Hardware and Ambient Variability in Botball Robots: Analysis and Mitigation

Mario Scholze
HTL St. Pölten, Austria
mario.scholze@htlstp.at

***Abstract*—Botball robots operate fully autonomously, making them prone to ambient variation and hardware differences. This paper identifies three of the most common sources of variation in Botball robots: battery voltage differences, manufacturing tolerances of motors and light sensors and ambient lighting affecting light sensors. Through DC motor equations, it is shown that a battery voltage difference directly causes motor speed differences of measurable magnitude. It is also shown that small variations in the motor's armature resistance and back-EMF constants can cause large differences in speed, which can cause the robots to miss their targets. With measured light sensor data, it is also shown how sunlight and other factors impact the light sensors. For each source of variability, mitigation strategies ranging from usable but inconvenient to very good are proposed, and some also tested and implemented. The benefits and tradeoffs of every mitigation method are also discussed.**

***Index Terms*—Hardware Differences, Ambient Differences, Battery Voltage Variance, Motor Variance, Sensor Variance, Botball, Robotics**

I. Introduction

In an ideal robot, there is no voltage sag, no motor differences and no parameters which could affect or alter the robot's performance. In a real-world robot, nothing of that is true.

Botball robots operate fully autonomously with zero human interaction for the entire run. This means that if they drift or deviate in an unpredictable way, there is a high chance that the robot will not be able to complete its designated task(s) anymore. Ambient lighting conditions present a similar risk, as prior work has shown that differences in lighting between testing and competition environments have caused sensor malfunctions at the ECER [1].

Example: In testing, the robot may work perfectly but swapping in a fully charged battery for the competition run changes the supply voltage from something like 6.1 V on a half-discharged battery-pack to around 7.0 V on a full one, a difference of 14.8%.

All assumed values are averages of real-world measurements to give the best accuracy.

II. The Impact of Battery Voltage Variations

To visualize the impact on motor performance (not accounting for hardware differences), battery voltages of $U_{\text{Testing}} = 6.1$ V and $U_{\text{Competition}} = 7.0$ V can be assumed to calculate the estimated motor performance for each of these voltages.

For that, the motor voltages will first be calculated using the equation $U_{\text{Motor}} = U_{\text{Open Circuit}} - I \cdot R_{\text{Internal}}$ [2], where:

- U_{Motor} is the voltage on the motor's terminals,
- $U_{\text{Open Circuit}}$ is the battery voltage when not under load,
- I is the current flowing through the motor,
- R_{Internal} is the battery's internal electrical resistance.

The average current used by a motor on a LiFePO4 battery was measured by running a motor on a battery with an ampere-meter attached, taking a few samples (around 20), and then averaging them out. The internal resistance was assumed to be 50mΩ, as it is hard to pinpoint an exact internal resistance for all kinds of batteries, as it depends on state of charge, battery wear and more. [3], [4]

If the voltages are inserted into the above mentioned formula, while using a measured average current draw of 200mA and a measured average internal resistance of 0.05Ω for the battery, it results in

- $U_{\text{Motor,Competition}} = 7.0\text{V} - 0.2\text{A} \cdot 0.05\Omega = 6.95\text{V}$
- $U_{\text{Motor,Testing}} = 6.1\text{V} - 0.2\text{A} \cdot 0.05\Omega = 6.05\text{V}$

For calculating the angular velocity of the motors, $\omega = \frac{U - I \cdot R_A}{K_E}$ can be used, where:

- ω is the motor's angular velocity,
- U is the voltage on the motor's terminals,
- I is the current flowing through the motor,
- R_A is the armature resistance (electrical resistance of the copper windings of the motor's coils),
- K_E is the back-EMF constant

This equation can be derived from the more detailed equation $L_A \cdot \frac{\mathrm{d}i_a}{\mathrm{d}t} = U - R_A \cdot I_A - K_E \cdot \omega$ [5]. Assuming a steady-state DC motor for simplicity, the inductive resistance (L_A) and the rate of change of the electrical current ($\frac{\mathrm{d}i_a}{\mathrm{d}t}$) disappear. If the equation then gets rearranged for the angular velocity ω, like this:

- $L_A \cdot \frac{\mathrm{d}i_a}{\mathrm{d}t} = U - R_A \cdot I_A - K_E \cdot \omega$, where $\frac{\mathrm{d}i_a}{\mathrm{d}t}$ is 0
- $0 = U - R_A \cdot I_A - K_E \cdot \omega$
- $K_E \cdot \omega = U - R_A \cdot I_A$
- $\omega = \frac{U - R_A \cdot I_A}{K_E}$

it results in $\omega = \frac{U - R_A \cdot I_A}{K_E}$, the formula for angular velocity of steady-state DC motors.

