Unlocking High-Speed Data Tracking: Leveraging Affordable IMUs as a Replacement for Costly Systems

Hudson Aibel[1,2], Marc Mitjans i Coma[2], Roberto Tron[2]

Crossroads School, 1714 21st Street, Santa Monica, CA[1]; Boston University, 750 Commonwealth Ave, Brookline, MA[2]

Introduction

- The main objective of this project is to reduce the price of the currently used Inertial Measurement Unit (IMU) system that tracks the rotation of users' legs.
- A paper by Brossard et al shows that gyroscopic data can be filtered using a convolutional neural network (CNN). A CNN involves slides a small filter over input data to perform element-wise multiplication and summation, enabling the network to capture spatial patterns and structures.

❖ Can we find an alternative that is lower priced but matches the data accuracy?

❖ Using a hobby grade IMU in combination with a machine learning filter, can the same performance be reached at a fraction of the cost?

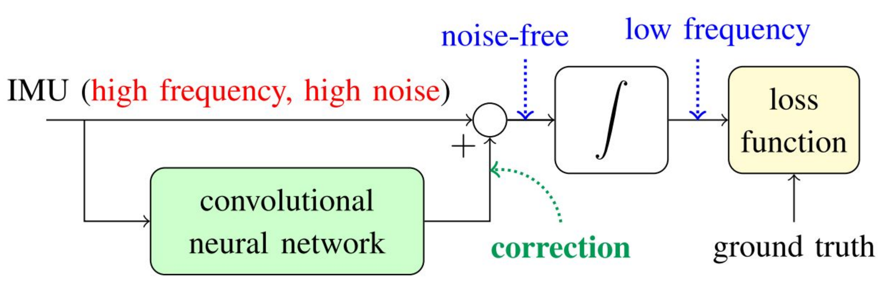


Figure 1. The process for denoising IMU data, from Brossard et al.

Methods

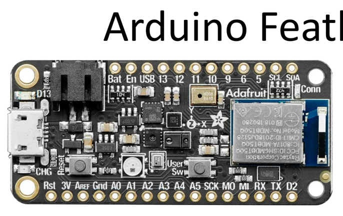
- Three different methods were tested to see which is the best performing. The criteria consisted of lowest latency, highest refresh rate, and all-around ease of use.

Wearable device

3 different setups were tested:

The 2nd and 3rd setups were connected to a BNO08X IMU

Arduino Feather with built-in IMU



This setup simply uses a serial connection to communicate with the ground station and has an internal IMU to measure rotation.

Figure 2. Adafruit Feather nrf52840 Sense.

Arduino Feather with LORA Radio protocol

Using a radio protocol, this system transmits its data wirelessly. The IMU is connected via I2C bus with a built-in plug.

Figure 3. Adafruit Feather RP2040 with rfm95.

Arduino Feather with Raspberry Pi with WIFI

This system uses the same Feather and IMU as the previous one, however now the Feather is connected to a Raspberry Pi that transmits the data over WiFi using Robot Operating System (ROS). **This was the system used in testing, except that the raspberry pi was swapped with a full laptop.**

Figure 4. Raspberry Pi 3b

Ground Station

Each of the different types of systems connected to a laptop that recorded the data. For the radio system, an Arduino at the ground station first received the data, then sent to the computer over serial.

Learning Model

A CNN model was used for learning. It consisted of one convolutional layer - with an input channel size of 3, a temporal size of 30, and an output channel of 32. It is then followed by a pooling layer, a flattening layer, and finally a linear layer with an output size of 3. The model attempts to predict the next point using the prior 30 points of data.

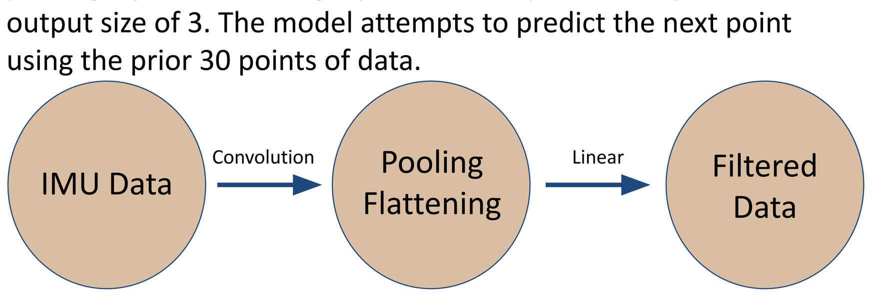


Figure 5. The IMU data goes through a convolution, a pooling and flattening layer, and then a linear layer.

