_f)$$

Input Gate:

$$i_t = \sigma(W_i * [h_{t-1}, x_t] + b_i)$$

$$\widetilde{C}_t = tanh(W_C * [h_{t-1}, x_t] + b_C)$$

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t$$

Output Gate:

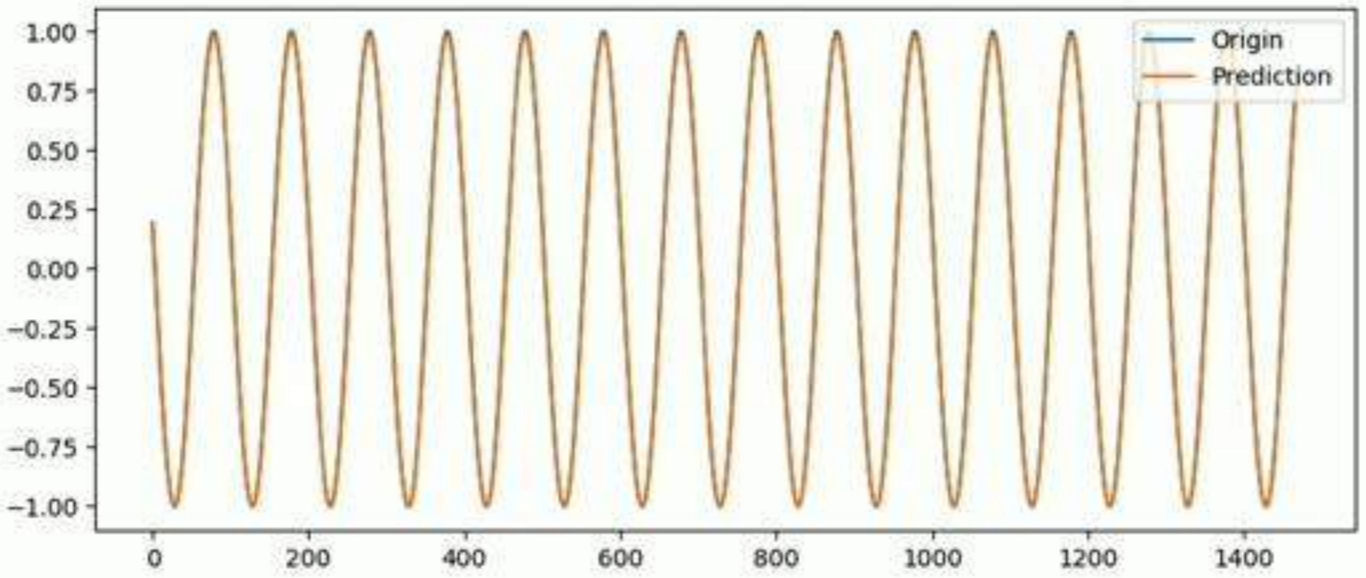
$$o_t = \sigma(W_o * [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh(C_t)$$

