Mimicking a Motion Processing Library
Utilizing Deep Neural Networks to emulate any Black Box system

Rohan Chakraborty, Prashanth Krishnamurthy, Farshad Khorrami

***Abstract*— Modern inertial measurement sensors based on MEMS technology require on board processors and dedicated processing libraries implementing large and sophisticated sensor fusion techniques to yield rigid-body rotations and states. An artificial deep neural network (DNN) can potentially learn such a sensor fusion algorithm given the inputs and outputs of the processing library. This project aims to mimic a proprietary black-box motion processing algorithm by leveraging the self-learning capabilities of DNNs. The MPU-9150 is an inertial measurement unit (IMU) that is used along with the Invensense Motion Processing Library (MPL) to generate 3 axis angle and angular velocity measurements. DNN architectures are trained on the raw IMU data extracted from another IMU sensor (the LSM9DS0) and mapped to the MPU yaw, pitch and roll outputs. For this purpose, a parallelized Fully Connected DNN architecture is utilized to map a time sequence of raw IMU values to the yaw, pitch, and roll angles that are most likely to occur at the end of the time window. This allows the DNN to learn the unknown internal sensor fusion algorithm that has been implemented on the MPUs motion processing library and produce outputs with the LSMs raw data that have an accuracy similar to that of the MPL.**

I. INTRODUCTION

In the field of robotics, localization and mapping via sensors require an agent to measure and periodically validate its location in space in relation to its surrounding environment. The movement of a rigid-body in three-dimensional space requires the measurement of its six degrees of freedom (6DoF). Such a rigid-body is free to rotate and translate in along the three dimensional axes. An Inertial Measurement Unit (IMU) is an electronic device that measures the acceleration, angular velocities, and even the magnetic field surrounding the body. However, no sensor is perfect and robotic systems often employ multiple IMU sensors and combine their measurements to obtain the best possible information on their spatial position and orientation [1]. Such an approach of combining sensory data from multiple sources to produce resulting information with less uncertainty is called sensor fusion.

In this paper we propose that a DNN can be utilized to learn any sensor fusion algorithm without access to its underlying mechanisms, and can therefore be used as a substitute to achieve platform independence. Furthermore, once trained, a DNN can then be used along with other IMUs that yield similar raw sensor data to perform the learned sensor fusion up to a reasonable degree of variance.

II. METHOD

A. Data Extraction

We use the MPU-9150 Invensense MEMS IMU which has an on-board three axis accelerometer, gyrometer, and magnetometer to obtain the initial raw sensor data [2].

Raw readings from the MPU-9150 are acquired over a period of 126 seconds and serve as an initial training dataset for the DNN. This dataset is visualized in Figure 1. The LSM9DS1 is another IMU with similar internal accelerometer, gyrometer, and magnetometer sensors to that of the MPU-9150. The LSM9DS1 was used alongside the MPU-9150 to obtain simultaneous readings from each of their internal sensors over the 126 seconds.

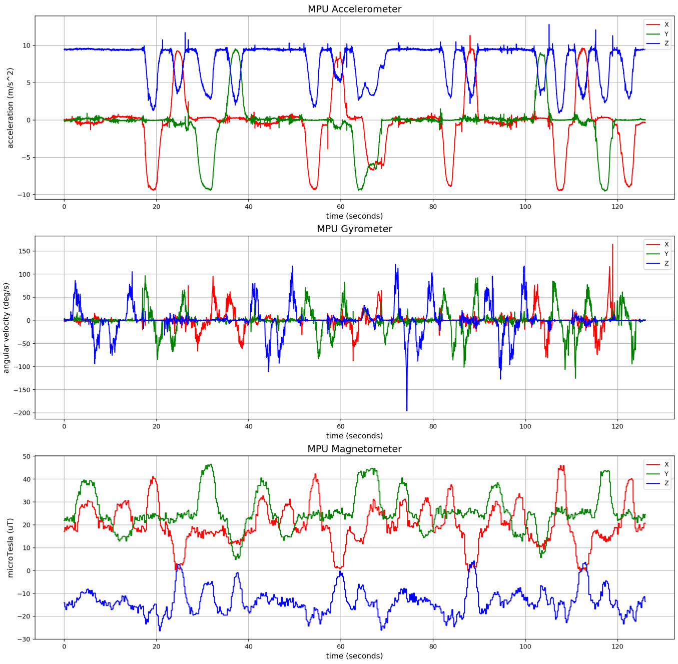

Fig. 1. Raw MPU-9150 accelerometer, gyrometer, and magnetometer readings.

