Introduction

- IoRL Indoor Positioning Overview
- System Latencies
- Kalman filtering
- Model Variables
- Results and Analysis

IoRL Positioning Overview

- Mmwave TDOA performed in the uplink channel by RAN within the RRLHC
- VLC RSS measured in the downlink channel
- Location Database (LD) stores respective measurement data
- Location Server (LS) generates user position estimates
- Estimates retrieved by User Equipment (UE)

Location Database

- Anchor points – priory known antenna coordinates
- Mmwave Data – TDOA collected via the RAN
- VLC Data – RSS measured at UE
- Position estimates – calculated by the Location Server

Location Server

- Utilises Kalman Filter for data fusion of estimates

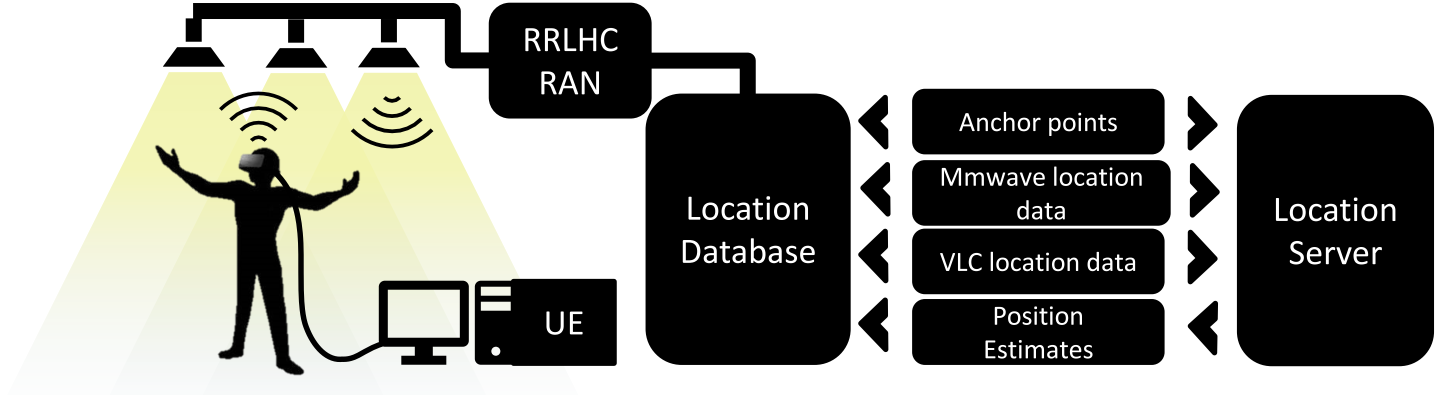


*Figure 8.*Overview of IoRL positioning system..

Location Server sequence

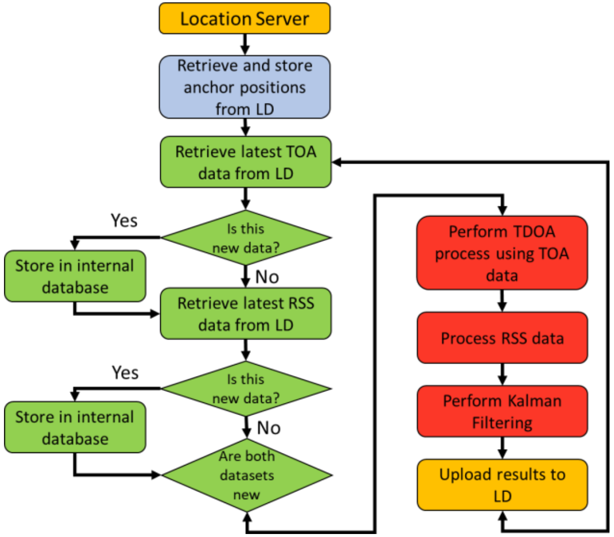
Location Database

- ❑ Anchor points – priory known antenna coordinates
- ❑ Mmwave Data – TDOA collected via the RAN
- ❑ VLC Data – RSS measured at UE
- ❑ Position estimates – calculated by the Location Server

Location Server

- ❑ Utilises Kalman Filter for data fusion of estimates

The checks are performed to reduce computation of redundant data.