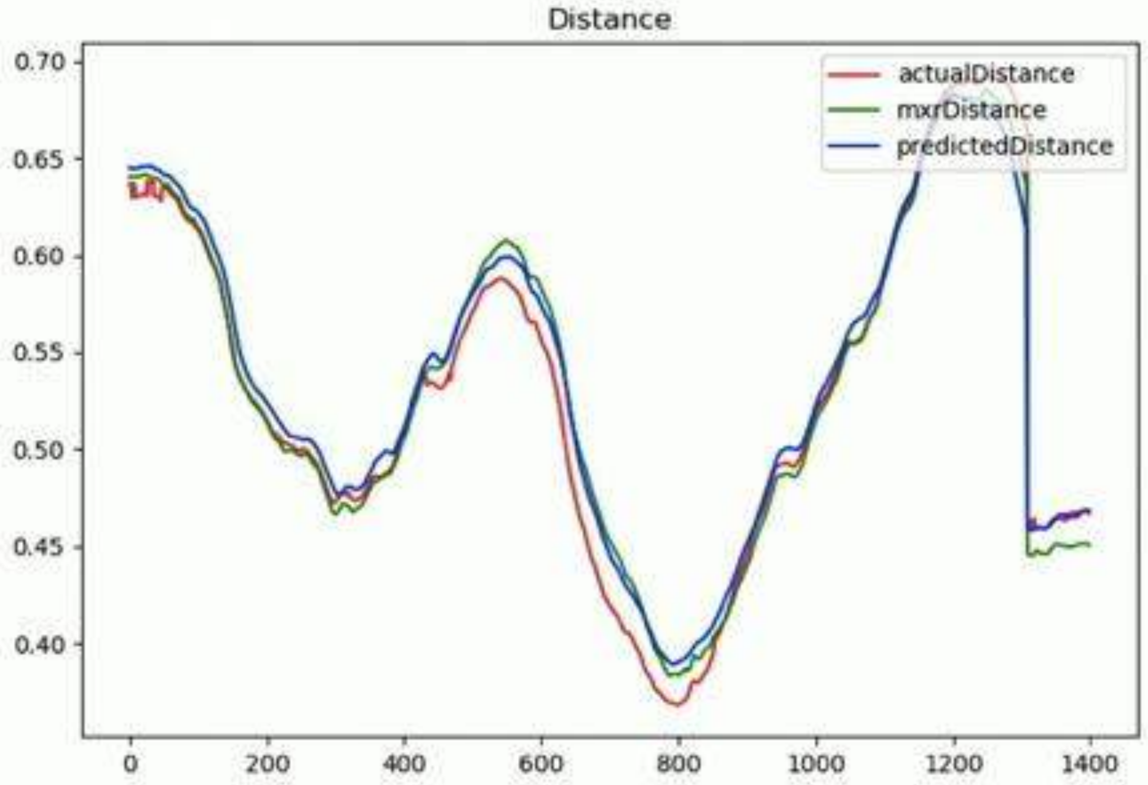
2. Learning-based Model for Tracking

- Tracking performance for Sinusoid and MXR data

Autoregression on sinusoid



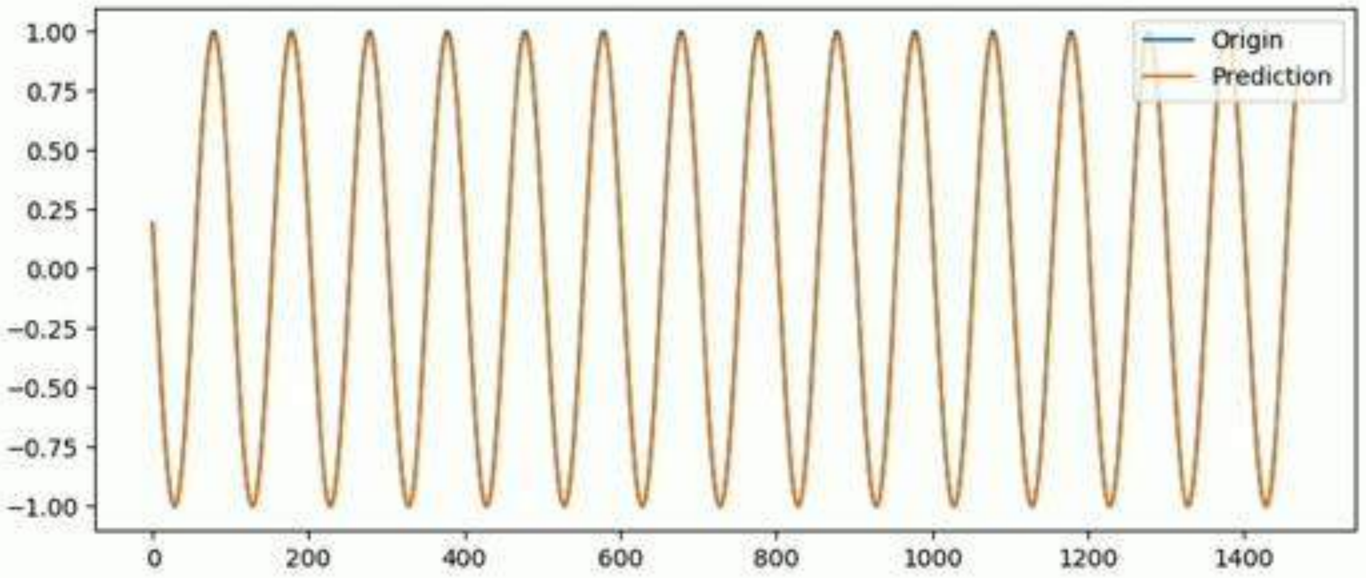
Autoregression on MXR Data



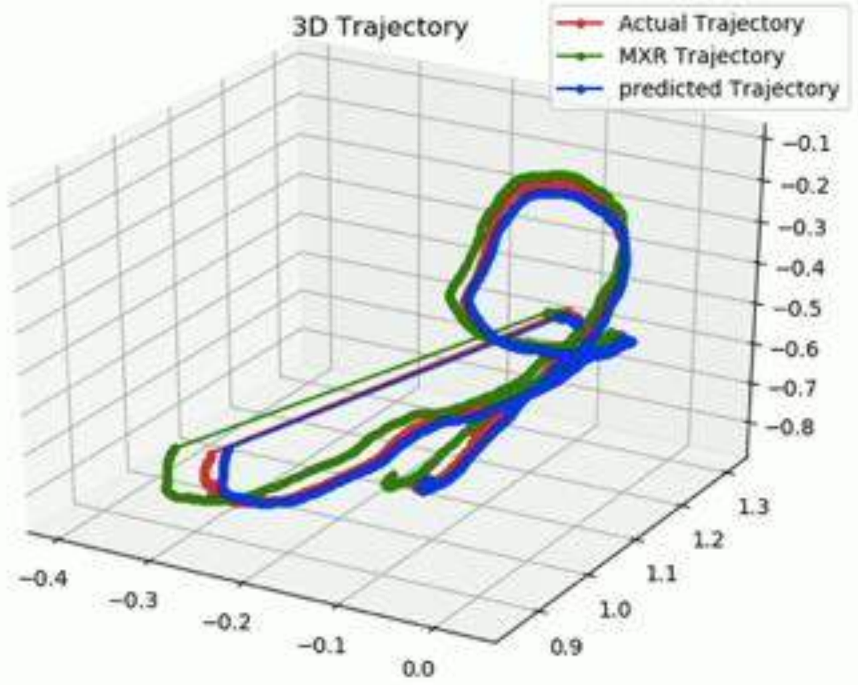
2. Learning-based Model for Tracking

- Tracking performance for Sinusoid and MXR data

Autoregression on sinusoid

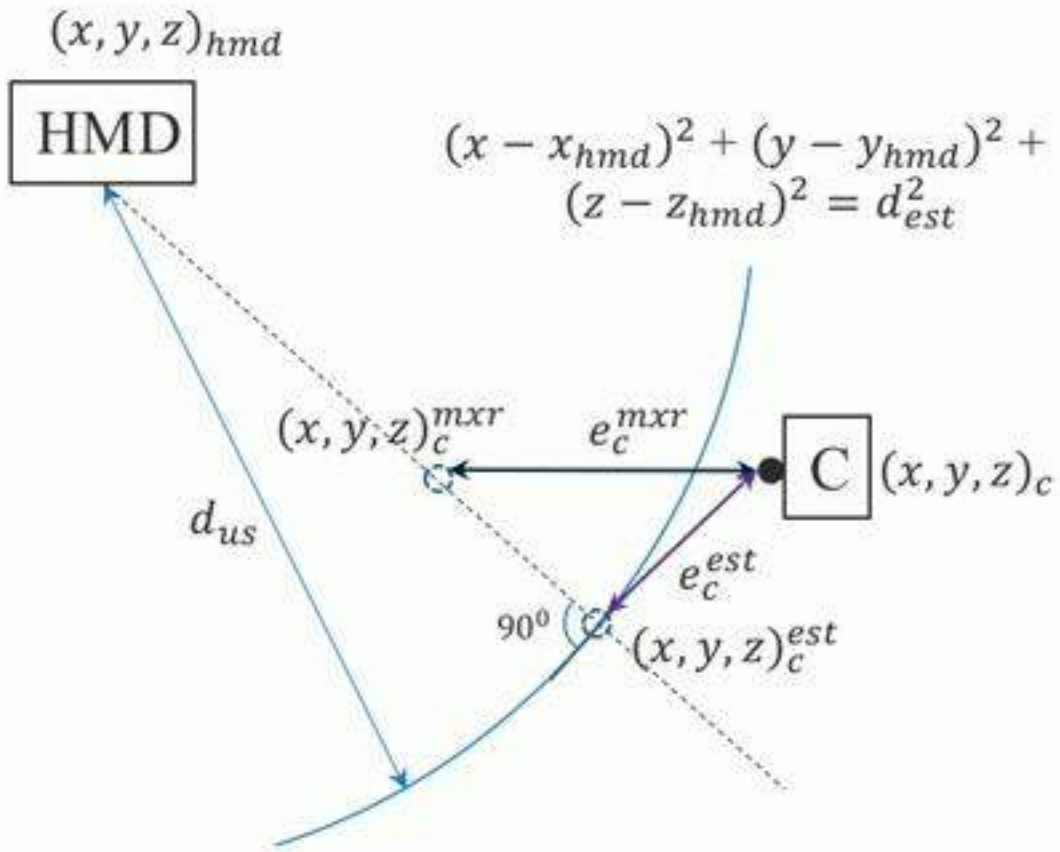


Autoregression on MXR Data



3. Data fusion System for Trajectory Tracking

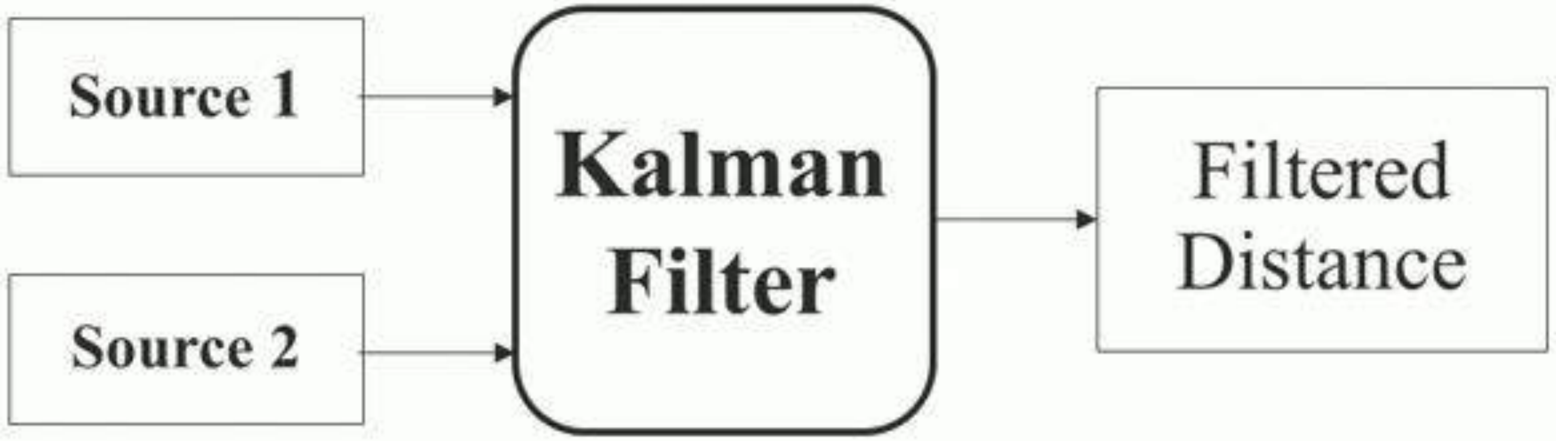
- Spherical Projection to get the estimated position



Given d_{est}, $(x, y, z)_c^{est} = \frac{d_{est}}{d} * [(x_c^{mxr} - x_{hmd}), (y_c^{mxr} - y_{hmd}), (z_c^{mxr} - z_{hmd})] + [x_{hmd}, y_{hmd}, z_{hmd}]$

3. Data fusion System for Trajectory Tracking

- How to get the d_{est}
  - Source 1: Ultrasound ranging
  - Source 2: LSTM model/MXR



Kalman Filter

- Predict: $\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_k \hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_k \mathbf{u}_k$

  $\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^{\mathrm{T}} + \mathbf{Q}_k$

- Update: $\tilde{\mathbf{y}}_k = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}$

  $\mathbf{S}_k = \mathbf{R}_k + \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^{\mathrm{T}}$

  $\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^{\mathrm{T}} \mathbf{S}_k^{-1}$

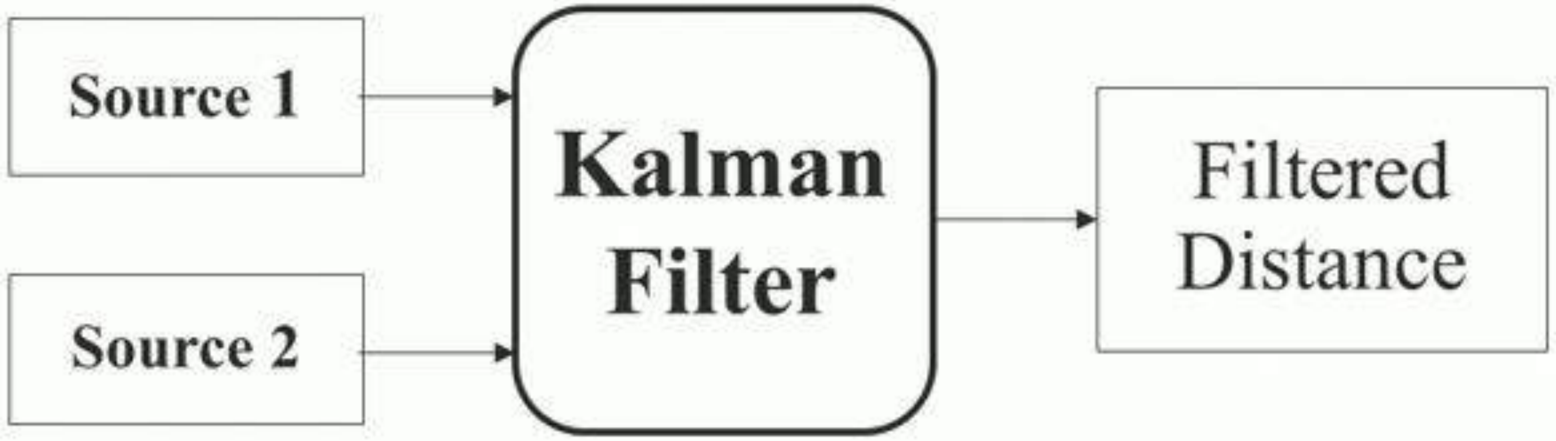
  $\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \tilde{\mathbf{y}}_k$

  $\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1} (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^T + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^T$

  $\tilde{\mathbf{y}}_{k|k} = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k}$

3. Data fusion System for Trajectory Tracking

- How to get the d_{est}
  - Source 1: Ultrasound ranging
  - Source 2: LSTM model/MXR



Kalman Filter

- Predict:

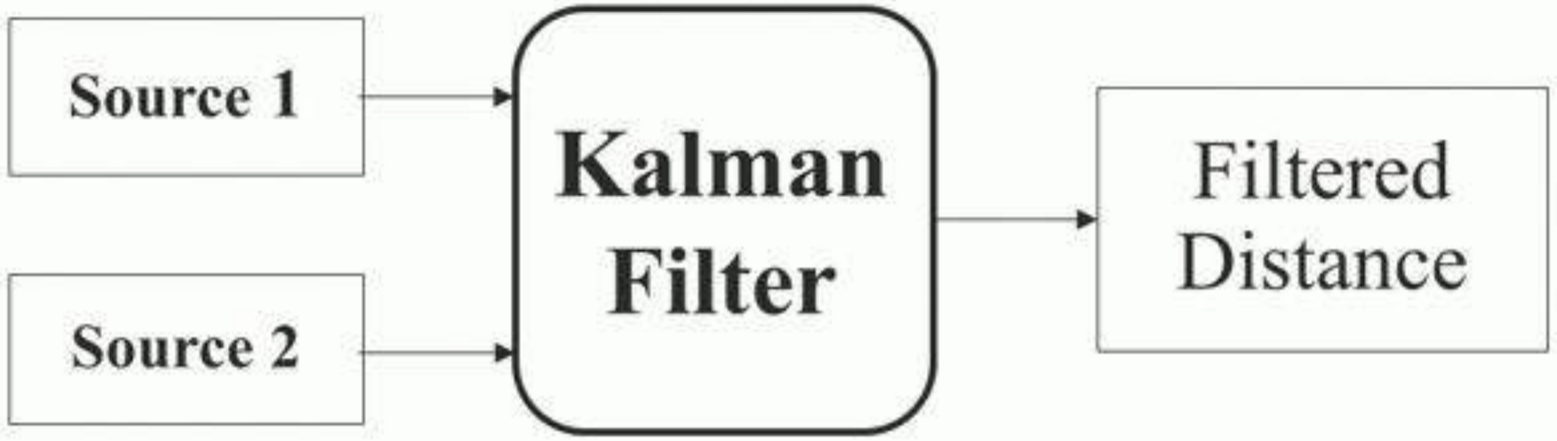
$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_k \boxed{\hat{\mathbf{x}}_{k-1|k-1}} + \mathbf{B}_k \mathbf{u}_k$$
$$\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^{\mathrm{T}} + \mathbf{Q}_k$$

- Update:

$$\tilde{\mathbf{y}}_k = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}$$
$$\mathbf{S}_k = \mathbf{R}_k + \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^{\mathrm{T}}$$
$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^{\mathrm{T}} \mathbf{S}_k^{-1}$$
$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \tilde{\mathbf{y}}_k$$
$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1} (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^T + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^T$$
$$\tilde{\mathbf{y}}_{k|k} = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k}$$

3. Data fusion System for Trajectory Tracking

- How to get the d_{est}
  - Source 1: Ultrasound ranging
  - Source 2: LSTM model/MXR



Kalman Filter

- Predict: $\boxed{\hat{\mathbf{x}}_{k|k-1}} = \mathbf{F}_k \hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_k \mathbf{u}_k$

  $\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^{\mathrm{T}} + \mathbf{Q}_k$

- Update: $\tilde{\mathbf{y}}_k = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}$

  $\mathbf{S}_k = \mathbf{R}_k + \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^{\mathrm{T}}$

  $\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^{\mathrm{T}} \mathbf{S}_k^{-1}$