Both the IMUs were read by the TM4C123G ARM cortex M4F 32-bit microcontroller over the I2C protocol. The entire setup is integrated on a custom development board called the BAP (Baby Auto-Pilot) board, depicted in Figure 2. Data was read at approximately 17.17 Hz yielding a total of 2164 unique raw IMU data points for each of the two IMUs.

The MPU Motion Driver 6.0, the proprietary Invensense Motion Processing was downloaded as set of binary files to perform the sensor fusion necessary to obtain yaw, pitch, and roll Euler angles from the MPU-9150 raw IMU data [3]. Code Composer Studio 8.0.0 was the IDE of choice for the IMU data extraction process.

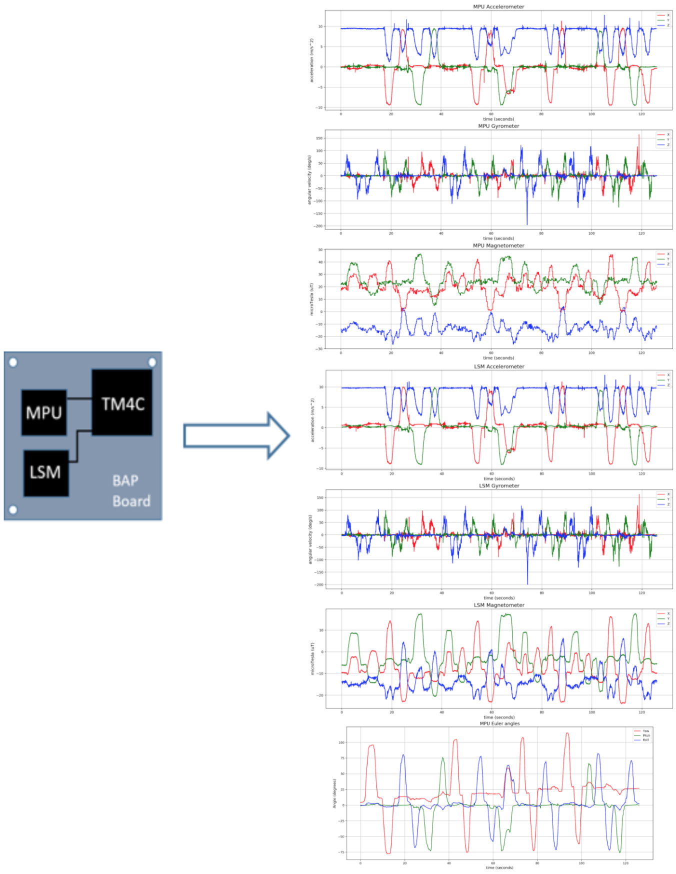

Fig. 2. Data extraction from the MPU and LSM IMUs on the BAP board.

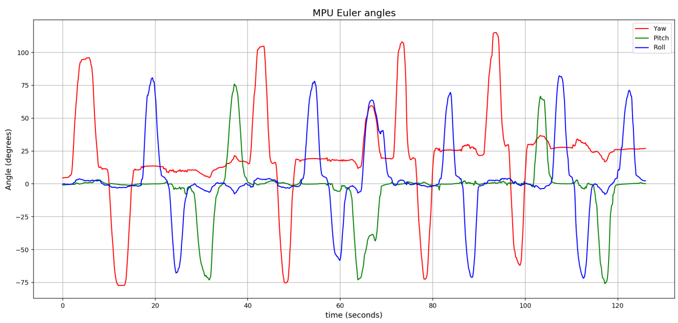

Fig. 3. MPU-9150 yaw, pitch, and roll euler angles (in degrees) output by the Invensense MPL sensor fusion algorithm.