IoRL IPS latency

MTPL (Δt_8) = Position estimate (Δt_{1-5}) + Render & display time (Δt_6)

- Render & display time variable depending on scenario complexity.
  - Pessimistically assumed to be 12.5 ms

Position estimate (Δt_{1-5}) = Measurement time (Δt_1) + calculation time(Δt_{2-5})

- **Δt_1** = IoRL proposes TDOA and RSS measurement sets every **2.5ms**
- All other latencies practically measured through execution runtime
- Calculation time determines how often a measurement can be read (**Refresh rate**)

MTPL = 29.2ms , Refresh rate = 12.5ms

TABLE 1. IoRL IPS Component latencies

Latency (Δt_n)	Time (ms)	Representation
Δt_1	2.5	Measurement Time
Δt_2	2.8	RRLHC to LD transmission
Δt_3	1.2	LS data request
Δt_4	5.4	LS Processing
Δt_5	4.8	LD to client
Δt_6	12.5	Render & display
Δt_7	12.5	Frame update delay
Δt_8	26.7	Overall Processing latency

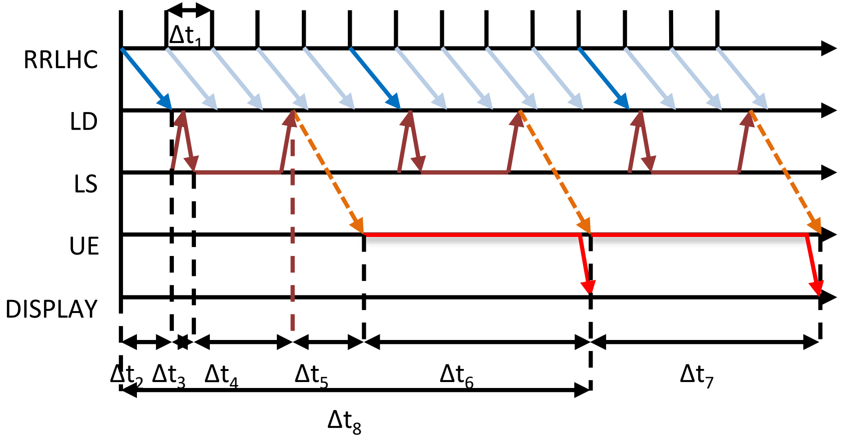
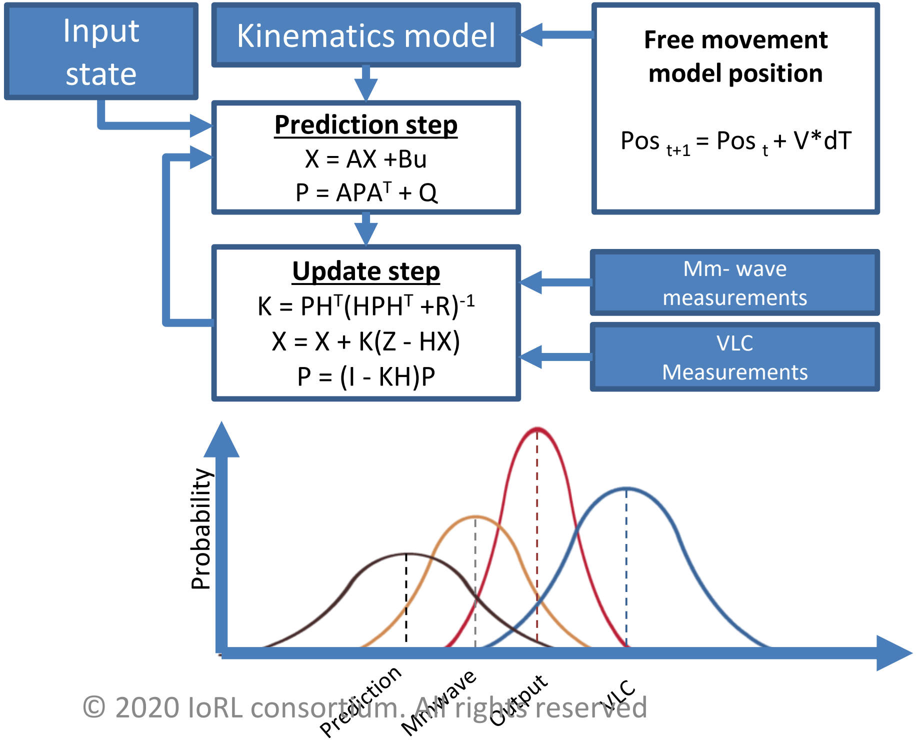


