Integrity-Based Fault Detection for GNSS Navigation Safety

Siddharth Tanwar (tanwar@stanford.edu) and Tara Yasmin Mina (tymina@stanford.edu)

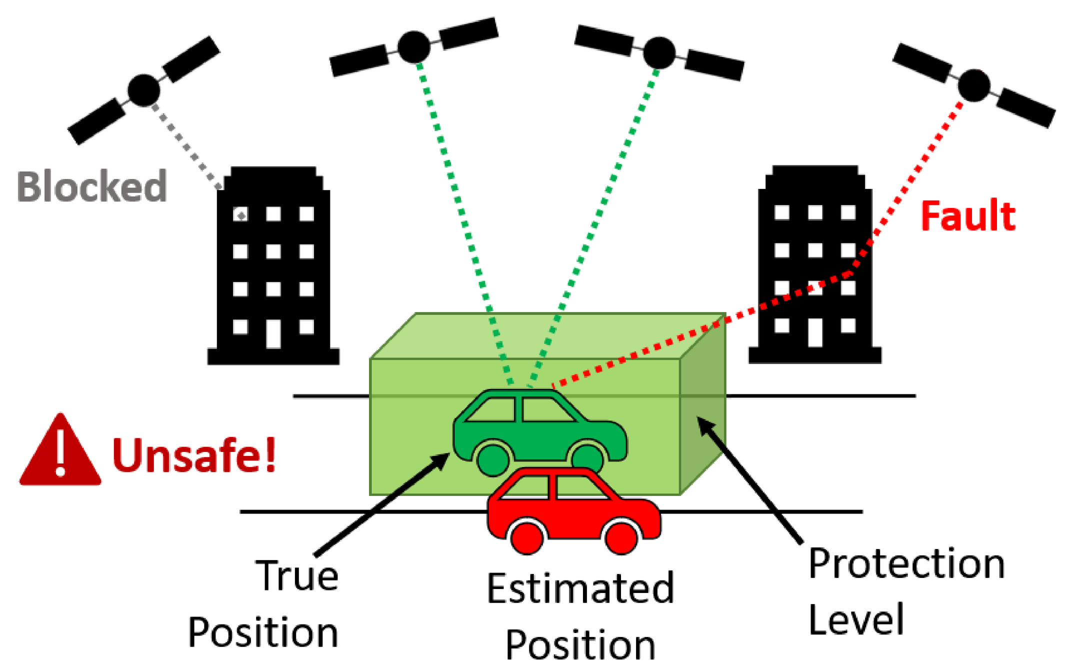
Objective

A key challenge of the autonomous vehicles industry is to guarantee a high level of safety for life-critical navigation applications. Though GNSS is required for global localization, measurements are prone to errors in urban environments. We classify a set of received GNSS measurements as **safe** or **unsafe** for navigation.

GNSS Integrity Definition

Integrity faults occur when state estimate $\hat{z}$ exceeds a **protection level** about the true state z with high probability:

$$P(\hat{z} \notin PL(z)) \geq \epsilon \tag{1}$$



Features

For each visible satellite k, the agent measures a range estimate ρ_k and SNR γ_k. The feature vector x includes:

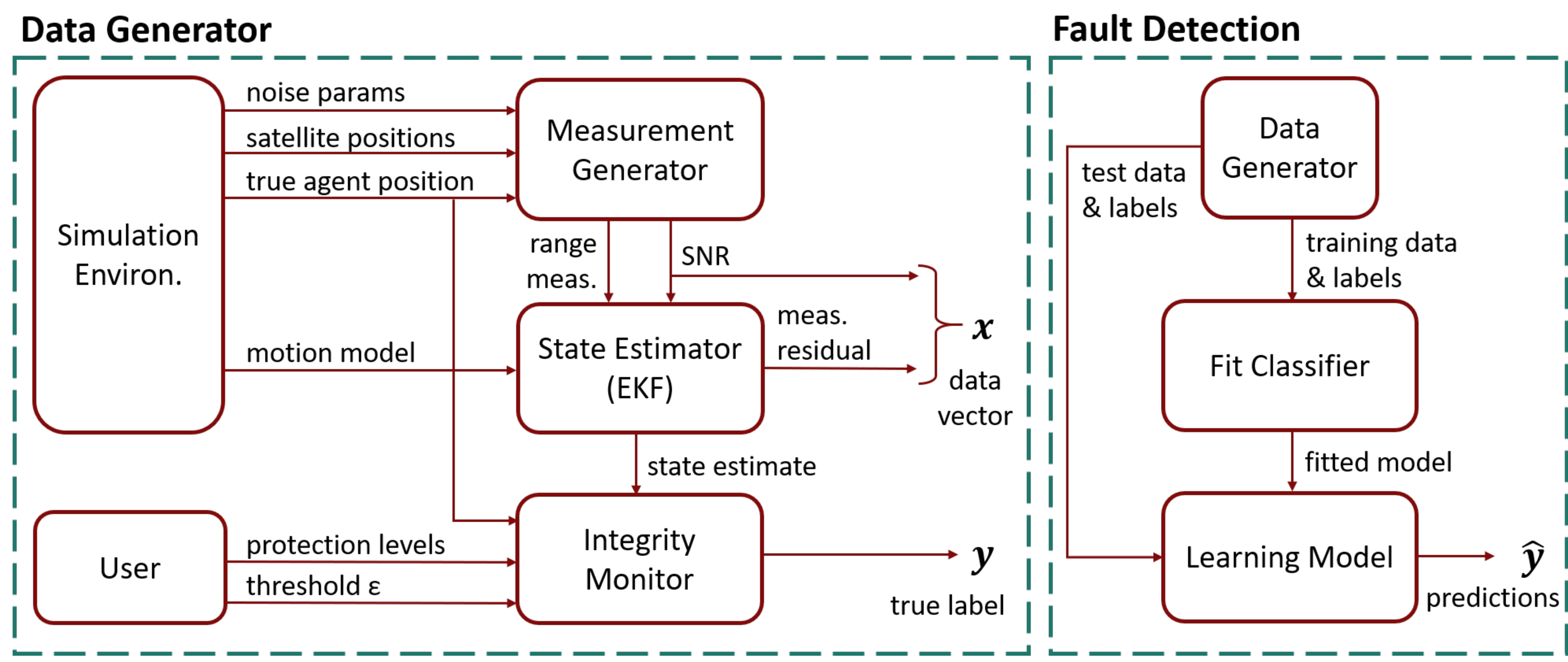
- Range residual (m): $\tilde{r}_k = \rho_k - ||\hat{z} - z_k||$
- Received SNR (dBHz) γ_k