B. *Deep Learning Architecture*

The raw 3-axis accelerometer, gyrometer, and magnetometer data constitute the 1x9 input vector that can be fed through a network architecture of choice. To map the raw values to the yaw, pitch, and roll angles produced by the Invensense MPL we use a parallelized fully connected deep neural architecture. The yaw, pitch, and roll outputs of the network are scaled by performing a modulus of outputs by 360 and subtracting 180. This rescales the network outputs to a range within [-180, 180]. The network was modeled with the popular pytorch library in Python.

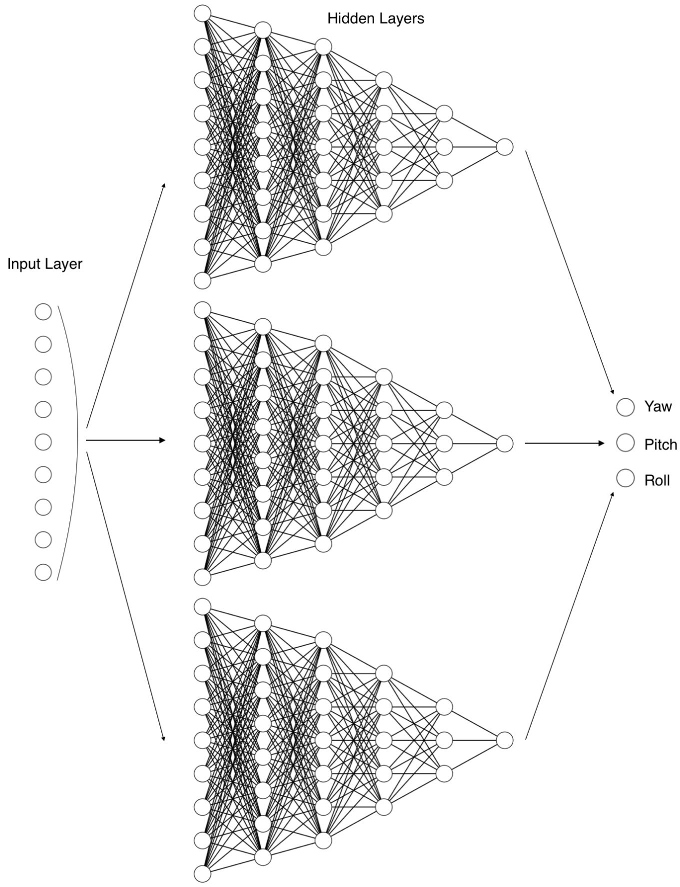

Fig. 4. The nine raw IMU values (comprising of 3 accelerometer, 3 gyrometer, and 3 magnetometer readings) are fed through a parallelized FCN architecture consisting of 3 parallel hidden layers, each mapping the input vector to the corresponding yaw, pitch, and roll angles.

C. *Network Training and Testing*

A cosine-distance loss function is used during network training. It is defined as -

$$loss = (1 - cos(\hat{y} - y))^2$$

The network uses an Adam Optimizer with a learning rate of 0.0001. The network was trained over 500 epochs on the exact dataset seen in Figure 1 with a 70/30 train-test split. A graph of the cosine loss over the first 250 epochs is shown in Figure 5.

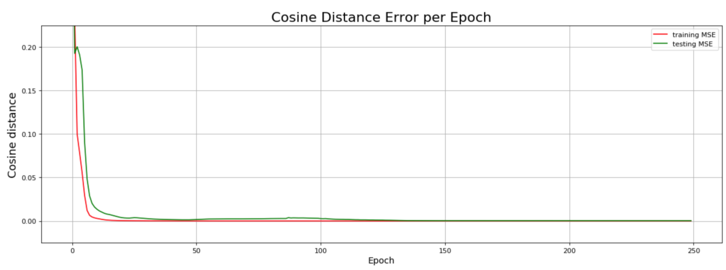

Fig. 5. Cosine loss during network training over the first 250 epochs.

The training and testing variance explained asymptotically reach 99.2% and 95.3% respectively. Once trained with the MPU data, the trained network weights can then be trained further to map the LSM raw data to the MPU MPL Euler angle outputs. The LSM raw accelerometer and gyrometer data was manually filtered with appropriate low pass filters, and the LSM magnetometer data was passed through standard Hard-Iron and Soft-Iron calibration. Lastly, each of the LSM's accelerometer, gyrometer, and magnetometer were transformed to be precisely oriented to that of the MPU accelerometer, gyrometer, and magnetometer.