Figure 10. Data transmissions and latencies within the IoRL IPS

Sensor fusion – Kalman filtering

- **Kalman filtering has proven to be a robust tracking filter**
  - Enabling prediction of user movements with no data
  - Considering weightings of uncertainty's in noisy measurements
  - Responsive and non computationally intensive
- **A recursive filter consisting of two stages**
  - Prediction stage - use a physical model to suggest position
  - Update – compare prediction and measurements to evaluate the output



Kalman filter prediction

- Implemented a generic Constant Velocity model
- *Pos(t+1) = Pos(t) + V(t)* Δt*

$$\begin{pmatrix} pos_x \\ pos_y \\ pos_z \\ v_x \\ v_y \\ v_z \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} pos_x \\ pos_y \\ pos_z \\ v_x \\ v_y \\ v_z \end{pmatrix}$$

Prediction step

X = AX +Bu

P = APA^T + Q

- Where Bu = 0

$$Q = \begin{pmatrix} \frac{\Delta t^4}{4}(I_{3x3}) & \frac{\Delta t^3}{2}(I_{3x3}) \\ \frac{\Delta t^3}{2}(I_{3x3}) & \frac{\Delta t^4}{4}(I_{3x3}) \end{pmatrix} * \sigma_q^2$$

- Variance of acceleration (σ_q^2) = 0.3 from testing

Kalman filter update

Update step

K = $PH^T(HPH^T + R)^{-1}$

X = X + K(Z - HX)

P = (I - KH)P

Update *(repeated sequentially for both mmwave and VLC measurements)*

- Kalman gain $(K) = P_{(k)}H^T\left(HP_{(k)}H^T + R\right)^{-1}$
- State update $(x_{(k)}) = X_{(k)} + K\left(z_{(k)} - HX_{(k)}\right)$
- Covariance update $(P_{(k)}) = (I - KH)P_{(k)}$

Where z and R represent a **Position Estimate** and the respective **covariance matrix**

$$z = \begin{pmatrix} x \\ y \\ z \end{pmatrix}, \qquad R = \begin{pmatrix} {\sigma_x}^2 & 0 & 0 \\ 0 & {\sigma_y}^2 & 0 \\ 0 & 0 & {\sigma_z}^2 \end{pmatrix}$$

- VLC R Values taken directy from ISEP RSS measurement campaign
- $\sigma_{x(vlc)} = 0.1436\,, \sigma_{y(vlc)} = 0.1228\,, \sigma_{z(vlc)} = 0.1332$

- MmWave *R* values calculated by taking the average variance for each axis when simulating TDOA estimation at every 10cm point in the 3m cubed environment.
- $\sigma_{x(mm)} = 0.016, \sigma_{y(mm)} = 0.016, \sigma_{z(mm)} = 0.186$

Kalman filter Initialisation

- Initial values
  - P = Q
  - Initial X state is given as the users actual position
- Delta T value
  - 12.5ms (Location Server Refresh Rate)

Results and Analysis

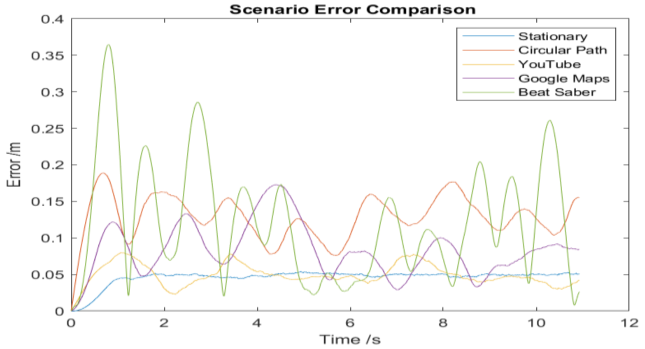
Scenario acceleration and Error

- Strong correlation, as expect, between the two variables
- Due to the constant velocity model used
- Overall average 5-12cm error range