Results

To verify that the machine learning model is viable, artificial data (a sine wave with noise) was first tested against a linear model.

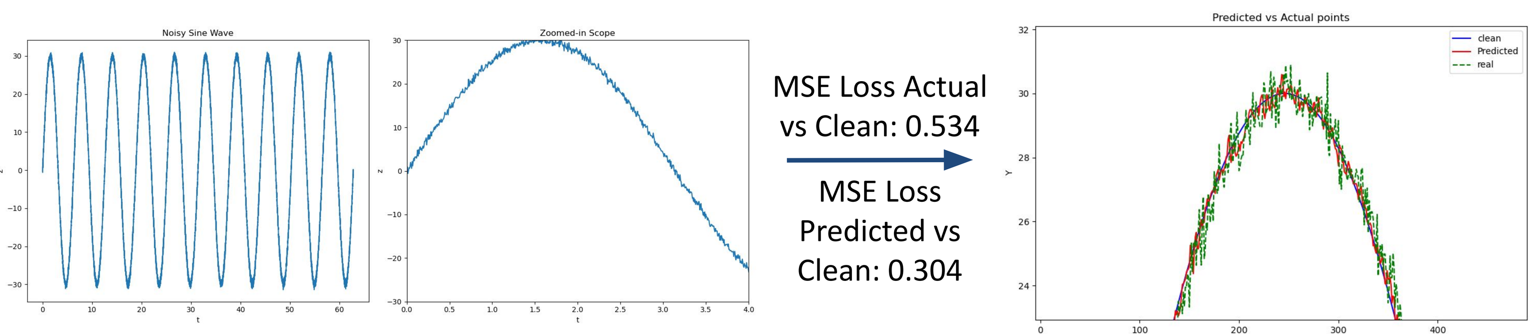


Figure 5. (a) The noisy sine wave shown. (b) Zoomed in photo of the sine wave, with the clean data, noisy data, and the data predicted by the linear model. It is a model where every input has a linear relationship with one another.

The Sine waves were then tested against a CNN

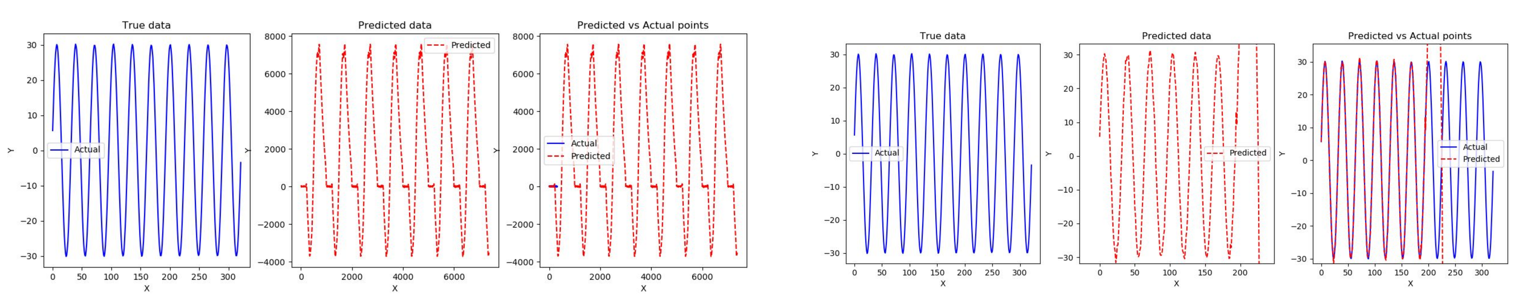


Figure 6. (a) Sine wave Y, the prediction for Sine wave Y, and the data overlapped. (b) The same graph, but with the predicted data zoomed in to the same size as the original sine wave. It is unclear why the model continues to extrapolate past the given data.

