A Smart Shoe for Runners

Kyle Berkson

Design with Microprocessors, Cambridge Future Scholar Program

Dr. George Anwar, Mechanical Engineering Professor, UC Berkeley

1 - Abstract

Elite runners are constantly striving to enhance their running economy and prevent injury. Traditional kinematic analysis, conducted in laboratory settings, can offer valuable insights into doing just that, but such analysis is often time-consuming and necessitates running within a controlled environment. Recent innovations have placed "Smart Shoes" at the forefront of scientific and athletic discussion; however, current smart shoes lack abilities like hands-free interfaces real-time analysis of data, and during-run user feedback—as most shoes on the market allow the user to view various statistics linked to their run afterwards. Alternative data-collection and feedback devices, like smart watches, do provide runners with real-time data and feedback, but their information value is constrained, and user feedback can divert a runner's attention.

This is where this smart shoe prototype becomes significant. Utilizing an Inertial Measurement Unit (IMU), the shoe monitors the runner's foot movement and continuously analyzes foot-strike patterns. If it finds the form to be erroneous, it sends a signal to a vibration motor, which signals the runner to adjust their form. This real-time feedback empowers runners to make immediate adjustments to their form, reducing excessive force and stress associated with both heel striking and forefoot striking. Notably, heel striking is often linked to impact in a runner's gait in front of the body, which can potentially reduce running efficiency, especially at higher speeds. Forefoot striking is also associated with heightened stress and tension in the knee area. In contrast, midfoot striking—as I was taught by a cross country and track coach—is commonly employed by elite runners and supposedly has optimal energy return and does not create stress specific to a certain limb—which can cause injury.

It is important to note though that there is limited scientific research confirming midfoot striking is the most efficient, "biomechanically sound," and least injury-prone foot strike. All runners are different and optimal foot strike behavior can be distance and type (training vs. race) specific. The goal of this shoe is not to say that there is only one acceptable running form. The underlying importance of this smart shoe is its ability to analyze data mid-run and help the runner make decisions based upon those factors.

2 - Introduction

The goal of this project was to create a prototype of a shoe which could reflect the quantitative power of an IMU (or other sensing hardware) in a gait analysis application which broke free from lab confines and whose application would be to aid runners in fine tuning their form. For example, think of a runner wearing a smart watch. A smart watch is useful as it provides details such as pace, heart rate, cadence, elevation, etc. This is useful; however, to an elite runner, it may only pose as a distraction from their run/workout. As demonstrated by the shoe's foot strike detecting ability, this system aims to be an example of a solution to that problem, which can notify the runner of potential errors in their form, giving them the ability to adjust their form mid-run seamlessly.

To achieve the goals above, the following variables associated with the movement of a runner's shoe were measured by the IMU:

- X, Y, Z axis acceleration
- X, Y, Z axis angular velocity

This data was used to estimate magnitude of deceleration caused by a foot strike, foot strike timing, angle of the foot at time of foot strike, and stride pace.

3 – Methods

The focus of this section is to discuss hardware and their implementation, how the code achieves measurements of variables in a runner's gait, and the prototype design of the smart shoe.

3.1 Hardware Background

In designing the smart shoe, various systems and sensors were used. To estimate the parameters as mentioned in section 2—i.e., magnitude of deceleration caused by a foot strike, foot strike timing, angle of the foot at time of foot strike, and distance between each stride—the system used an ESP32 microcontroller connected via a QWIIC connection to other I2C enabled sensors. A QWIIC connection simplifies wiring (allowing for an efficient, simplified shoe design) as it connects slave devices to a master via a ground, 3.3V, SDA (Serial Data), and SCL (Serial Clock) cable which connects via a 4-pin JST, allowing for simplified use of I2C interfacing.

The I2C communication protocol was ideal for this application as it allows for multiple systems to be chained to a master which in turn enabled multiple sensors to be utilized at once with seamless control from the ESP32. In essence, I2C allows a master to initiate reading/writing of data—utilizing addressing to transfer data to respective slave devices.