Stabilisation time

- Noticeable immediate rise before oscillating or stabilising
- Concluded this is due to the filter used in a dynamic environment never reaching a steady state
- Initially relying too strongly on the prediction values
- Increased certainty in measurement datasets by decreasing R values by an order of 10 (R/10)

Scenario	Mean Velocity (m/s)	Mean Acceleration (m/s^2)	Mean Error (m)
Stationary	0.034	0.3552	0.0459
YouTube video	0.0783	0.7977	0.0512
Google Maps VR	0.2604	1.5536	0.0848
Circular Path	0.4808	1.6174	0.1294
Game 'Beat Saber'	0.4803	2.7019	0.1230

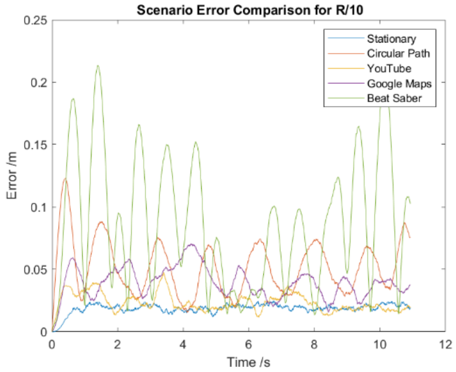


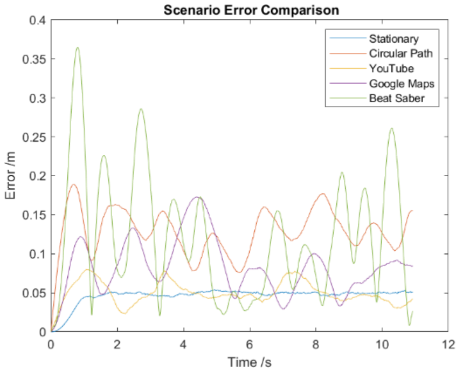
Results and Analysis

Accuracy vs precision tradeoff

- Large reduction in overall average error (approx. 2-8cm error)
- Visibly more irregular error profiles highlighted by increase in average standard deviations
- Reliance on unstable measurements

Scenario	Mean Velocity (m/s)	Mean Acceleration (m/s^2)	Mean Error (m)	Mean Error for R/10 (m)	Standard Deviation (m)	Standard Deviation for R/10 (m)
Stationary	0.034	0.3552	0.0459	0.0188	0.034	0.3552
YouTube video	0.0783	0.7977	0.0512	0.0232	0.0783	0.7977
Google Maps VR	0.2604	1.5536	0.0848	0.0393	0.2604	1.5536
Circular Path	0.4808	1.6174	0.1294	0.0511	0.4808	1.6174
Game 'Beat Saber'	0.4803	2.7019	0.1230	0.0870	0.4803	2.7019





Conclusions

Performance of the IoRL IPS, is impressive for a 5G wireless localization system

- Meets the IoRL localization accuracy target (<10cm)
- Very low end to end latency

- Further study will be required to achieve a more stable tracking response.

Further Work

Alternative filtering techniques and tuning parameters

- Adaptive filters
- Improved kinematics models
- Improved filtering (Particle filters)

Reducing MTPL

- Currently run in python interpreter
- Allows for more frequent measurements and reduced error.
- Investigate reduced Render and Display latency

Measurement error values

- values used are greater than those of other mmwave and VLC localization studies
- Possible IoRL measurement campaigns to explore more realistic values

Complimentary data

- Inertial Measurement Units (IMU) within certain devices

Acknowledgement and disclaimer

- Project IoRL has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 761992

- This presentation reflects the author's view, only, and the Commission is not responsible for any use that may be made of the information provided.