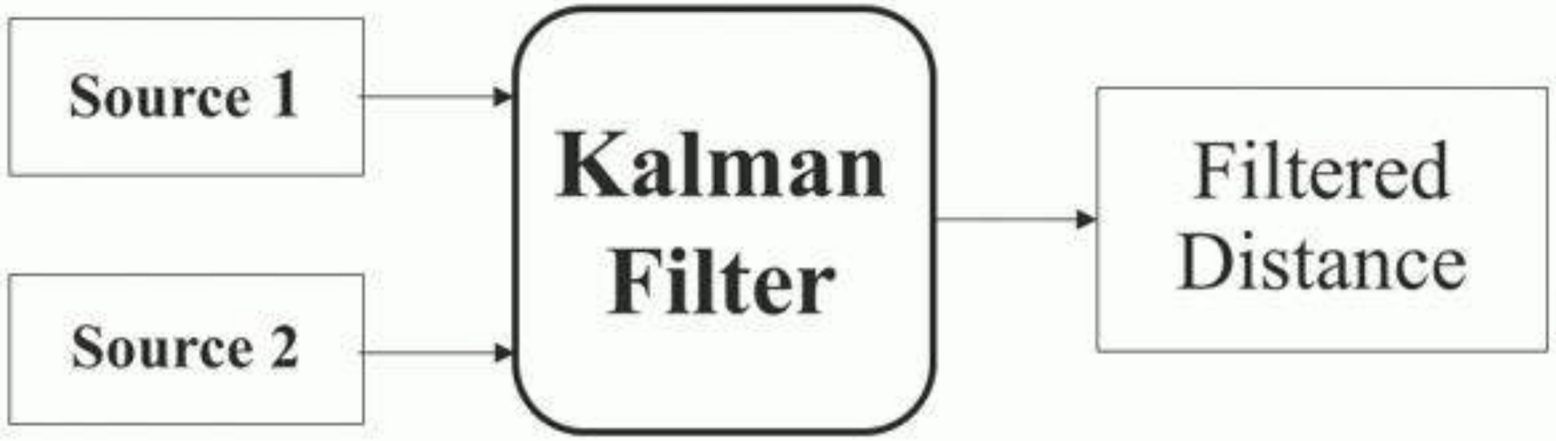
  $\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \tilde{\mathbf{y}}_k$

  $\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1} (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^T + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^T$

  $\tilde{\mathbf{y}}_{k|k} = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k}$

3. Data fusion System for Trajectory Tracking

- How to get the d_{est}
  - Source 1: Ultrasound ranging
  - Source 2: LSTM model/MXR



Kalman Filter

- Predict: $\boxed{\hat{\mathbf{x}}_{k|k-1}} = \mathbf{F}_k \hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_k \mathbf{u}_k$

  $\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^{\mathrm{T}} + \mathbf{Q}_k$

- Update: $\tilde{\mathbf{y}}_k = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}$

  $\mathbf{S}_k = \mathbf{R}_k + \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^{\mathrm{T}}$

  $\boxed{\mathbf{K}_k} = \mathbf{P}_{k|k-1} \mathbf{H}_k^{\mathrm{T}} \mathbf{S}_k^{-1}$

  $\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \tilde{\mathbf{y}}_k$

  $\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1} (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^T + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^T$

  $\tilde{\mathbf{y}}_{k|k} = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k}$

3. Data fusion System for Trajectory Tracking

- Pipeline

Tracking Status (every 20ms)

2	**1**	**2**	**1**	**1**	**1**	**1**	**…**	**2**	**2**	**2**

MXR Prediction (every 20ms)

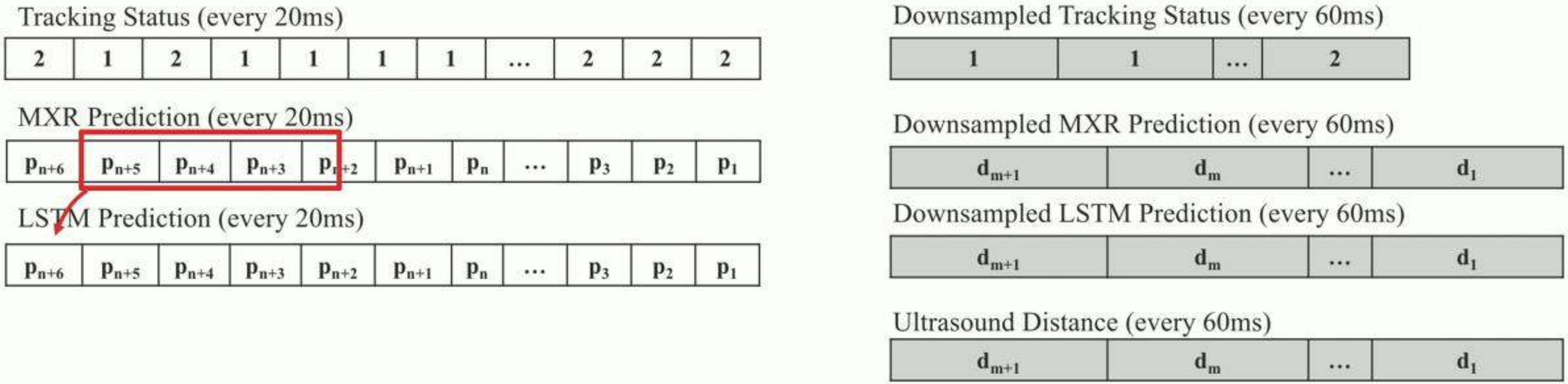
p_{n+6}	p_{n+5}	p_{n+4}	p_{n+3}	p_{n+2}	p_{n+1}	p_n	…	p_3	p_2	p_1

LSTM Prediction (every 20ms)

p_{n+6}	p_{n+5}	p_{n+4}	p_{n+3}	p_{n+2}	p_{n+1}	p_n	…	p_3	p_2	p_1

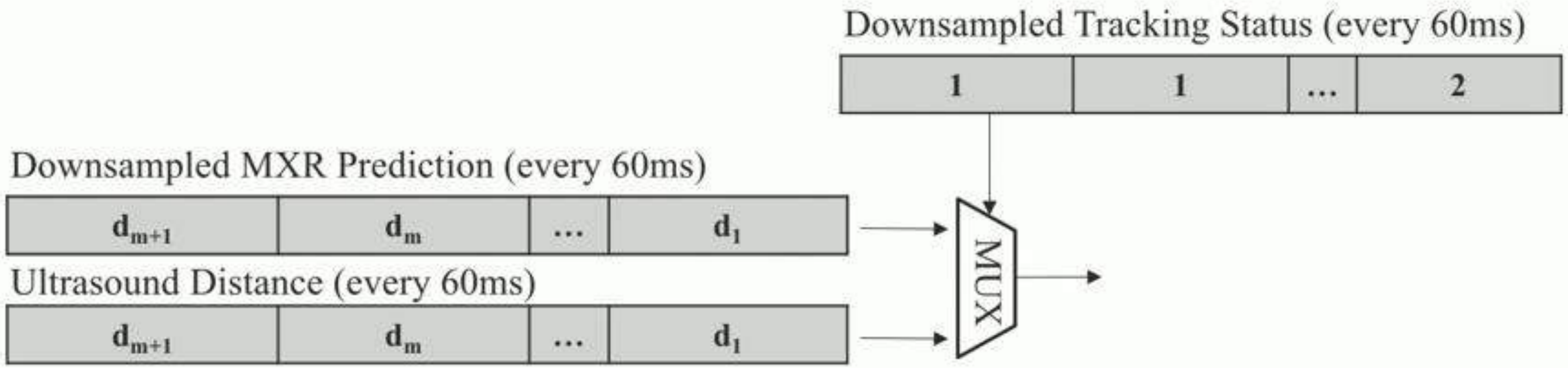
3. Data fusion System for Trajectory Tracking

- Pipeline



3. Data fusion System for Trajectory Tracking

- Pipeline



3. Data fusion System for Trajectory Tracking

- Pipeline

Tracking Status (every 20ms)

2	1	2	1	1	1	1	…	2	2	2

MXR Prediction (every 20ms)

p_{n+6}	p_{n+5}	p_{n+4}	p_{n+3}	p_{n+2}	p_{n+1}	p_n	…	p_3	p_2	p_1

LSTM Prediction (every 20ms)

p_{n+6}	p_{n+5}	p_{n+4}	p_{n+3}	p_{n+2}	p_{n+1}	p_n	…	p_3	p_2	p_1

Downsampled Tracking Status (every 60ms)

1	1	…	2

Downsampled MXR Prediction (every 60ms)

d_{m+1}	d_m	…	d_1

Downsampled LSTM Prediction (every 60ms)

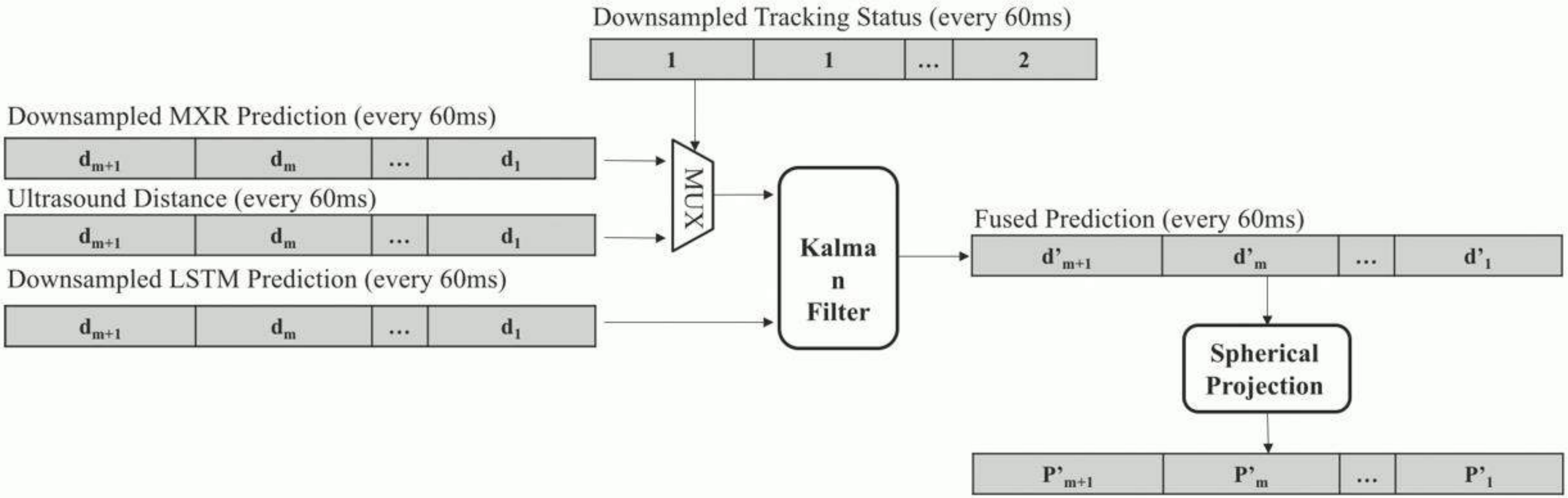
d_{m+1}	d_m	…	d_1

Ultrasound Distance (every 60ms)

d_{m+1}	d_m	…	d_1

3. Data fusion System for Trajectory Tracking

- Pipeline



3. Data fusion System for Trajectory Tracking

- Pipeline

Tracking Status (every 20ms)