If the previously measured current draw of 200mA is taken and values of $R_A = 2\Omega$ and $K_E = 1\frac{\text{V}}{\text{rad/s}}$, which will later also be measured in Table I, are assumed, the following speeds can be calculated:

- $\omega_{\text{Testing}} = \frac{6.05\text{V} - 0.2\text{A} \cdot 2\Omega}{1\frac{\text{V}}{\text{rad/s}}} = 5.65\frac{\text{rad}}{\text{s}}$

- $\omega_{\text{Competition}} = \frac{6.95\text{V} - 0.2\text{A} \cdot 2\Omega}{1 \frac{\text{V}}{\text{rad/s}}} = 6.55 \frac{\text{rad}}{\text{s}}$

This means that if a new battery is swapped in, the speed of the robot increases by $\Delta\omega = 0.9 \frac{\text{rad}}{s}$, which is an increase of 15.9% in speed. Consequently, if the robot was traveling a distance of 15 cm using the old battery, it would travel 17.39 cm with the new one, assuming the robot is travelling time-based.

III. Hardware Differences in Motors

If an ideal two-wheeled robot gets instructed to drive forward by setting both motors to a speed of 100%, it will move exactly like this:

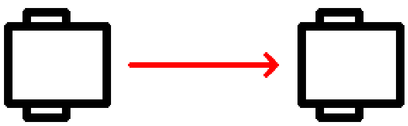

Fig. 1. An ideal robot driving forward

If a real robot is instructed to do the same, its path will likely look like this:

Fig. 2. A real robot driving forward

This happens because when manufacturing motors, it is impossible to build every single motor identical to one another. Small variations in magnet strength, armature resistance and more create these imperfections. The impact of these differences can be proven by again using the DC motor speed formula $\omega = \frac{U - I \cdot R_A}{K_E}$ to calculate the speed of the slowest and fastest motors.

To illustrate this with real measurements, Table I shows the measured parameters of three motors at a supply voltage of $U = 6.6\text{V}$:

TABLE I
Measured motor parameters at $U = 6.6\text{V}$

Motor	ω ($\frac{\text{rad}}{s}$)	I **(A)**	R_A **(Ω)**	K_E ($\frac{\text{V}}{\text{rad/s}}$)
Motor 1	5.731	0.150	1.7	1.107
Motor 2	5.815	0.160	2.0	1.080
Motor 3	5.973	0.250	1.7	1.034

These values were figured out by connecting each motor to a 6.6V constant voltage source, letting the motor run for 20 seconds and noting down its number of rotations, from which it was possible to derive ω using the equation $\omega = \frac{n_{\text{Rotations}}}{20} \cdot 2\pi$. The motor's average current was also determined by letting it run on the same 6.6V voltage source. R_A was measured using an ohmmeter. Using all these now known values, it was possible to solve the equation $\omega = \frac{U - I \cdot R_A}{K_E}$ for K_E by rearranging to get $K_E = \frac{U - I \cdot R_A}{\omega}$.

Taking the measured extremes of $R_{A,\text{min}} = 1.7\Omega$ (Motors 1 and 3) and $R_{A,\text{max}} = 2.0\Omega$ (Motor 2), and assuming $U = 6.6\text{V}$, $I = 0.2\text{A}$ and $K_E = 1.08 \frac{\text{V}}{\text{rad/s}}$, the results are:

- $\omega_{R_A,\text{min}} = \frac{6.6\text{V} - 0.2\text{A} \cdot 1.7\Omega}{1.08 \frac{\text{V}}{\text{rad/s}}} \approx 5.796 \frac{\text{rad}}{s}$
- $\omega_{R_A,\text{max}} = \frac{6.6\text{V} - 0.2\text{A} \cdot 2.0\Omega}{1.08 \frac{\text{V}}{\text{rad/s}}} \approx 5.741 \frac{\text{rad}}{s}$

This difference of $\Delta\omega \approx 0.055 \frac{\text{rad}}{s}$, or 1%, is caused by an ΔR_A of just 0.3Ω across the measured motors.