$$x = [\tilde{r}_1, \dots, \tilde{r}_{N_s}, \gamma_1, \dots, \gamma_{N_s}]^\top$$

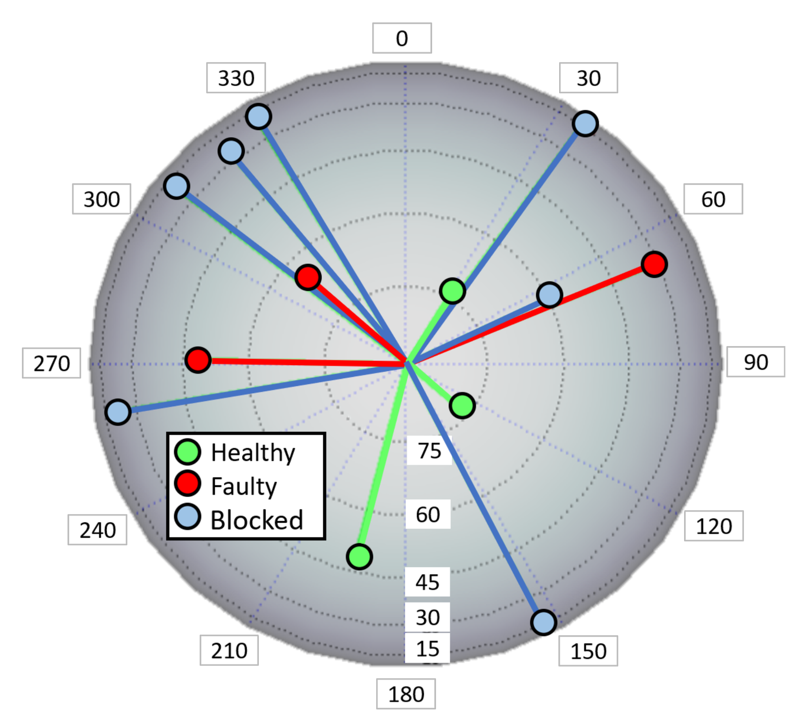
Selected References

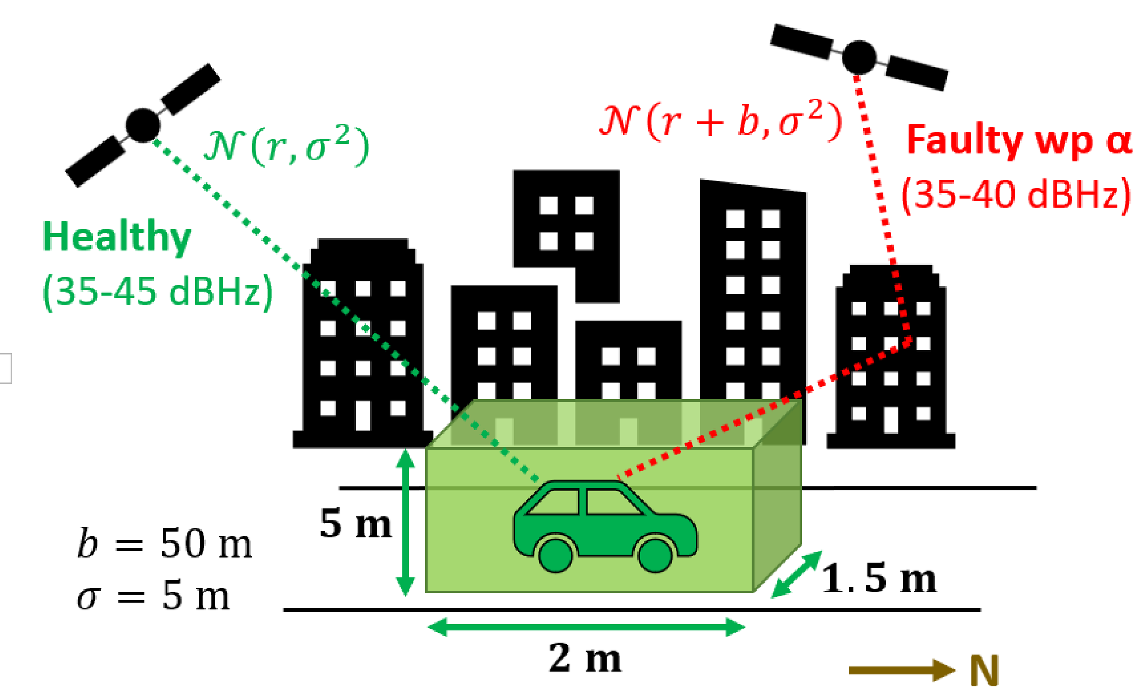
[1] Walter, T. and Enge, P., "Weighted RAIM for Precision Approach," ITM 1995.

High-Level Architecture



Simulation Environment





We model satellites in red as faulty with probability $0.2, 0.1, 0.5$, respectively. We generate 10^5 training samples, as well as 10^4 validation and testing samples.

Models Tested

- **Support Vector Machine**: SVMs solve the following primal problem

$$\min_{w,b,\zeta} \left(\frac{1}{2} w^T w + C\left(\sum_{i=1}^{n} \zeta_i\right) \right) \quad \text{subj. to } y_i(w^T \phi(x_i) + b) \geq 1 - \zeta_i, \zeta_i \geq 0, \forall i$$

- **Neural Network**: Using ReLU activation for 1-2 hidden layers and sigmoid activation for the output layer, we experiment with hidden layer widths: $h_1, h_2 \in \{2, 4, 6, 8, 12, 16, 32, 64, 128\}$. We use the following loss:

$$\mathcal{L}(\hat{y}, y) = -\frac{1}{B} \sum_{i=1}^{B} (y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)) + \lambda \left(\sum_{j=1}^{n_l} ||W^{[j]}||^2 \right)$$

Experimental Results

The table below includes performance results (accuracy, recall, precision) of the SVMs as well as a few key neural networks from the 90 architectures tested:

	Train (10^5 pts)			Test (10^4 pts)		
Learning Model	**Acc**	**Rec**	**Prec**	**Acc**	**Rec**	**Prec**
SVM (`linear`)	0.7429	0.7577	0.7165	0.7414	0.7590	0.7160
SVM (`RBF`)	0.9994	0.9997	0.9990	0.8196	0.6707	0.9321
