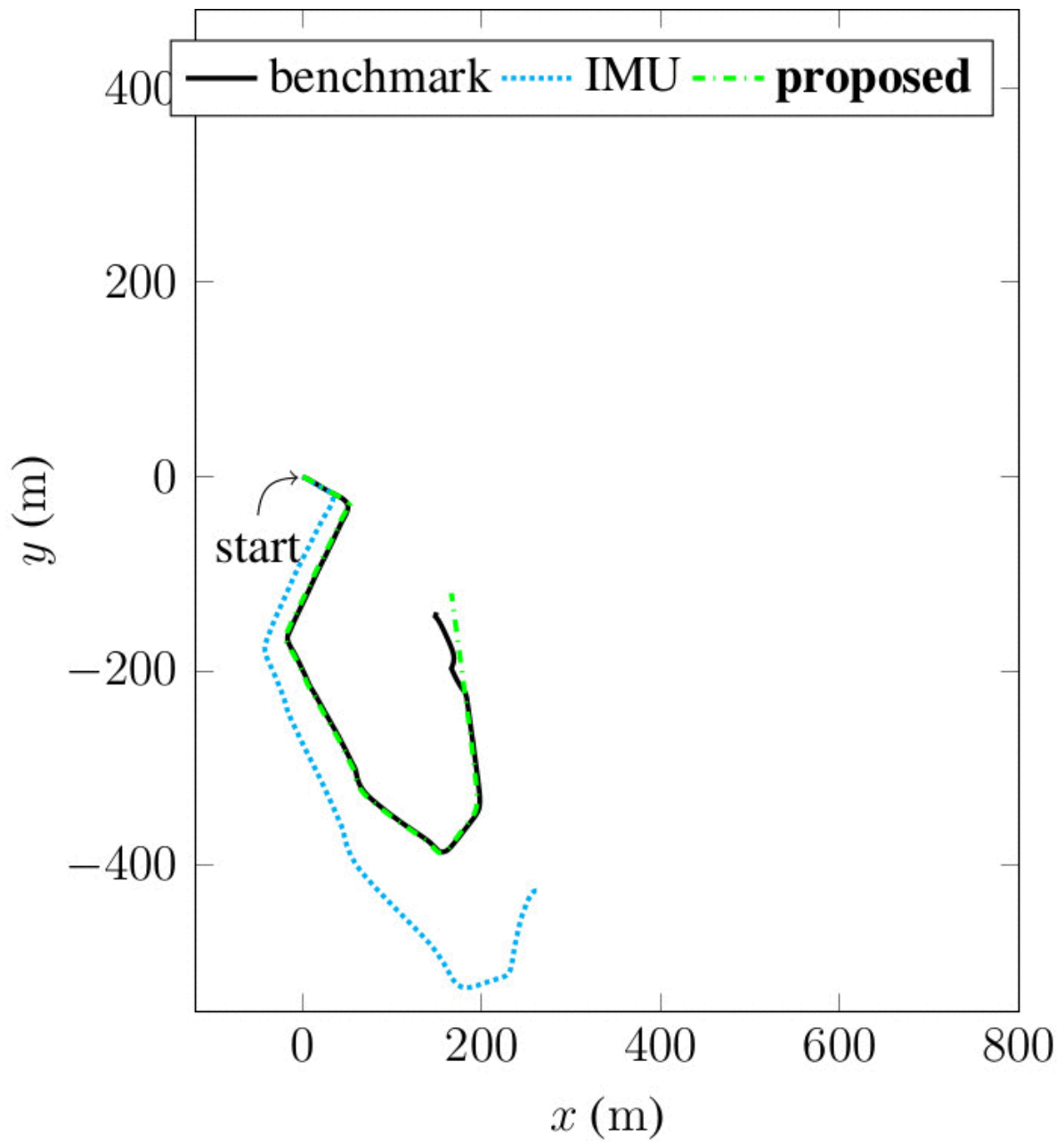


---

## AI-IMU Dead-Reckoning [IEEE paper, ArXiv paper]

1.10% translational error on the KITTI odometry sequences with **only** an Inertial Measurement Unit.