2	1	2	1	1	1	1	…	2	2	2

MXR Prediction (every 20ms)

p_{n+6}	p_{n+5}	p_{n+4}	p_{n+3}	p_{n+2}	p_{n+1}	p_n	…	p_3	p_2	p_1

LSTM Prediction (every 20ms)

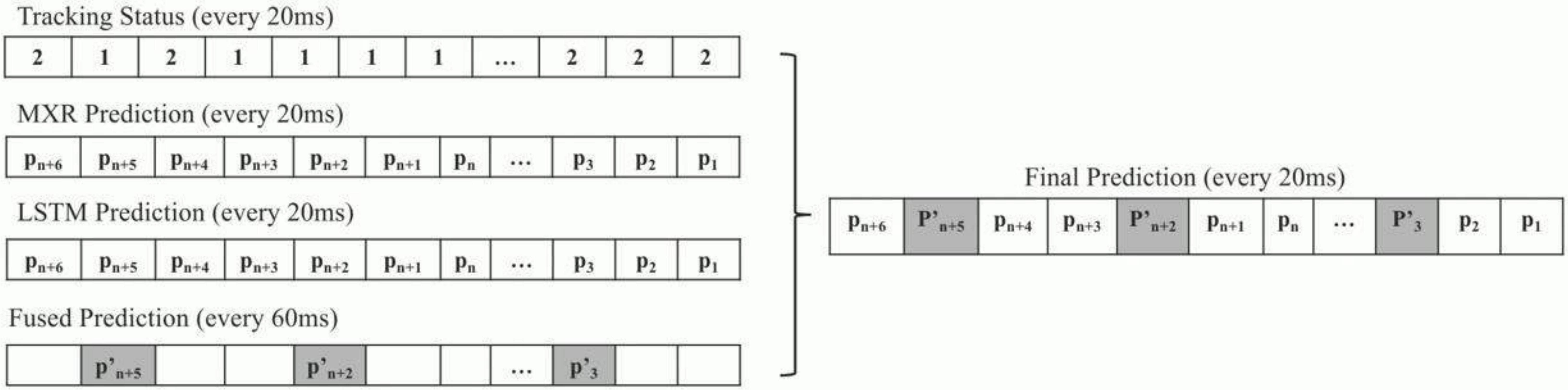
p_{n+6}	p_{n+5}	p_{n+4}	p_{n+3}	p_{n+2}	p_{n+1}	p_n	…	p_3	p_2	p_1

Fused Prediction (every 60ms)

	p'_{n+5}			p'_{n+2}			…	p'_3		

3. Data fusion System for Trajectory Tracking

- Pipeline



3. Data fusion System for Trajectory Tracking

- Offset-based Correction

Tracking Status (every 20ms)

2	1	2	1	1	1	1	…	2	2	2

MXR Prediction (every 20ms)

p_{n+6}	p_{n+5}	p_{n+4}	p_{n+3}	p_{n+2}	p_{n+1}	p_n	…	p_3	p_2	p_1

LSTM Prediction (every 20ms)

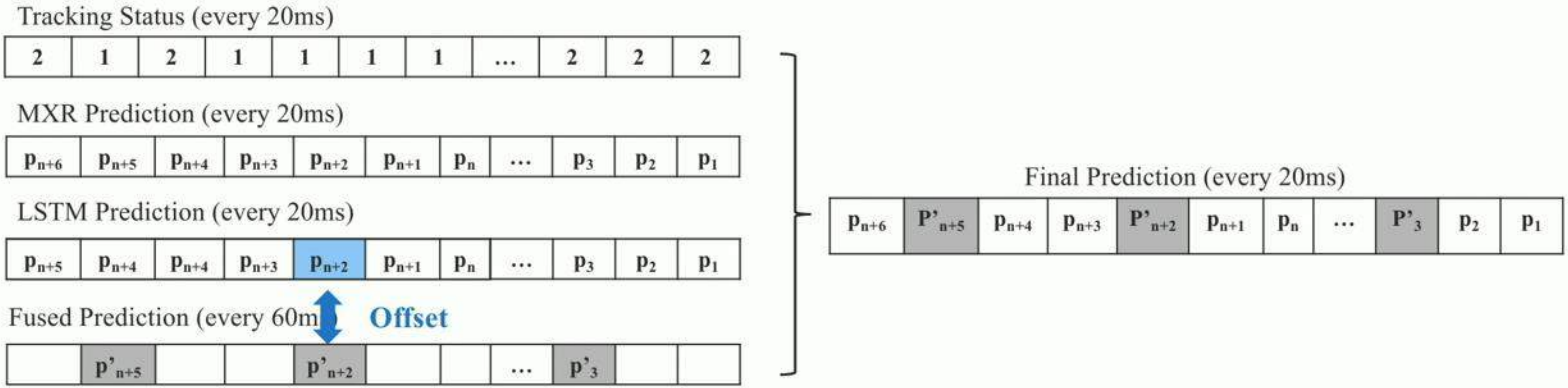
p_{n+6}	p_{n+5}	p_{n+4}	p_{n+3}	p_{n+2}	p_{n+1}	p_n	…	p_3	p_2	p_1

Fused Prediction (every 60ms)

	p'_{n+5}			p'_{n+2}			…	p'_3		

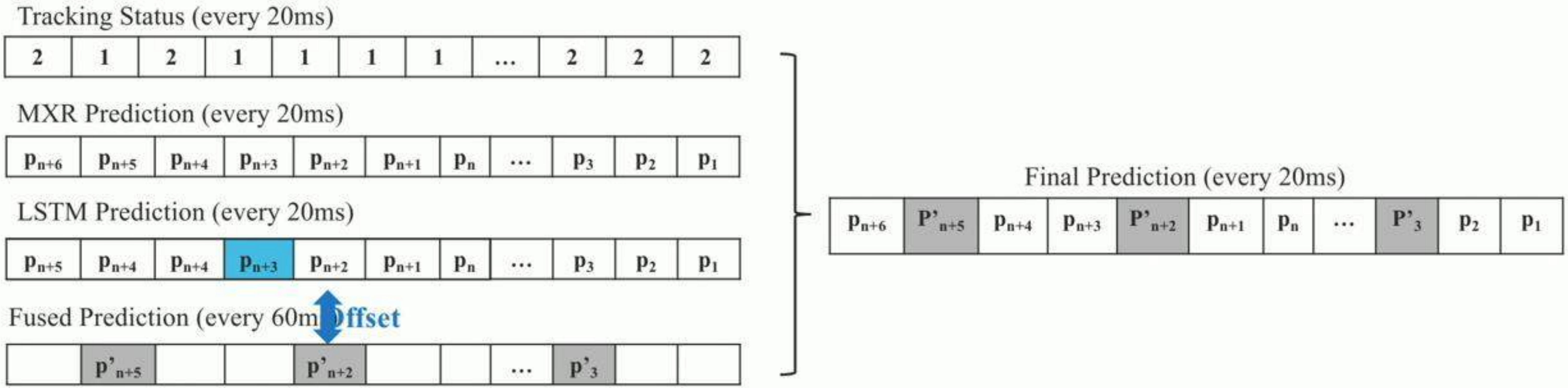
3. Data fusion System for Trajectory Tracking

- Offset-based Correction



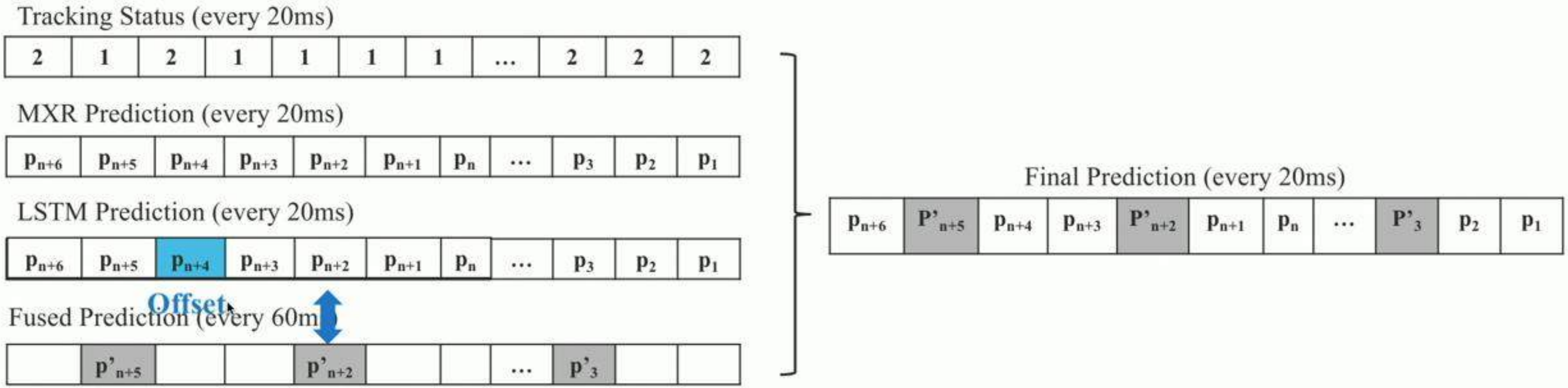
3. Data fusion System for Trajectory Tracking