NN ($h_1 = 2$, 1 layer)	0.5597	0.9782	0.5205	0.5567	0.9759	0.5173
NN ($h_1 = 6$, 1 layer)	0.9231	0.9245	0.9143	0.9202	**0.9245**	**0.9097**
NN ($h_1 = 128$, 1 layer)	0.9235	0.9264	0.9135	0.9208	0.9256	0.9100
NN ($h_1, h_2 = 6, 128$)	0.9232	0.9230	0.9158	0.9202	**0.9218**	**0.9119**
NN ($h_1, h_2 = 128$)	0.9235	0.9256	0.9143	0.9204	0.9241	0.9104

Comparison with RAIM[1]

We further compare our model with a standard fault detection algorithm called RAIM[1]. Since RAIM cannot handle multiple faults, we examine performance against 10^4 samples which have at most 1 fault.

Classifier	Acc	Rec	Prec
NN ($h_1 = 6$, 1 layer)	0.9478	0.9629	0.9307
NN ($h_1, h_2 = 6, 128$)	0.9490	0.9627	0.9331
Residual RAIM[1]	0.9395	0.9431	0.9316

Conclusion

We observe that our model performs better than RAIM, with regards to recall, indicating fewer missed detections of integrity faults, while maintaining higher precision, indicating fewer false alarms. Our model further handles the presence of *multiple faults*. We observe the neural networks performed better than the linear SVM at multi-fault detection, likely due to the highly nonlinear decision boundary. With the $L2$ regularization, the neural networks perform better than the RBF SVM at learning which combinations of measurements induce a fault given the satellite geometry, without overfitting to the training data.

Future Work

We plan to extend our feature vector to incorporate a sequence of measurements. Since GNSS measurements are temporally and spatially correlated, we can leverage this inherent property to improve detection of faults.