3.2 Hardware Design

The devices/materials included in the design were:

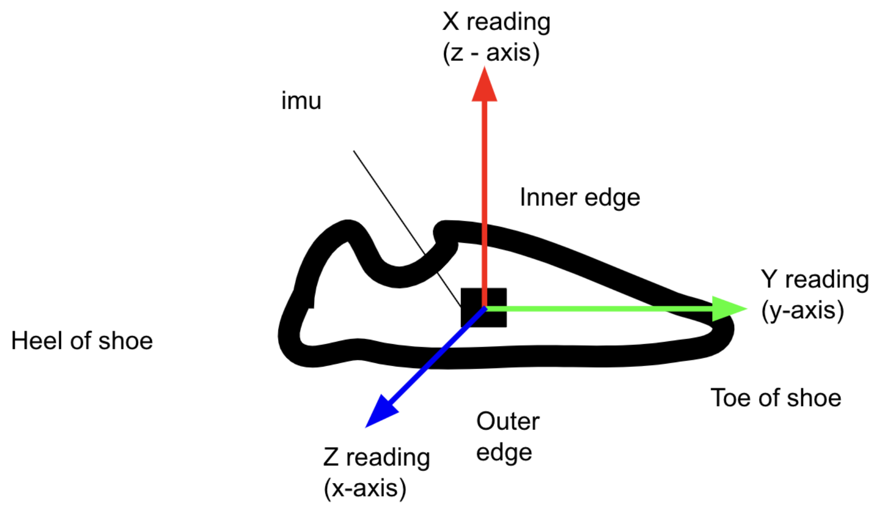
- ESP32 microcontroller
    - Dual core, 32-bit, equipped with wireless or WIFI network and Bluetooth Low energy
- QWIIC enabled 6 degrees of freedom IMU (LSM6DSO)
    - Capable of reading accelerometer and gyroscope data in 3 axes at various resolutions
- QWIIC enabled motor driver
    - Allows for various levels of drive strength

- Coreless vibration motor
    - 1.5-3v, 8000-16000RPM
- 2 AA battery holder

Attributes associated with distance, velocity, and orientation were measured by the built-in accelerometer and gyroscope in the IMU, which was placed on the outer edge of the right heel. As only one system was made—and placed on the right heel—the assumption was made that the data for the right foot would also reflect how the left foot acts. In the future, it would be worth exploring a system on both feet to avoid such assumptions. Positional velocity and displacement—if needed—were determined through single or double integration given an acceleration at a given time interval. Similarly, angular displacement was determined through single integration—but also checked for accuracy through filtering methods described later in the paper.

3.3 Design Summary

As shown in the figure, the placement and orientation of the IMU meant that its readings had to be interpreted slightly differently than normal. When placed flat on a table, the axes that the IMU reads data on would reflect a 3D orientation given by the right-hand rule. Since the IMU was placed flat on the side of the shoe, "x-axis" was considered analogous to the z-axis as given by the right-hand rule, the "z-axis" was considered analogous to the x-axis, and the y-axis remained the same. When references to the x, y, and z axes will be of the true axis, not that of the readings.



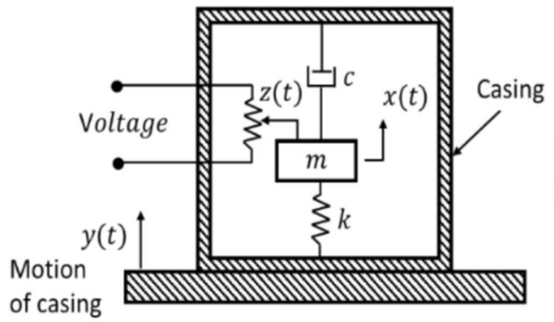
All hardware components were attached near the rear end of the shoe. The reasoning behind this placement is that it is unobstructive and less obtrusive to the runner. The total weight of the system came out to be ~95 grams which was 0.001% of the body weight. Increasing the weight of a runner's shoe has effects on running economy and even gait patterns [3]. While there is no specific scientific conclusion for this claim, the weight of this system is of little magnitude compared to body weight so it likely to have minimal unwanted effects on the gait; further, future iterations of this prototype could be reduced in size and weight.

The above figure shows the design of the actual prototype. The hardware was connected to a Nike Pegasus 38 running shoe. Cardboard was placed below the hardware components for stability; the unit was secured to the show via duct tape.

While the ESP32 has wireless capabilities, all designing and testing was done by connecting the ESP32 to a computer serially.