- Offset-based Correction

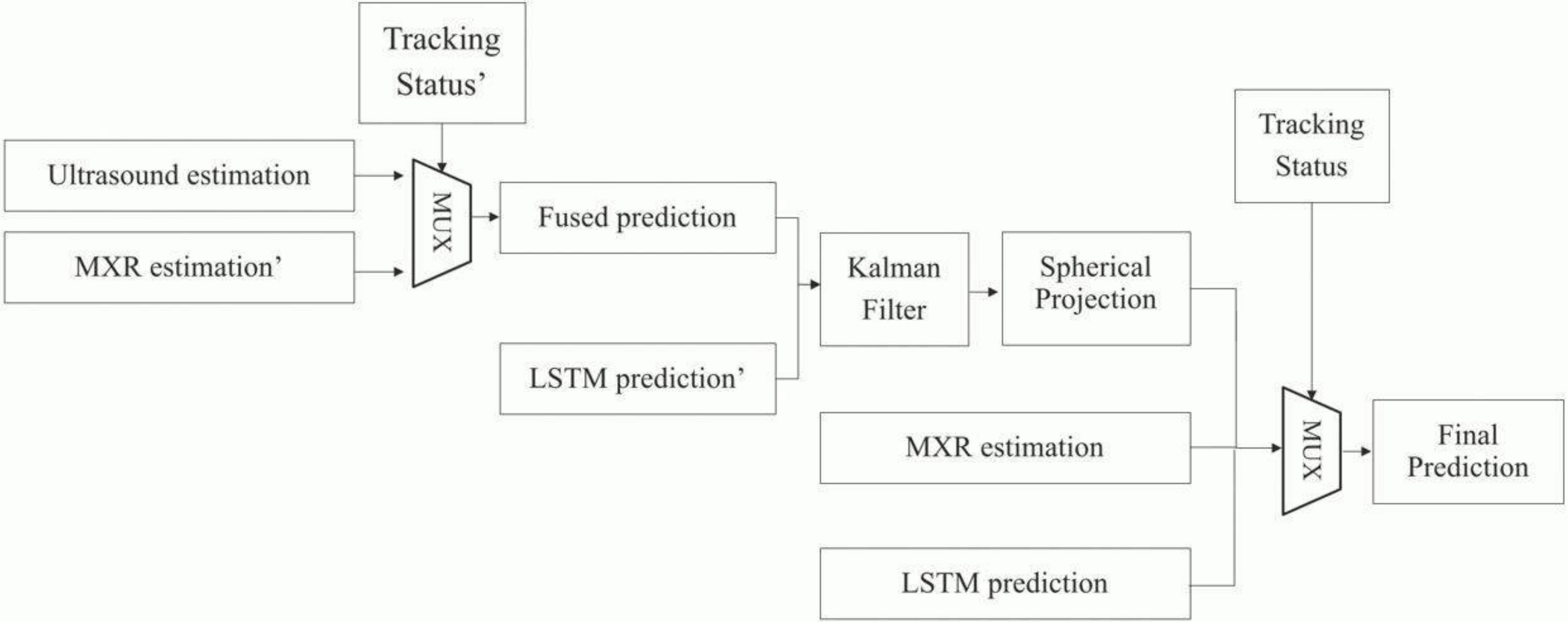


3. Data fusion System for Trajectory Tracking

- Offset-based Correction

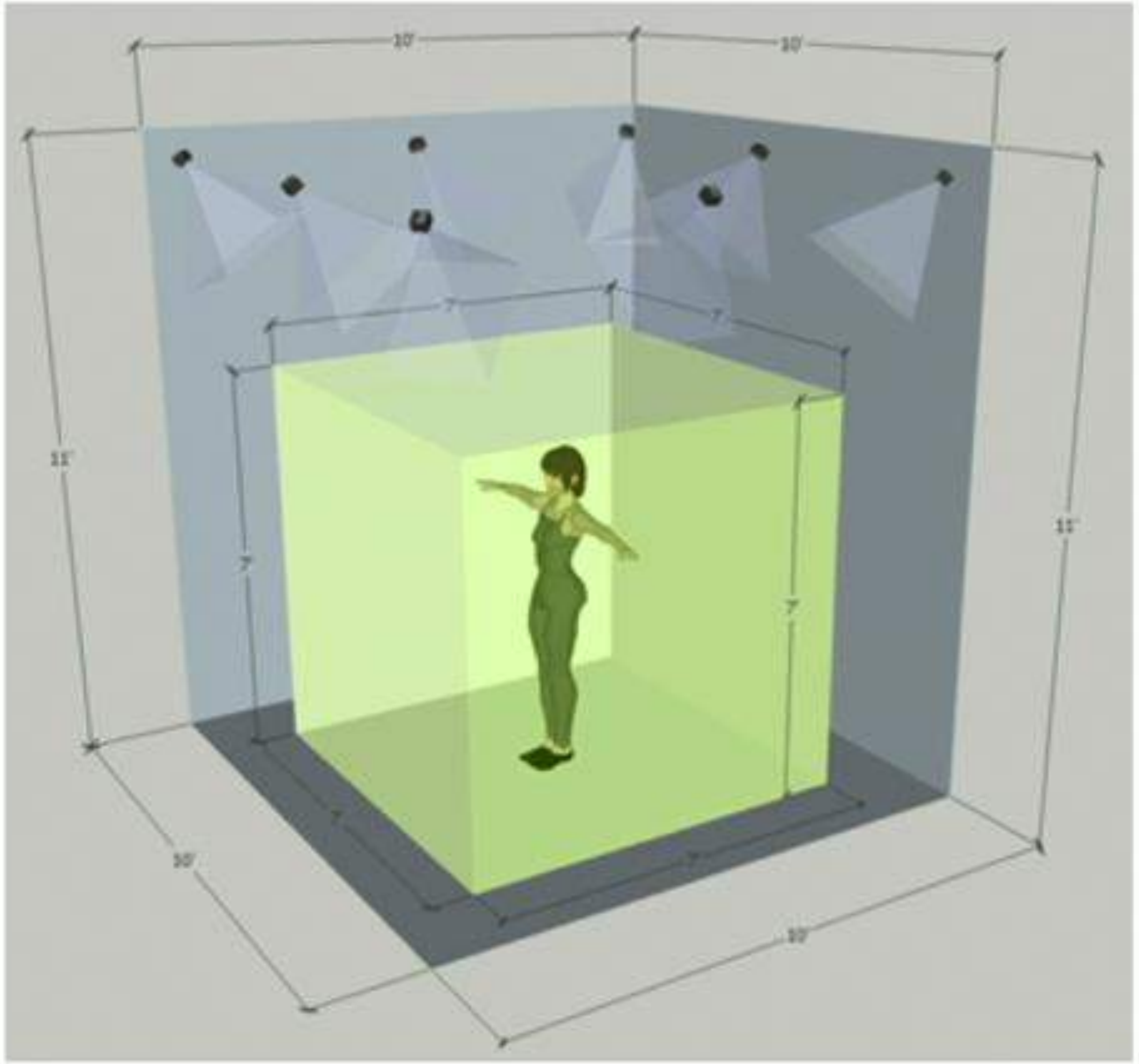


3. Data fusion System for Trajectory Tracking



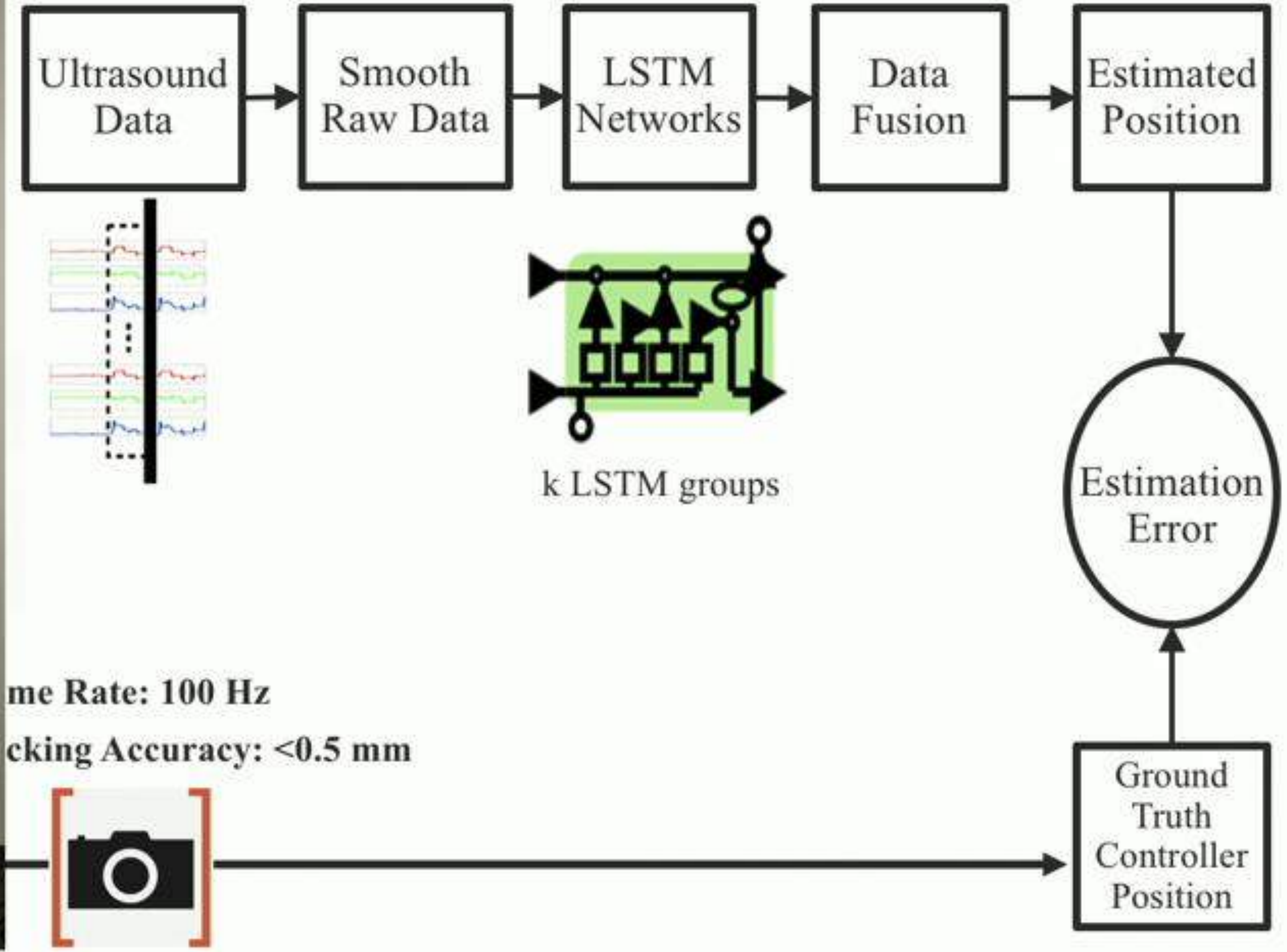
Data Collection Setup

- Motion capture
  - 8-camera OptiTrack, Windows Mixed Reality, in-FOV tracking
- Additional Information
  - Ultrasound Sensor, NN-based Model