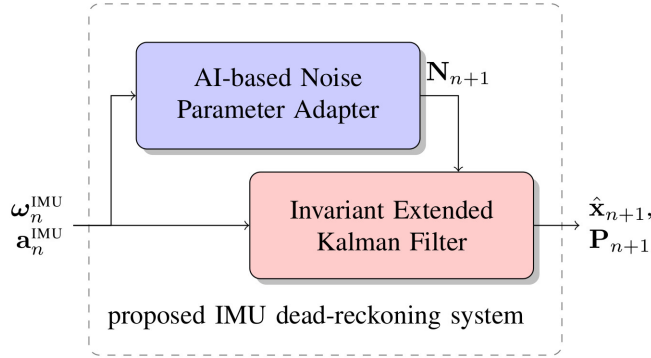


---

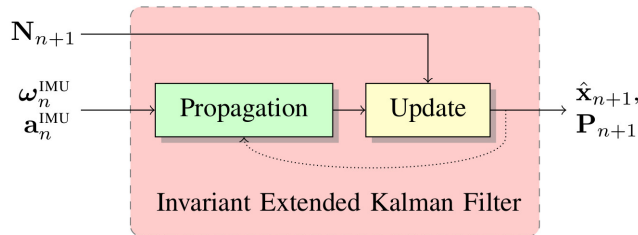
## Overview

In the context of intelligent vehicles, robust and accurate dead reckoning based on the Inertial Measurement Unit (IMU) may prove useful to correlate feeds from imaging sensors, to safely navigate through obstructions, or for safe emergency stop in the extreme case of other sensors failure.

This repo contains the code of our novel accurate method for dead reckoning of wheeled vehicles based only on an IMU. The key components of the method are the Kalman filter and the use of deep neural networks to dynamically adapt the noise parameters of the filter. Our dead reckoning inertial method based only on the IMU accurately estimates 3D position, velocity, orientation of the vehicle and self-calibrates the IMU biases. We achieve on the KITTI odometry dataset on average a 1.10% translational error and the algorithm competes with top-ranked methods which, by contrast, use LiDAR or stereo vision.



The above figure illustrates the approach which consists of two main blocks summarized as follows: 1. the filter integrates the inertial measurements with exploits zero lateral and vertical velocity as measurements with covariance matrix to refine its estimates, see the figure below; 2. the noise parameter adapter determines in real-time the most suitable covariance noise matrix. This deep learning based adapter converts directly raw IMU signals into covariance matrices without requiring knowledge of any state estimate nor any other quantity.



---

## Code

Our implementation is done in Python. We use Pytorch for the adapter block of the system. The code was tested under Python 3.5.

## Installation & Prerequies

1. Install pytorch. We perform all training and testing on its development branch.
2. Install the following required packages, `matplotlib`, `numpy`, `termcolor`, `scipy`, `navpy`, e.g. with the `pip3` command

```
1 pip3 install matplotlib numpy termcolor scipy navpy
```

4. Clone this repo

```
1 git clone https://github.com/mbrossar/ai-imu-dr.git
```

## Testing

1. Download reformatated pickle format of the 00-11 KITTI IMU raw data at this url, extract and copy then in the `data` folder.

```
1 wget "https://www.dropbox.com/s/ey41xsvfqca30vv/data.zip"
2 mkdir ai-imu-dr/results
3 unzip data.zip -d ai-imu-dr
4 rm data.zip
```

These file can alternatively be generated after download the KITTI raw data and setting `read_data = 1` in the `main.py` file.

2. Download training parameters at this url, extract and copy in the `temp` folder.

```
1 wget "https://www.dropbox.com/s/77kq4s7ziyvsrcmi/temp.zip"
2 unzip temp.zip -d ai-imu-dr/temp
3 rm temp.zip
```

4. Test the filters !

```
1 cd ai-imu-dr/src
2 python3 main_kitti.py
```

This first launches the filters for the all sequences. Then, results are plotted. Note that the parameters are trained on sequences 00, 01, 04-11, so only sequence 02 is a test sequence.

---

## Training

You can train for testing another sequence (we do not find difference in the results) or for our own sequence by modifying the dataset class.

## Paper

The paper M. Brossard, A. Barrau and S. Bonnabel, “AI-IMU Dead-Reckoning,” in *IEEE Transactions on Intelligent Vehicles*, 2020, relative to this repo is available at this url.

## Citation

If you use this code in your research, please cite:

```
1 @article{brossard2019aiimu,  
2   author = {Martin Brossard and Axel Barrau and Silv\`ere Bonnabel},  
3   journal={IEEE Transactions on Intelligent Vehicles},  
4   title = {{AI-IMU Dead-Reckoning}},  
5   year = {2020}  
6 }
```

## Authors

Martin Brossard, *Axel Barrau<sup>°</sup>* and *Silvère Bonnabel*

\*MINES ParisTech, PSL Research University, Centre for Robotics, 60 Boulevard Saint-Michel, 75006 Paris, France

<sup>°</sup>Safran Tech, Groupe Safran, Rue des Jeunes Bois-Châteaufort, 78772, Magny Les Hameaux Cedex, France