3.4 Hardware Use

The 6 degrees of freedom IMU contains a mass which is suspended in air and connected to a capacitor. When the mass is subjected to acceleration it moves, which results in a change in level of capacitance. At rest, the IMU reads about –9.81/s^2 along the axis which points upwards (i.e., force of gravity), highlighting its ability to measure both static and dynamic acceleration. [4]



The IMU has the capability to output acceleration on scales ranging from ±2 g to ±16 g. (g = acceleration due to gravity, -9.81m/s^2). Angular velocity is output on scales ranging from ±250 dps to ±2000 dps (degrees per second). The higher the scale, the lower the accuracy/resolution of the resulting data, so careful considerations had to be made. As peak forces can range anywhere from 1000N to 2000N in a runner's stride [1], I chose a scale of ±8g. For the body weight with which the shoe was tested (my own), this allows reading forces of ±4630N which safely exceeds this 2000N threshold, preventing any overflow errors. Angular velocity was read on a scale of ±1000dps. To determine that this scale fit, I tested running with the shoe at a pace of 7:30 min/mile at various scales and saw if overflow errors occurred (which were characterized by extraneous readings). It was determined that ±1000 dps was able to read without any such errors and maximized accuracy.

3.5 Software

The general architecture of the software relies on a loop that reads data at an interval of 0.01 seconds for at least 5 seconds, analyzes the data and stores it in a list, and outputs it for feedback. This interval of 0.01 seconds was chosen since it is a sampling speed fast enough that it accounts for most of the movement in a runner's gait [2].

Upon the initiation of the program—started either from the computer or ESP32—data x, y, and z acceleration and angular velocity are immediately read. Experimentation was required to determine how to interpret these values.

3.5.1 Determining Shoe angle

The initial approach in determining the angle of the shoe at impact given the known data involved the dot product.

$$u \cdot v = ||u||||v||cos\theta \rightarrow \theta = \cos^{-1}\left(\frac{u \cdot v}{||u||||v|}\right)$$

Where u is the current acceleration vector and v is the acceleration vector of the shoe at rest, θ is the angle between them. The angle was signed as negative if acceleration reading along the y-axis was less than 0.79 $\frac{m}{s^2}$ (this measurement changes as the shoe is rotated around the x-axis). This approach seemed to work; however, further testing revealed that there were extraneous levels of error introduced by different forces acting on the shoe during running.

This model for finding the orientation of the shoe works when the same forces that act on the shoe at rest are acting on the shoe in a different state. At rest, the main force acting is the force due to gravity. When the shoe is rotated, different amounts of force due to gravity will act in the x, y, and z directions. So, if we were to rotate the shoe, not moving it up and down, this returns the correct angle of the shoe (in radians). During running, there are applied forces acting on the shoe which scale its acceleration in the x, y, and z direction. Since θ is dependent on u and its magnitude, a shoe with an orientation of 0 degrees hitting the ground has a different calculated angle than a shoe with an orientation of 0 degrees that is stationary.

This challenge resulted in a riskier approach. The new strategy was to determine the angle of the shoe using gyroscope measurements. Specifically, the angle of the shoe would be given by integrating the angular velocity of the shoe with respect to time. Assuming the shoe started flat on the ground, the angle of the shoe was given by:

$$\theta = \theta_{prev} + \omega t$$

where ω is angular velocity and t is the interval, 0.01 s. This would correctly return the angle of the shoe if the following were true:

1. The velocity of each interval always accurately reflects the motion of the shoe
2. Electrical noise does not skew measurements

Since these assumptions cannot be made, this model needs to be improved upon. To achieve a better predicted angle, the code utilized a 1D Kalman Filter. The formula for a 1D Kalman Filter, which predicts x_t is given by:

$$x_t = x_{t-1} + \frac{P_{t-1} + Q}{P_{t-1} + Q + R}\left(z_{t-1} - x_{t-1}\right)$$

...in terms of the angle of the shoe (predicting θ_t):

$$\theta_t = \theta_{t-1} + \frac{P_{t-1} + Q}{P_{t-1} + Q + R}\left(\left(\theta_{t-1} + \omega t\right) - \theta_{t-1}\right)$$

The Kalman filter is a mathematical algorithm which is used to estimate the state of a system, based upon a series of measurements where there is uncertainty in the process used to get that measurement and where there is noise present in obtaining measurements used in the process [5]. This fits the application perfectly as there is uncertainty and error when using single integration to find the angle of the shoe, and there is noise present in obtaining the angular velocity, which is used to calculate the shoe's angle.