Taking the measured extremes of $K_{E,\text{min}} = 1.034 \frac{\text{V}}{\text{rad/s}}$ (Motor 3) and $K_{E,\text{max}} = 1.107 \frac{\text{V}}{\text{rad/s}}$ (Motor 1), and assuming $U = 6.6\text{V}$, $I = 0.2\text{A}$ and $R_A = 1.8\Omega$, it results in:

- $\omega_{K_E,\text{min}} = \frac{6.6\text{V} - 0.2\text{A} \cdot 1.8\Omega}{1.034 \frac{\text{V}}{\text{rad/s}}} \approx 6.035 \frac{\text{rad}}{s}$
- $\omega_{K_E,\text{max}} = \frac{6.6\text{V} - 0.2\text{A} \cdot 1.8\Omega}{1.107 \frac{\text{V}}{\text{rad/s}}} \approx 5.637 \frac{\text{rad}}{s}$

This is a difference of $\Delta\omega \approx 0.398 \frac{\text{rad}}{s}$, or 7.1%, caused by a $\Delta \frac{\text{V}}{\text{rad/s}}$ of only 0.073.

Combining both effects and using the actual measured values from Table I for the slowest and fastest motors:

- $U = 6.6\text{V}$
- $I_{\text{Motor1}} = 0.150\text{A}$, $I_{\text{Motor3}} = 0.250\text{A}$
- $R_{A,\text{Motor1}} = 1.7\Omega$, $R_{A,\text{Motor3}} = 1.7\Omega$,
- $K_{E,\text{Motor1}} = 1.107 \frac{\text{V}}{\text{rad/s}}$, $K_{E,\text{Motor3}} = 1.034 \frac{\text{V}}{\text{rad/s}}$

it is possible to calculate both motors' rotational speeds:

- $\omega_{\text{Motor1}} = \frac{6.6\text{V} - 0.150\text{A} \cdot 1.7\Omega}{1.107 \frac{\text{V}}{\text{rad/s}}} \approx 5.732 \frac{\text{rad}}{s}$
- $\omega_{\text{Motor3}} = \frac{6.6\text{V} - 0.250\text{A} \cdot 1.7\Omega}{1.034 \frac{\text{V}}{\text{rad/s}}} \approx 5.972 \frac{\text{rad}}{s}$

This difference of $\Delta\omega \approx 0.24 \frac{\text{rad}}{s}$, or 4.2%, manifests as the robot driving an arc instead of a straight line, as seen in Fig. 2.

IV. Light Sensor Variability

With analog sensors, there are two types of variability: hardware and ambient. For these examples, the old botball tophat sensor (discontinued after the 2023 season), shown in Fig. 3, will be used, as it's a good candidate for demonstrating both types of variability. There is also a new version of the tophat sensor (since the 2024 season), shown in Fig. 4.

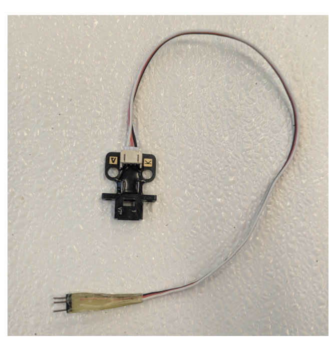

Fig. 3. The old botball tophat sensor (EE-SB5) [6]

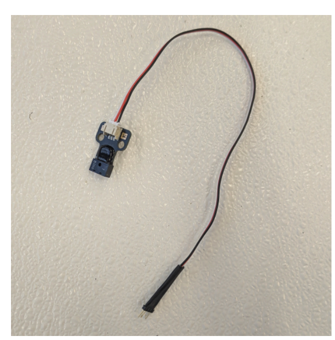

Fig. 4. The new botball tophat sensor (EE-SF5) [7]

The two sensor variants were tested in normal and sunny lighting conditions and with multiple distances to the ground. In Fig. 5, the setup to test the sensors can be seen.

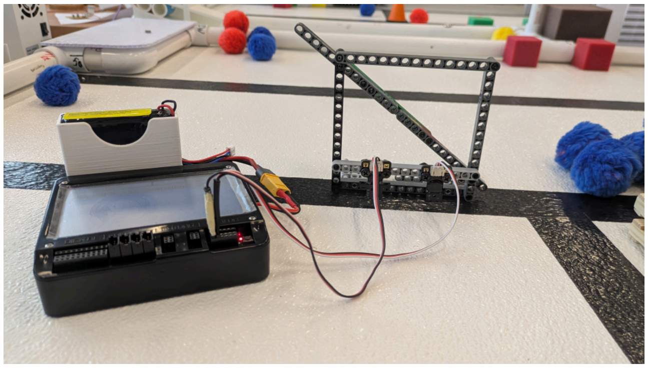

Fig. 5. The setup used for determining the sensor values in Table II and Table III

TABLE II
MEASURED TOPHAT READINGS AT A DISTANCE OF 7MM TO THE FLOOR

Sensor	Normal Light		Direct Sunlight	
	White	Black	White	Black
Old Tophat	180	2930	50	170
New Tophat	1270	3800	140	250

TABLE III
MEASURED TOPHAT READINGS AT A DISTANCE OF 23MM TO THE FLOOR

Sensor	Normal Light		Direct Sunlight	
	White	Black	White	Black
Old Tophat	1830	3810	50	130
New Tophat	3440	4030	130	220

For the values, three samples per category were taken and averaged out. Normal light means the measurements were taken with only overhead lighting and no direct and only negligible indirect sunlight. Direct sunlight means the measurements were taken with overhead lighting and direct sunlight, with the only object between the sun and gametable being a glass window.