Flex 3, OptiTrack System

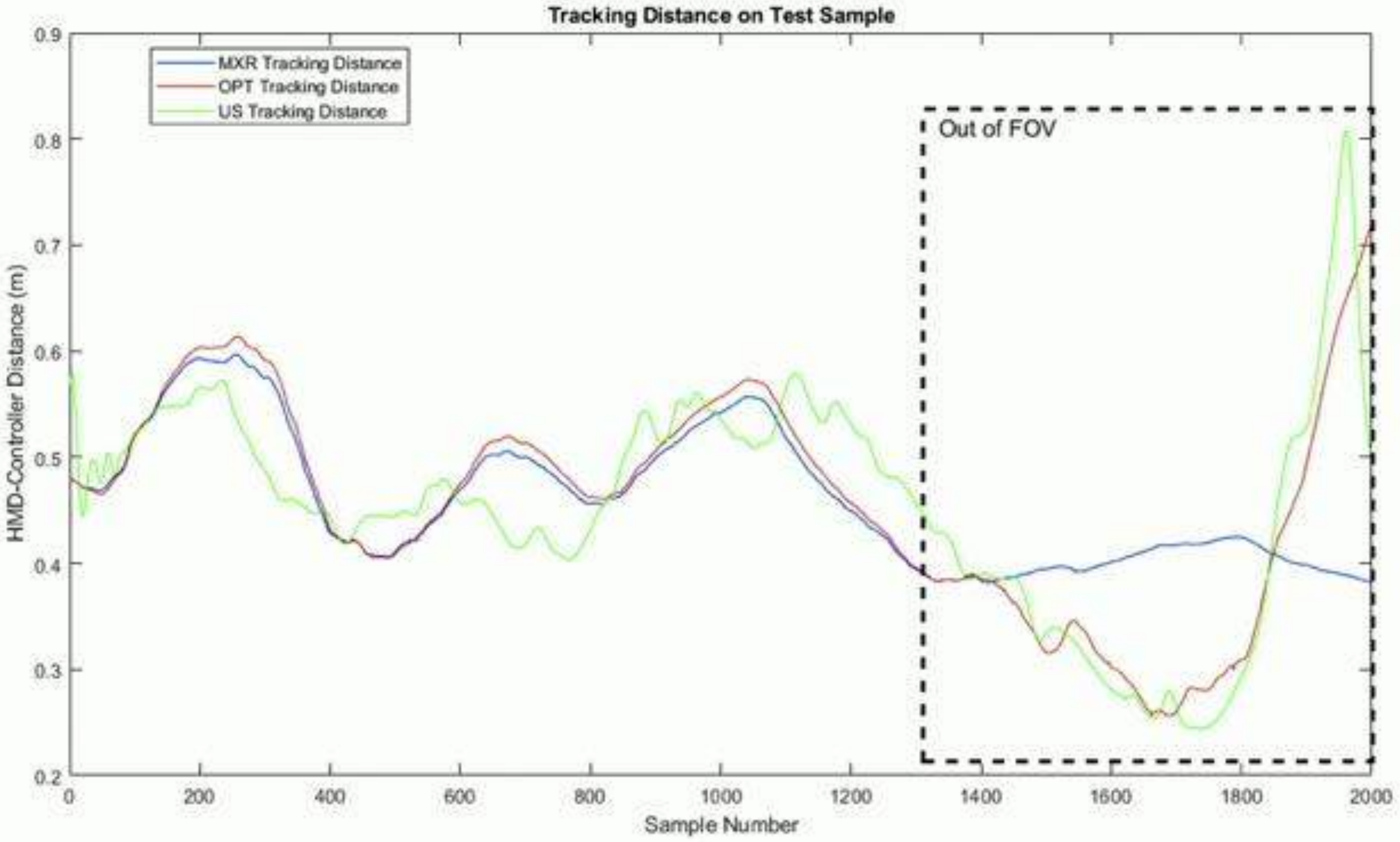
Data Collection Framework



Dataset	
Total no. trials	10
Avg. length of each trial	12 mins
Total length of data	2 hours
Ultrasound sampling freq.	60 Hz
OptiTrack sampling freq.	100 Hz
Mixed reality sampling freq.	50-60 Hz
Ground truth collection	8 Flex3 Camera + Motive 2.0.2

Tracking Performance

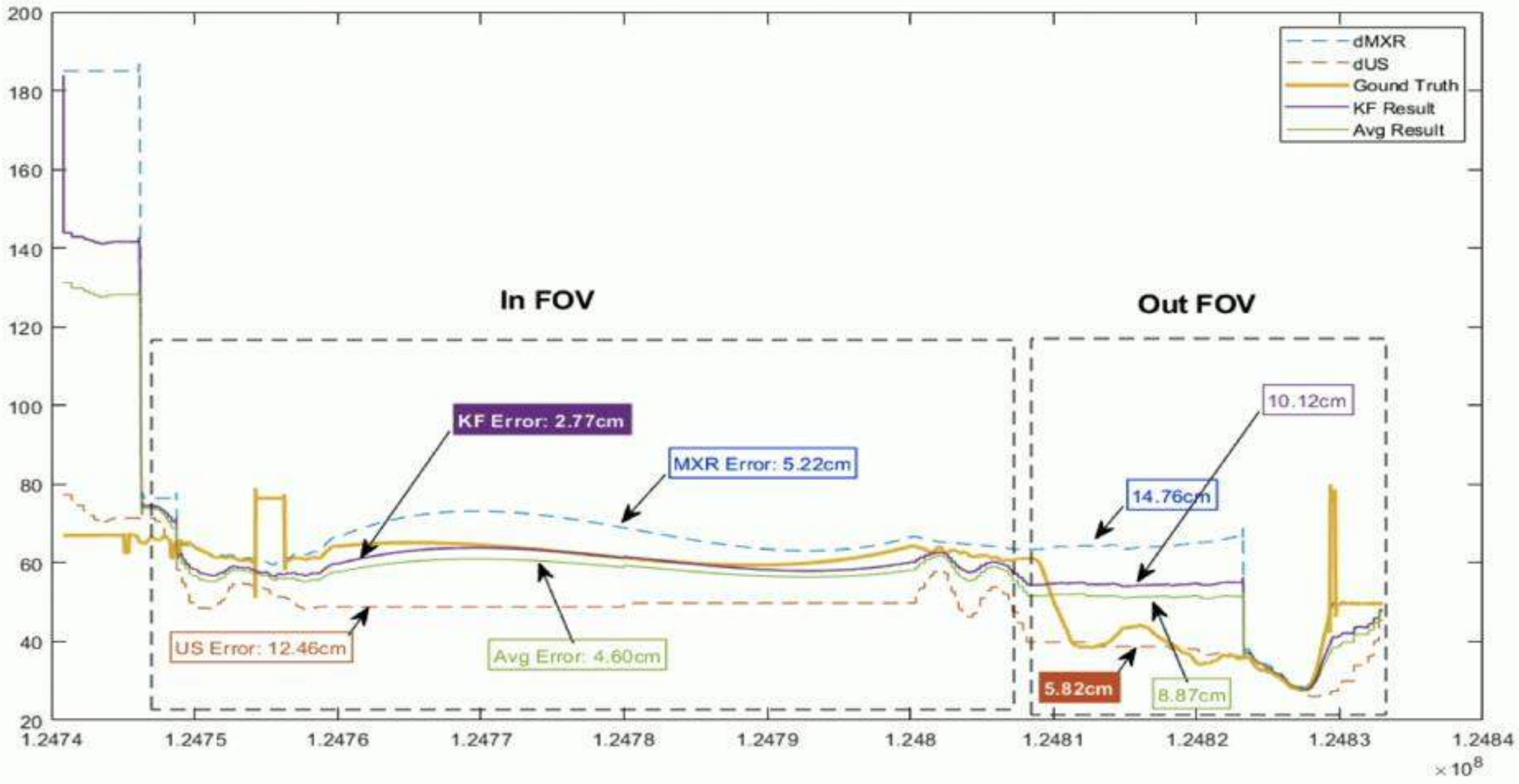
- Ultrasound distance tracking



Average Tracking Distance Error (cm)		
In FOV (Status High)	*MXR*	*3.4859+3.0068*
	US	*8.7898+6.208*
Out FOV (Status Approx)	*MXR*	*23.5274+14.2919 (depends on the movement)*
	US	*8.4739+8.1515*