<u>Initial Values</u>

Variable	Description	Initialized Value	Unit
R	Measurement noise covariance	.001	deg^2
Q	Process noise covariance	5	deg^2

P_{t-1}	Error covariance of the state estimate	0.5	deg^2
θ_{t-1}	Initial angle of shoe	0	deg

Since θ_{t-1} is initialized at a value of 0, the program must be started while the shoe is flat and at rest—this was chosen for simplicity. P_{t-1} is 0.5, which represents how much we think the initial angle of the shoe deviates from the actual. Finding a value for the constant *R* involved finding the standard deviation of the angular velocity given by the shoe. To do this, I wrote and ran a program for 20 minutes that read the angular velocity about the x-axis for a shoe at rest and found how much the measurements deviate from 0 on average. This gave an average noise reading of 3.16283 $\frac{deg}{s}$. To get the average deviation of the *angle* of the shoe, I multiplied this by the time interval, 0.01 and then squared it to find covariance, resulting in an R value of 0.001. Q was determined through experimentation and seeing what value of Q minimized errors. Finally, *P* is updated as:

$$P_t = \left(1 - \left(\frac{P_{t-1} + Q}{P_{t-1} + Q + R}\right)\right) \cdot (P_{t-1} + Q)$$

P_t is then stored in a variable to be used in the next iteration. The angle of the shoe at each time interval was stored in a list for later use.

The Kalman filter does well at minimizing errors in determining the angle of the shoe, but extra security measures can help further.

From experimentation done when the dot product was initially used to (incorrectly) determine the shoe's angle, it was determined that the acceleration along the z-axis was between 9.6 and 9.8 (due to noise) when the shoe was at rest. Given this assumption, before the angle of the shoe is updated by the Kalman filter, it is checked if the acceleration along the z-axis lies within this threshold, and if so, all the variables within the filter go back to their initialization values and the shoes angle is said to be 0° for that iteration.

3.5.2 Velocity

Velocity was given by integrating the acceleration along the y-axis over the time interval, 0.01s, and adding to the previous velocity each iteration. Like calculating the angle of the shoe, calculating the velocity of the runner required the implementation of a Kalman filter, due to the presence of electrical noise and the buildup of error caused by integration. The only difference with this Kalman filter was that R was given a value of 0.4, which was calculated in the same manner as for the angle of the shoe. The velocity value at each interval was stored in a list for later use.

3.5.3 Determining when the Shoe is on the Ground

With investigation, it was determined that the overall magnitude of the acceleration would be highest upon impact. This is because as the foot hits the ground, its velocity must go from some value straight to 0 in a rapid manner. This was used to determine when the shoe was on the ground.

$$||A|| = \sqrt{x^2 + y^2 + z^2}$$

For each iteration of data that was retrieved, the magnitude of the shoe's acceleration was also recorded and stored in a list for later use.

3.5.4 Evaluation

Once a minimum of 5 seconds has passed and the user is stepping on the ground, code iterates through the list of magnitudes finding the indexes where the magnitude is at a maximum relative to those around it and is at least 45 $\frac{m}{s^2}$ (which was determined through experimentation). Those indexes are subsequently used to output the respective angles for each step. Velocity is simply output as a list.

Each angle in the output list is categorized as either a midfoot strike, forefoot strike, or heel strike.

Category	Threshold
Forefoot strike	$\theta < -1.2$
Midfoot strike	$-1.2 < \theta < 6.0$
Heel strike	$\theta > 6.0$

These classifications were based upon findings from a study [6] but scaled by a factor of ¾. This was done because relatively, the calculated angles in the code are mostly accurate; however, the calculated angles are affected by coefficients in the Kalman filter like R and Q which were experimentally discovered. When rotated, to a supposed angle of 90 degrees, the calculated angle only read about 69 degrees, ~¾ of the actual angle.

For certainty, the determined foot strike pattern of the runner was given by the majority pattern of foot strikes. Given a class haptic feedback signals to the runner if there is any issue.

For the purpose of keeping the runner within a predetermined velocity, the average velocity over the interval was calculated from the list of velocities over the 5+ second interval where data was recorded. If the average is not within a threshold of ±10s/mile from the predetermined velocity, it is flagged as a high or low velocity and haptic feedback is triggered

Event vs response:

Response	Haptic feedback
Heel strike	One 0.5 second motor pulse
Forefoot strike	Two 0.5 second motor pulses
Low velocity	Three 0.5 second motor pulses
High velocity	Four 0.5 second motor pulses

4 – Results

Minimal testing was done to demonstrate the use of this shoe in an application. To test, I ran for a 60 second interval purposely heel, midfoot, and forefoot striking, with a desired pace set to 3.35 m/s.