The acceleration from the built in filtering vs the unfiltered acceleration

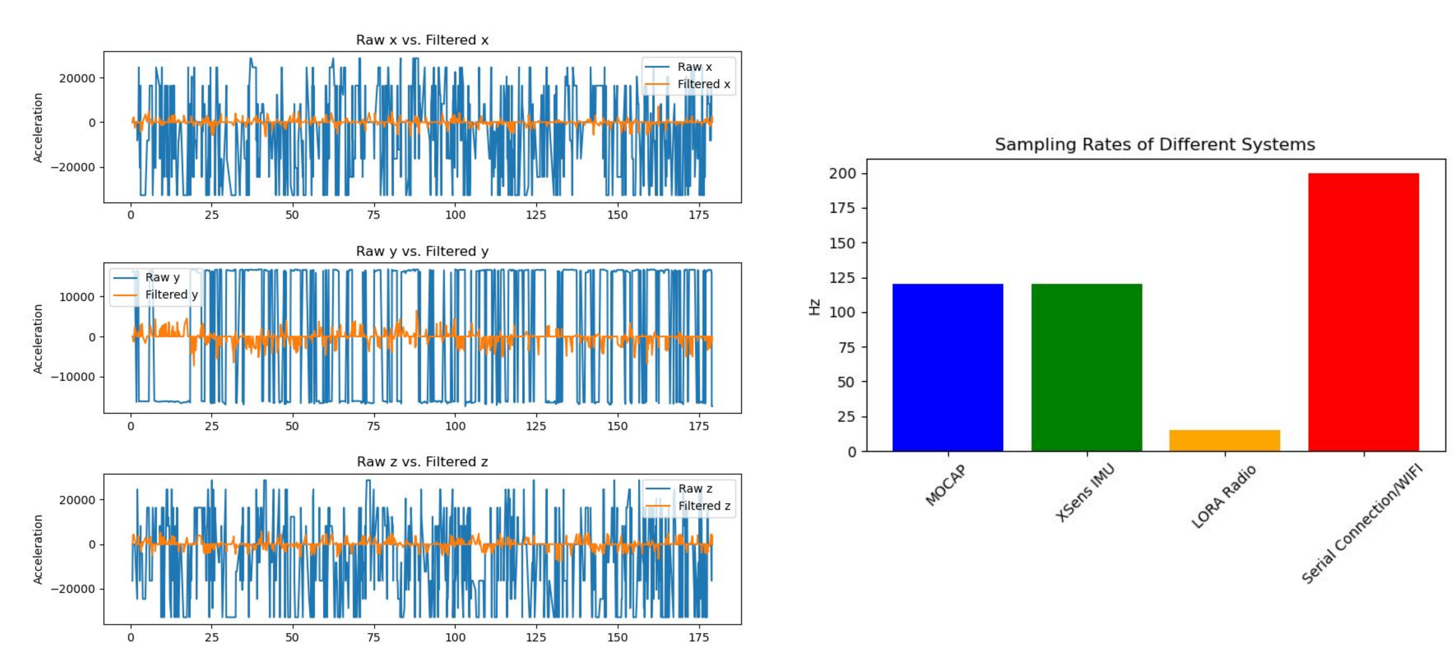


Figure 7. (a) Comparing the amount of noise of the filtered acceleration (orange) versus the unfiltered acceleration (blue) (b) Comparing how fast each system can transmit data.

The raw IMU data was put through the CNN model and compared against the filtered data