A. Hardware Differences in Light Sensors

Just like with motors, there are no identical tophat sensors, as it is impossible to create two sensors with the exact same IR LED brightness, photodiode sensitivity and more. Differences in sensor revision [6], [7] are also an important factor. The mounting distance is important as well, as it directly influences the reliability of the sensor.

a) *Sensor Generation Differences:* As shown in Table II, it is observable that the new tophat reads 1270 while the old tophat reads 180 at a distance of 7mm to the floor, which is a difference of 1090. This means that if the threshold were to be set to 1000, switching to the new generation would cause issues.

b) *Mounting Distance Effect:* If the readings from the old tophat sensor with only normal light and a distance of 7mm are used, it results in around 180. At a distance of 23mm, the value rises to 1830. Once again, this demonstrates that a threshold of for example 1000 would not work if the height of the tophat sensor was changed. For the new sensor, this is way worse. The value is 1270 for the new sensor with only normal light and a distance of 7mm, whereas with 23mm, it is 3440. When taking into account that the value for black for the same sensor (23mm, normal light) is 4030, the difference becomes rather small.

B. Ambient Light Variations

If the old sensor is used with a distance of 7mm and measured under normal light conditions, it results in a value of around 180 for white and 2930 for black. If the sensors are measured again with direct sunlight, it results in approximately 50 for white and 170 for black. This is only a difference of 120, which is negligible compared to our difference of 2750 for normal conditions. This small difference makes it very unreliable to operate the sensors without constant calibration, as just a shadow moving by could easily spike the values by several hundred.

V. MITIGATION STRATEGIES

A. Battery Voltage Difference Mitigation

The simplest way to ensure consistent motor speed is to verify that the battery's state of charge is always the same, by for example swapping in a fresh battery after every run. This, of course, is annoying and prone to errors, because if a half-discharged battery is accidentally grabbed, the speed will not be as expected.

A better approach is to measure battery voltage and scale motor output accordingly. If a reference voltage of $U_{\text{Reference}} = 7.0V$ is assumed, which should ideally be the maximum battery voltage that will be used, the battery voltage can be continuously measured and a factor can be determined, which will be k in this example. This factor is calculated by using the equation $k = \frac{U_{\text{Reference}}}{U_{\text{Measured}}}$. All motor output commands can then be multiplied with this factor to scale them to match the battery voltage difference to the reference voltage $U_{\text{Reference}}$. The big problem with this method is that if it is attempted to scale a motor power of 100% with a factor of for example $k = \frac{U_{\text{Reference}}}{U_{\text{Measured}}} = \frac{7.0V}{6.1V} \approx 1.148$, the motors won't be able to turn faster than 100%, as that's their maximum speed. This means that this method only works reliably if the robot drives at for example 80% speed.

A more reliable and robust solution would be a closed-loop system, meaning how much has already been driven gets measured to be sure of the distance that's been traveled. This is possible through back-EMF sensing, where back-EMF voltages, which get generated when a motor turns, are used to measure how much the motor has turned [8]. Although this is a very good solution, it does require knowing the back-EMF constant of the motor, which needs calibration to figure out. When this form of feedback is combined with with a closed-loop speed controller (like a PID controller), it results in a very precise and reliable driving correction. [9]

Another possibility to correct motor mismatch is to have an external sensor like an IMU to correct your heading. One of the best methods would be to combine back-EMF sensing and IMU fusion.

This last method is also the exact one used in the shared robotics library and toolchain built by the teams at the HTL St. Pölten, which is called RaccoonOS [10] but also known as Raccoon. The library implements an odometry system, which corrects the robot's heading using back-EMF values and the IMU.

B. Mitigation of Motor Hardware Differences

The simplest fix is to physically swap motors until motors that closely match in their hardware values like armature resistance R_A and back-EMF constant K_E are found, then keep these for all testing and competition runs. While this works, it is fragile: if a motor fails or is replaced, the calibration is lost and the process must be repeated.