III. EXPERIMENTAL RESULTS

A comparison of the direct computation of the Euler angles from Figure 3 against the network predicted angles is shown in Figure 6.

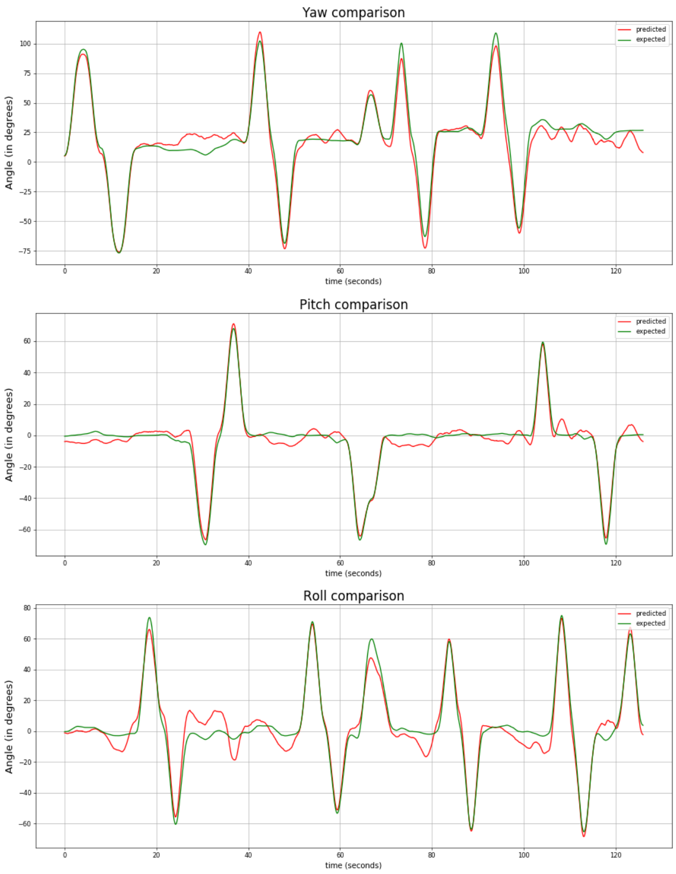

Fig. 6. Comparison of the expected Invensense MPL Yaw, Pitch, and Roll angles with the network predicted outputs.

The training/testing criteria after 500 epochs are-

Max training R^2 value	0.9928
Max training Variance explained	0.9928
Max test R^2 value	0.9301
Max test Variance explained	0.9538

The mean difference obtained for each of the Euler angles-

Criterion	Yaw	Pitch	Roll
Mean Difference (between expected and predicted in degrees)	5.68	3.44	5.32

IV. CONCLUSION

It may be concluded from the comparison graph in Figure 6 that Deep Neural Networks can be used to accurately approximate any black-box system to reproduce the system output without any access to the underlying mechanism. The proposed fully connected network can be used to perform a sensor fusion of any given IMU to produce results comparable to that of the Invensense Motion Processing Library's sensor fusion algorithm.

ACKNOWLEDGMENTS

- NYU Tandon School of Engineering, Undergraduate Summer Research Program
- Control/ Robotics Research Laboratory (CRRL), Professor Farshad Khorrami, and Dr. Prashanth Krishnamurthy

REFERENCES

[1] Krzysztof, K., Aleksandra, S., Mateusz, M., Karol, G., Igor, K. and Rybarczyk, A. (2013). Nine-Axis IMU sensor fusion using the AHRS algorithm and neural networks. [ebook] Pozna, Poland. Available at: https://dspace5.zcu.cz/bitstream/11025/11492/1/Krzysztof.pdf [Accessed 13 Aug. 2018].

[2] MPU-9150 Product Specification Revision 4.3. (2013). [PDF] Invensense, pp.7,9. Available at: https://www.invensense.com/wp-content/uploads/2015/02/MPU-9150-Datasheet.pdf [Accessed 13 Aug. 2018].

[3] Motion Driver 6.1 User Guide, Revision: 1.0. (2014). [ebook] Sunnyvale, CA: Invensense, pp.4,5, 9-14. Available at: http://www.digikey.tw/Web