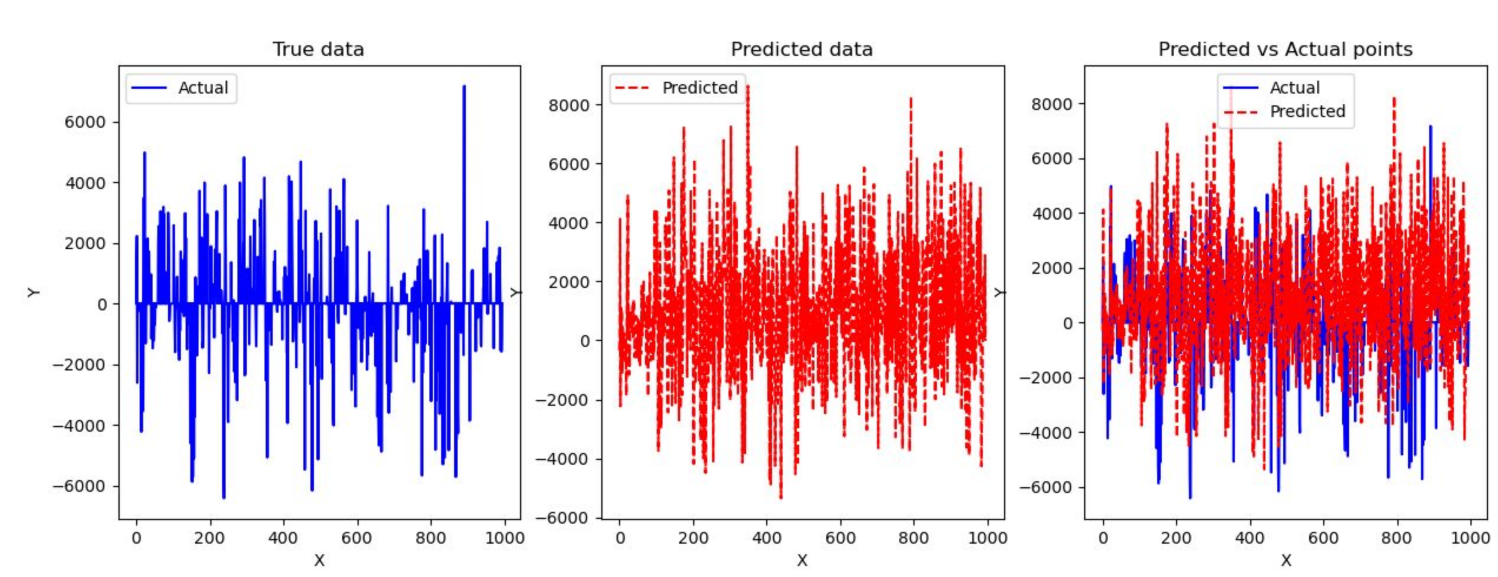


Figure 8. 1000 windows of IMU data were put into the CNN as outlined previously, the predicted data (red) was then overlapped over the clean data (blue).