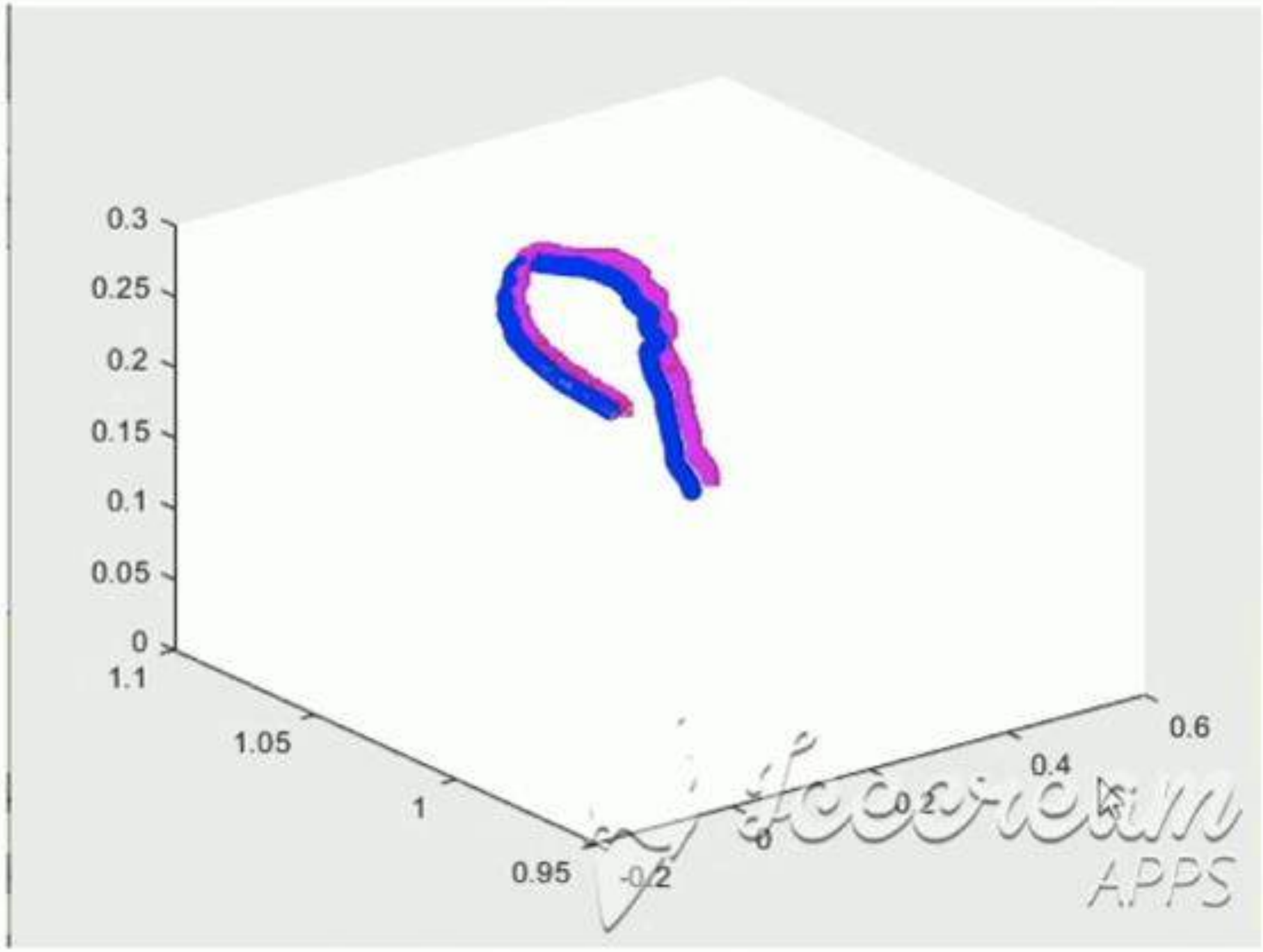
Tracking Performance

- Ultrasound distance tracking



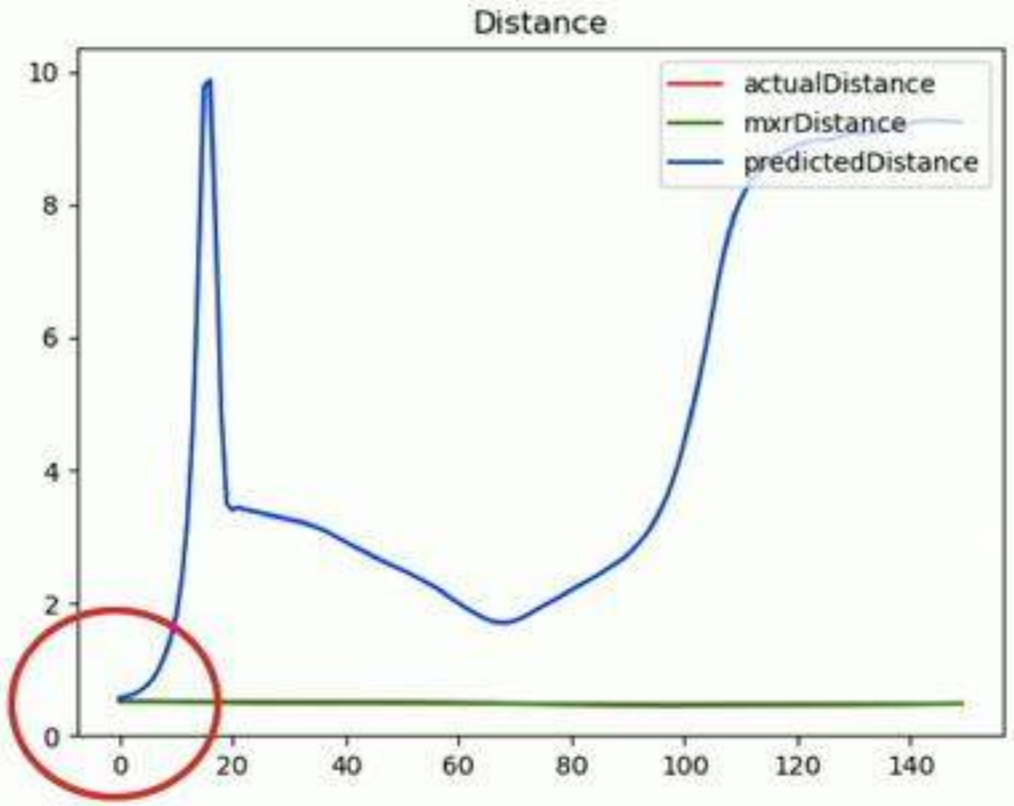
Tracking Performance

- LSTM tracking in FOV

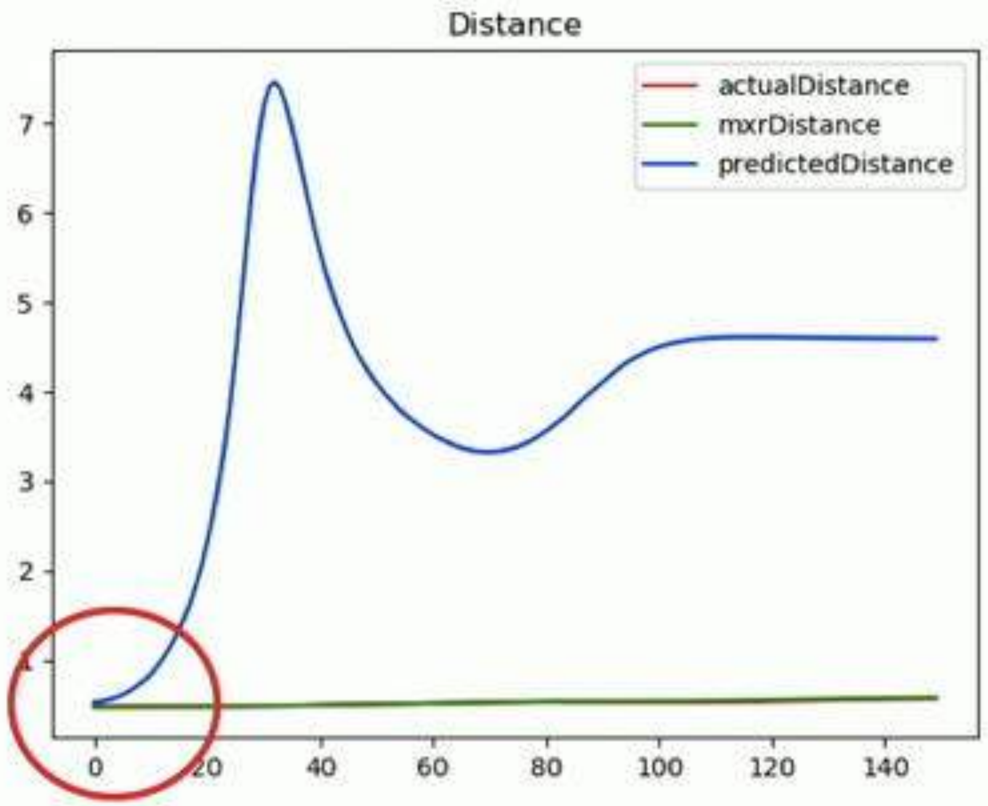


Tracking Performance – Out of FOV

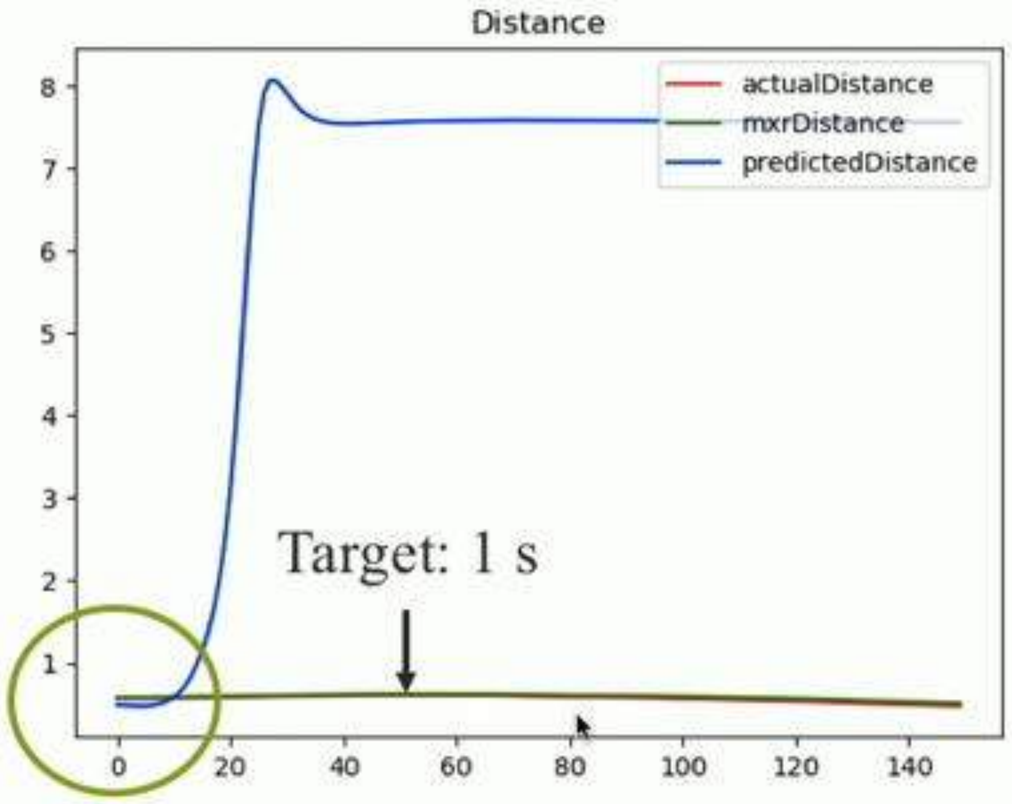
- LSTM tracking with different window size



Window Length = 51
(Using the past 1 s info)



Window Length = 101
(Using the past 2 s info)



Window Length = 201
(Using the past 4 s info)

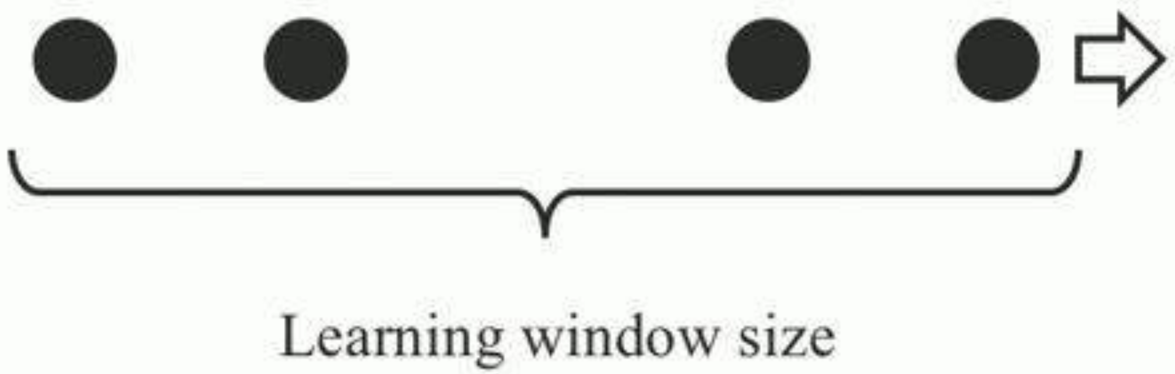
Able to predict up to 200 ms

Ongoing: Longer Out of FOV Tracking

- De-trend data, remove non-stationarity
- Larger models, hierarchical LSTMs, Windowed MLP
- Other forecasting strategies
  - Trade off computation for tracking accuracy
  - Direct H-step ahead forecasting, multiple input multiple output models