Test Description	Correctly returned heel strike feedback	Correctly returned pace feedback
Heel strike, maintain 3.35 m/s	4/4, 4/5, 5/6	3/4, 4/5, 5/6
Midfoot strike, maintain 3.35 m/s	5/6, 4/5	4/6, 4/5
Forefoot strike, maintain 3.35 m/s	4/5, 4/6, 3/5	3/5, 4/6, 4/5

To determine whether pace feedback was returned correctly, I used an Apple Watch Series 7 to record instantaneous pace in min/mile. By comparing the apple watch pace data to the type of pace feedback given by the shoe, feedback was deemed correct or incorrect.

5 - Discussion

Overall, the results from testing the shoe were positive. With accurate results, this shoe shows the haptic feedback model's potential for enhancing running efficiency and preventing injuries. Though there are positive results, there is likely bias present, so it would be important to test a larger population of different runners and gaits to determine effectiveness. Additionally, this shoe is only a model that demonstrates the potential effectiveness it could have on runners. This model can be expanded upon to monitor more variables associated with a runner's gait like impact forces or pronation and supination.

While the haptic feedback that a vibration motor is effective, a quantity of buzzes has no correlation to a property of gait. If the range of properties of gait that the shoe measures was expanded, it would be necessary to implement a haptic feedback system that does not require the user to memorize that a quantity of buzzes is associated with some sort of feedback. A system that applied haptic feedback to different areas / portions of the shoe could help the user differentiate and interpret feedback.

5.1 Related Research

After this project was done, I learned of research—Morris, S. (2004). A Shoe-Integrated Sensor System for Wireless Gait Analysis and Real-Time Therapeutic Feedback—that aimed to tackle a similar problem to my project.

Named the GaitShoe, Morris developed a shoe of extensive capability which could measure and provide data on heel strike timing, toe off timing, dorsi-/plantar- flexion, stride length, and stride velocity. Morris also created a feedback system which could provide "rhythmic auditory stimulation" capable of pace sensing, force distribution sensing, and peak force sensing.

It is important to point out that the ideas and design choices formulated in my project were not copied from this project and also that there are key differences between the two.

While Morris's project was developed as a therapeutic oriented application (an athletic application was also realized), this project was oriented towards runners. Additionally, the Kalman filter and foot strike angle sensing model I developed is a key difference.

6 – Conclusion

The goal of this project was to develop a shoe that was capable of tracking gait related metrics to help enhance runner's efficiency and reduce injury-proneness. A prototype shoe was designed along with python software to achieve this goal. The software took measurements from an IMU and used those data points to determine angle of the foot at impact, magnitude of acceleration at impact, foot strike timing, and stride pace; the system analyzed this data in real-time to return haptic feedback. Overall, the results from testing show that this model is accurate and could potentially aid runners by training them to foot strike a certain way or to help them maintain a certain pace during a workout.

In the future, this model could be improved by adding extra features that would be useful to enhance the efficiency and reduce injury-proneness of a runner.

7 – References

[1] Smith, G. A. & Fewster, J. B. (1996). VARIABILITY OF GROUND REACTION FORCE CHARACTERISTICS FOR SLOW RUNNING SPEEDS 517. *Medicine & Science in Sports & Exercise, 28* (5), 87. https://oce.ovid.com/article/00005768-199605001-00517

[2] Winter D. A. (1982). Camera speeds for normal and pathological gait analyses. *Medical & biological engineering & computing*, *20*(4), 408–412. https://doi.org/10.1007/BF02442398

[3] Gao, J.-J., Wang, I-L., Gu, C.-Y., & Wang, L.-I (2016). Effects of shoes mass on running gait analysis. *International Society of Biomechanics in Sports.* https://ojs.ub.uni-konstanz.de/cpa/article/view/7003

[4] image used with permission from UC Berkeley MechE professor

[5] (https://www.kalmanfilter.net/kalman1d.html), Becker, A., (2023). *Online Kalman Filter tutorial*. Kalman Filter Tutorial.

[6] Hoenig, T., Rolvien, T., & Hollander, K. (2020). Footstrike patterns in runners: Concepts, classifications, techniques, and implications for running-related injuries. *Deutsche Zeitschrift Für Sportmedizin*, *71*(3), 55–61. https://doi.org/10.5960/dzsm.2020.424

[7] Morris, S. J., & Paradiso, J. A. (2004). Shoe-integrated sensor system for wireless gait analysis and real-time feedback. *Proceedings of the Second Joint 24th Annual Conference and the Annual Fall Meeting of the Biomedical Engineering Society] [Engineering in Medicine and Biology*. https://doi.org/10.1109/iembs.2002.1053379