The BNO08X tested against the MOCAP system

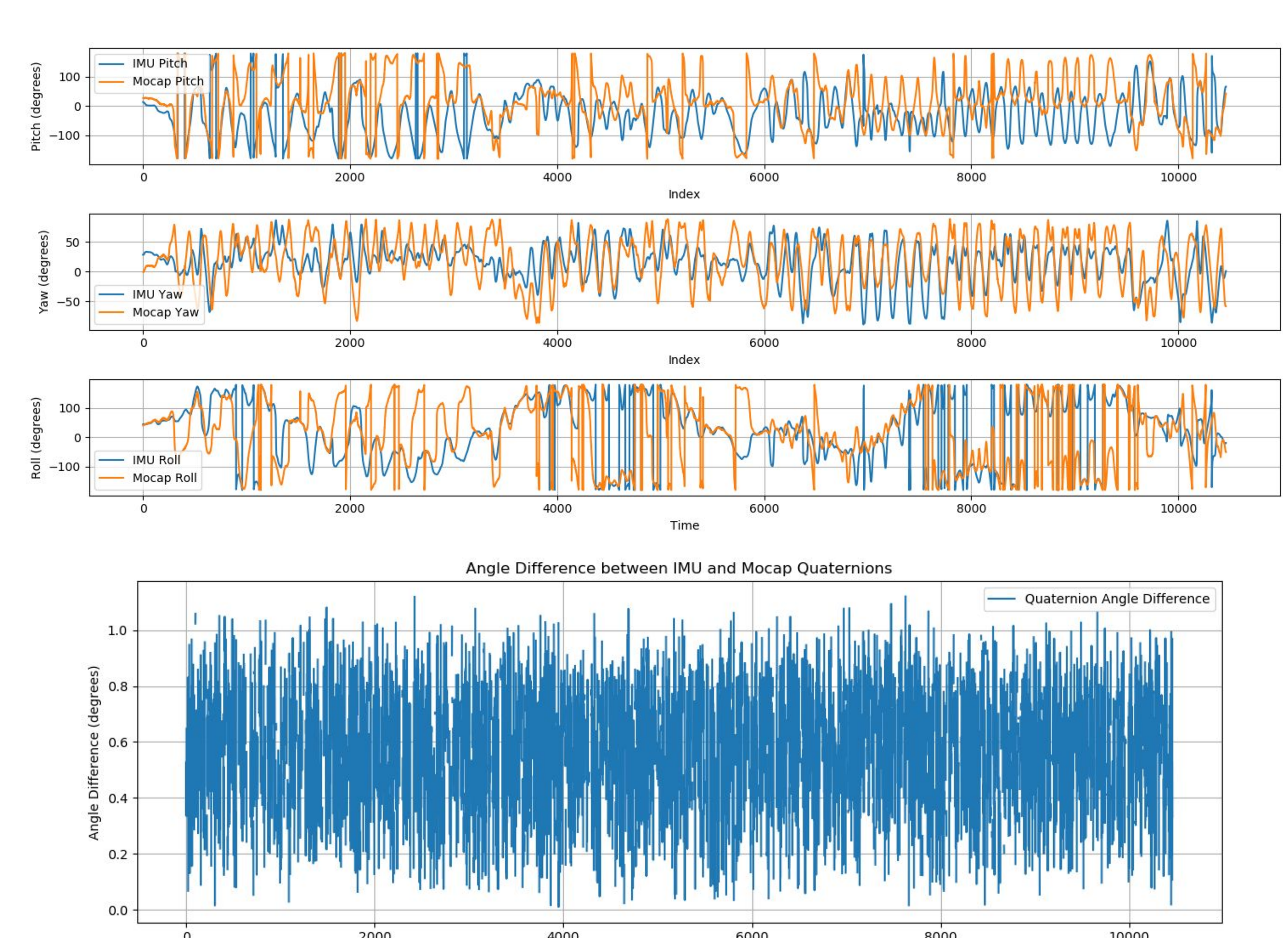


Figure 9. (a) The filtered IMU data versus the MOCAP data, graphed in euler. (b) The difference in angle (degrees) between the IMU and MOCAP system.

Conclusions

- Based upon early testing, it was clear that a machine learning model can reduce noise in a signal
- However, the CNN (Convolution Neural Network) struggled more than the simpler model, as evidenced by the skewed scaling
- When the raw IMU data was passed through the CNN, the loss function went down to near 0 after 100,000 epochs
- Even with a near 0 loss function, the predicted data did not align with the filtered data
- This points to the data being overfitted

❖ While the machine learning model did not outperform the built-in filtering, the fact that it over fitted means that the model is learning. By changing some parameters it is possible that the performance could improve.

- The IMU has built in filtering, which uses sensor fusion to combine the gyroscope, the accelerometer, and the magnamator to accurately denoise the data.
- When comparing the built-in filtering to the ground truth (MOCAP), the data matches very closely less then a 1.1 degree different at all times.
- The axis flips that appear are due to the cameras on MOCAP losing sight of the tracking nodes.
- The IMU appears to have more consistent and smooth data than the MOCAP system on the Z and W axis.
- The refresh rate was highest over serial and WiFi (200hz). This outperformed both the Xsens IMU (120hz) and the MOCAP system (120hz).

❖ With just built-in filtering, the IMU has an extremely high refresh rate and appears to be accurate, even without the machine learning filter. More testing needs to be done, like running the hobby grade IMU directly against the Xsens IMU, but the hobby grade IMU could prove to be a reliable replacement to the current system.

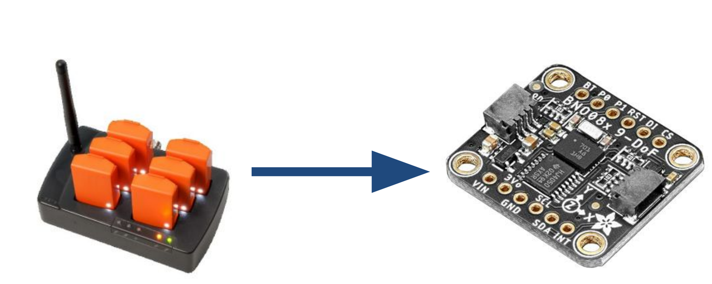


Figure 10. The Xsens IMU and the BNO08X are shown

References

(1)MTw Awinda | Movella.com. www.movella.com. https://www.movella.com/products/wearables/xsens-mtw-awind (accessed 2023-08-03).

(2)Brossard, M.; Bonnabel, S.; Barrau, A. Denoising IMU Gyroscopes with Deep Learning for Open-Loop Attitude Estimation. IEEE Robotics and Automation Letters 2020, 1–1. https://doi.org/10.1109/lra.2020.3003256.

(3) Aibel, H.; Mitjans, M.; IMU Testing. www.bitbucket.org. https://bitbucket.org/burobotics/imu_learn/src/master/

(4) Aibel, H. BU Blog. Hudsonaibel.com, 2023.

Figure 1. 1. Martin Brossard , Silvère Bonnabel , and Axel Barrau; Denoising IMU Gyroscopes With Deep Learning for Open-Loop Attitude Estimation; 2020, 5, 4796.

Acknowledgements

I'd like to thank to thank the entire BU Robotics lab for providing the resources to pursue my research, and I would like to specifically thank Professor Roberto Tron, Marc Mitjans i Coma, and Renan Jewell for guiding me through my project.