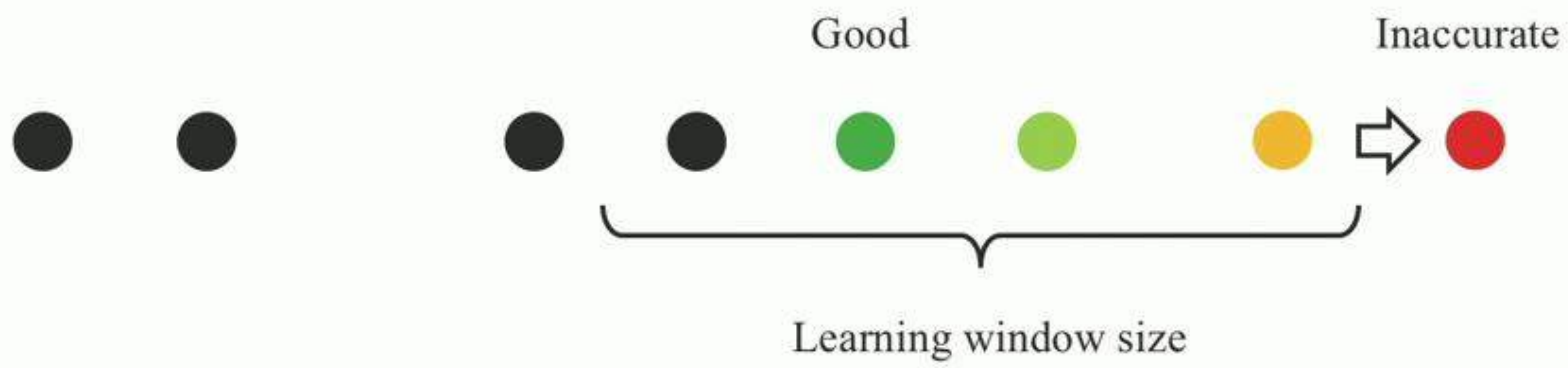
Ongoing: Longer Out of FOV Tracking

- Direct H-step ahead forecasting



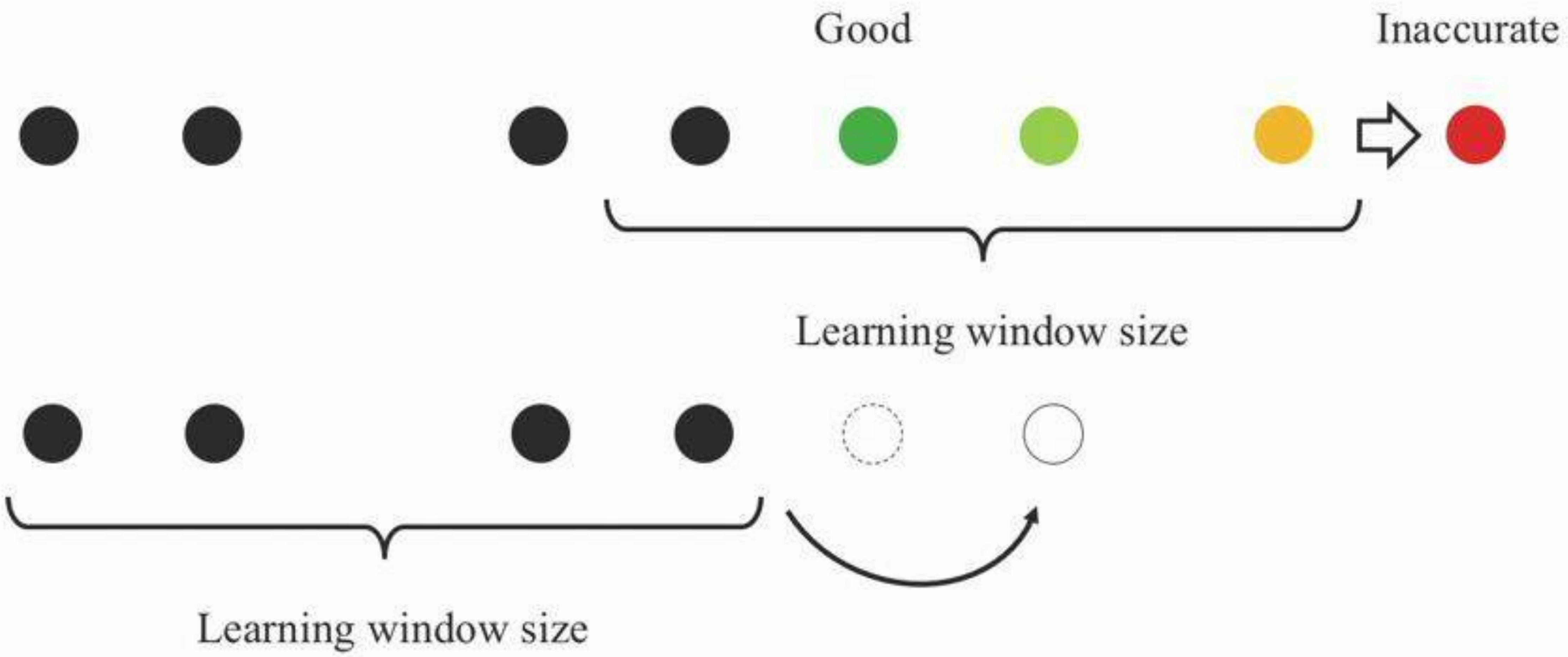
Ongoing: Longer Out of FOV Tracking

- Direct H-step ahead forecasting



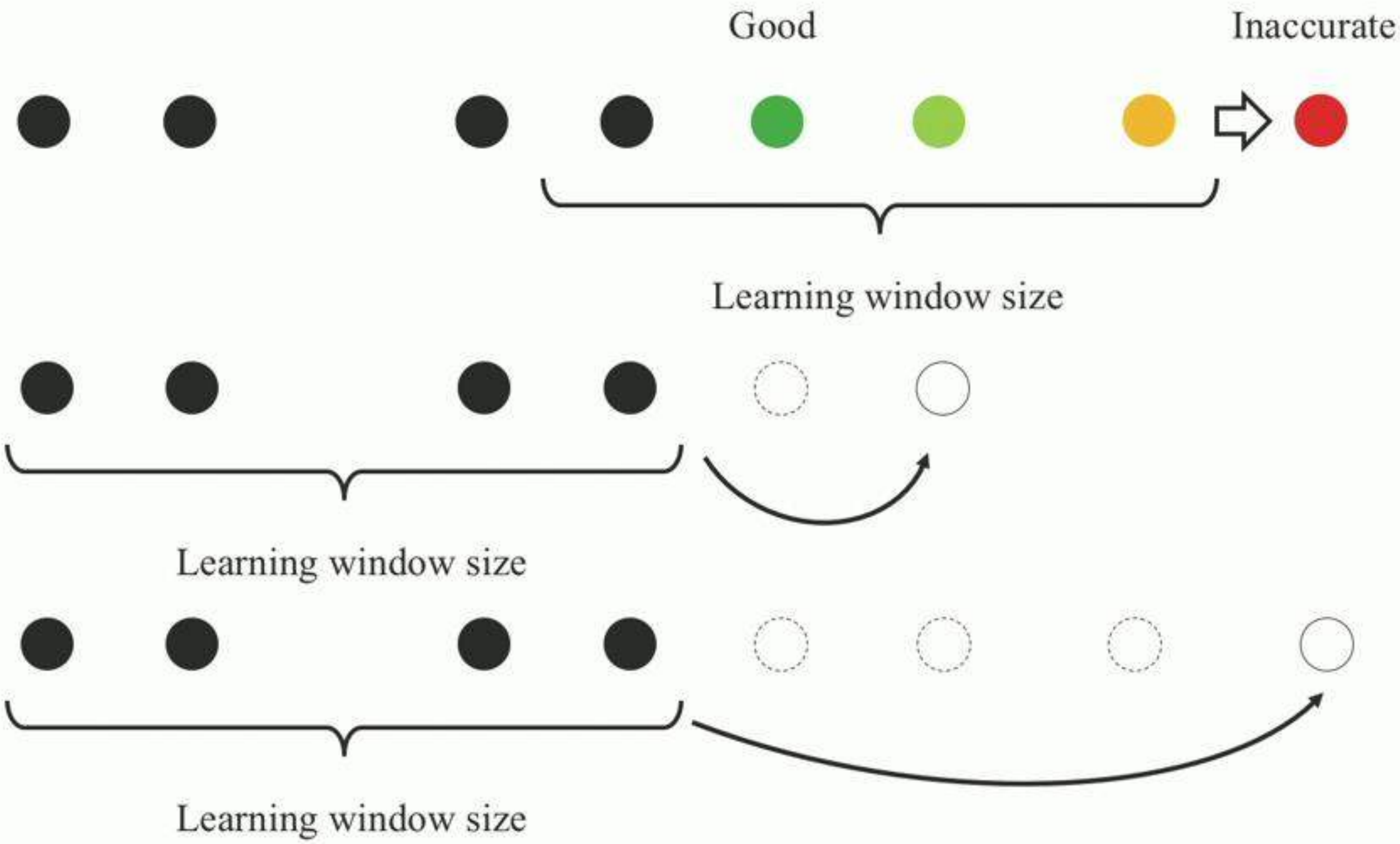
Ongoing: Longer Out of FOV Tracking

- Direct H-step ahead forecasting



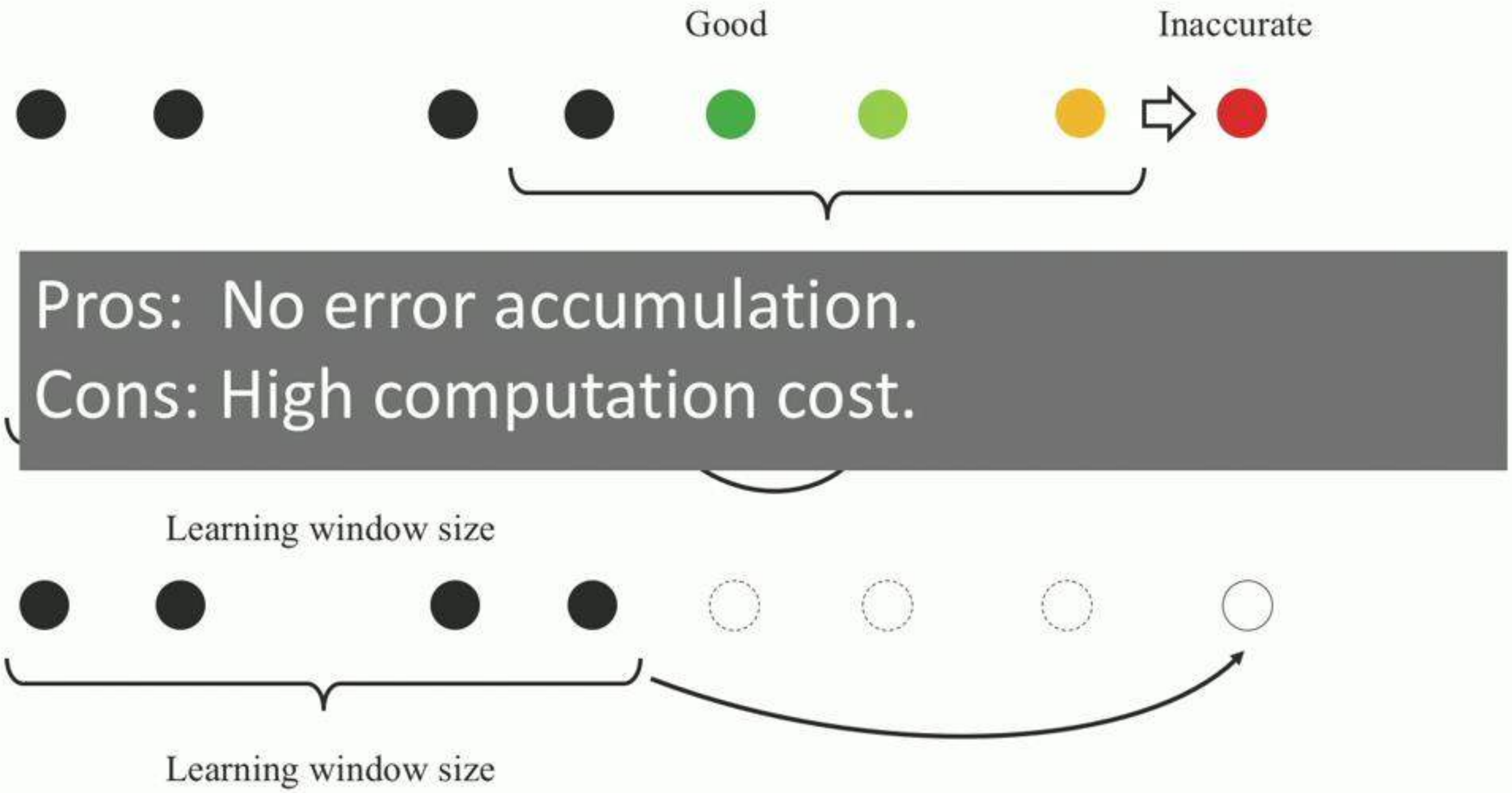
Ongoing: Longer Out of FOV Tracking

- Direct H-step ahead forecasting



Ongoing: Longer Out of FOV Tracking

- Direct H-step ahead forecasting



Ongoing: Longer Out of FOV Tracking

- Multiple input multiple output models

Ongoing: Longer Out of FOV Tracking

- Multiple input multiple output models

Pros: 1. Only one model is trained.
2. No error accumulation.

Cons: 1. Prediction length is fixed. Less flexibility.

Conclusions

- Built an ultrasound ranging system
  - Distance accuracy of around 8 cm
- Develop an iterative time series prediction model based on LSTM
  - Prediction range up to 200 ms
- Establish a sensing fusion system which fuses the available data to estimate the position information.

Acknowledgement

- My mentor Shuayb Zarar
- Audio and Acoustics Research Group
- Microsoft Research

Microsoft

Thank You