A more robust fix is to apply a static compensation constant, which will be called k in this example. Assuming motor 2 is the weakest motor, the maximum power of motor 2 (the weakest motor) is defined as the global maximum motor power. Then, one k constant is defined for every motor. If motor 1 is 3% faster, motor 3 is 1% faster and motor 4 is 5% faster, the values will be $k_{\text{Motor1}} = 0.97$, $k_{\text{Motor2}} = 1$, $k_{\text{Motor3}} = 0.99$ and $k_{\text{Motor4}} = 0.95$. Then, the power of every motor is multiplied with that motor's k factor, like this: `motor(0, 100 * k_Motor1)`. This strategy is relatively good, but as soon as motors are swapped out, significant re-calibration effort is required for the new k constant.

Once again, the best solution is closed-loop control. If the motor's back-EMF values are constantly monitored, it is possible to calculate how much each wheel has already turned. With that information, it can be calculated how the faster wheels need to be slowed down to compensate for the slower wheels.

This last method is also used in the raccoon library.

C. Analog Sensor Difference Mitigation

For semi-stable or slowly changing ambient light, calibrating the light sensors by taking a highest and lowest value and approximating a midpoint as threshold on program startup is a good enough option, although it is not suitable if the light is changing at a fast pace.

If the light is very unstable and/or quickly changing, on-the-fly calibration is needed: when driving around, sensor readings are constantly collected. As soon as there is enough variation in sensor values, it's possible to compute a midpoint/threshold. This only works well if the surface underneath features much variation in bright and dark.

The library uses a more advanced method, which is to detect clusters of values using machine learning, which represent the values for black and white. From that, a threshold is computed.

VI. Conclusion

This paper examined three principal sources of uncontrolled variability in Botball robots: battery voltage differences between testing and competition, DC motor manufacturing tolerances, and ambient lighting affecting tophat sensors. Through quantitative analysis using DC motor theory, it was shown that even small physical differences, such as a $0.073\frac{\text{V}}{\text{rad/s}}$ spread in back-EMF constants across three motors, can produce speed deviations of 7.1%, and combining all parameter differences yields a 4.2% speed deviation between the slowest and fastest motors, which is large enough to cause a robot to drift off course. Battery voltage alone can account for a 15.9% speed difference between a half-discharged and a fully charged pack.

For each source of variability, mitigation strategies were evaluated. Simple manual strategies such as swapping components and hard-coded thresholds provide a workable baseline but are fragile and require time-consuming re-calibration whenever hardware changes. Static compensation factors improve reliability without additional hardware, but remain open-loop and cannot respond to conditions changing mid-run. Closed-loop feedback, combining back-EMF sensing with a PID speed controller, represents the most robust motor mitigation, correcting for both voltage sag and motor-to-motor differences simultaneously. For sensors, per-sensor startup calibration reliably handles hardware variation and generation differences, while on-the-fly threshold adaptation addresses rapidly changing ambient lighting conditions.

References

[1] M. Schorf and N. Schechora, “The Impact of Light Intensity on Color Perception,” 2025.

[2] M. Soca, “BATTERY INTERNAL RESISTANCE INFLUENCE ON VOLTAGE-BASED BALANCING ALGORITHMS,” 2021.

[3] M. N. H. W. S. D. A. A. M. A. Purwanto A. Jumari, “Improving cylinder-type LiFePO4 battery performance via control of internal resistance,” 2018.

[4] T. M. T. R. J. Sihvo D. Stroe, “Fast Approach for Battery Impedance Identification Using Pseudo-Random Sequence Signals,” 2020.

[5] F. C.-C. M. A.-C. J. M. A.-M. G. S.-G. R. Silva-Ortigoza C. Márquez-Sánchez, “Hierarchical Velocity Control Based on Differential Flatness for a DC/DC Buck Converter-DC Motor System,” 2014.

[6] “EE-SB5 Photomicrosensor (Reflective) Datasheet, Accessed 2025-03-14.” 2021. [Online]. Available: https://omronfs.omron.com/en_US/ecb/products/pdf/en-ee_sb5.pdf

[7] “EE-SF5/EE-SF5-B Photomicrosensor (Reflective) Datasheet, Accessed 2025-03-14.” 2022. [Online]. Available: https://components.omron.com/eu-en/sites/components.omron.com.eu/files/datasheet_pdf/E485-E1.pdf

[8] D.-M. L. J.-S. R. D. H. Tae-Sung Kim Byoung-Gun Park, “A New Approach to Sensorless Control Method for Brushless DC Motors,” 2008.

[9] N. M. U. Kumar M. Nisha, “Tracking of a PID Driven Differential Drive Mobile Robot,” 2017.

[10] “RaccoonOS.” [Online]. Available: https://github.com/htl-stp-ecer