# Ground Barcode Reading for Warehouse Robotic Systems

2179734
*Department of Engineering Mathematics*
*University of Bristol*

2185253
*Department of Engineering Mathematics*
*University of Bristol*

***Abstract*—In the aim to use warehouse robots to the fullest of their potential, this report investigates how the asynchronous reading of linear ground barcodes can be optimised and modelled for different robot hovering speeds and barcode bit sizes. By running several experiments under these conditions, it has been found that the bit-reading operation's success is non-linear with regards to both hover speed and bit size, presenting optimums at 140 mm/s and 12 mm respectively. A deeper look into error sources is presented, and a linear model taking into account both temporal and spatial lags is proposed to explain the non-linear bit-reading abilities that were obtained.**

***Keywords:* - Barcode reading, Line-following robot, Arduino Programming**

## I. Introduction

Motivated by higher efficiency and the removal of human error, the mainstream use of robots in warehouses arrived earlier than anybody expected [1]. The smart warehouse field is not even yet at its peak, with currently high numbers of emerging competitors besides the well-known Amazon robot (ex-Kiva robot).

While prior research has mainly focused on path planning with track-following robots, more recent research explored the possibilities of multi-robot warehouse systems path planning in trackless environments [2], where the use of QR codes transmits the needed orientation capabilities to each individual robot [3].

As scanning QR codes requires a developed camera system, more simple warehouse configurations may need to read information in the form of a single barcode to provide the robot with basic heading information, task assignment or to inform about the system's abstract borders. Because of the simple configurations in which these operations are usually carried out, the reading of the barcode cannot be done with a barcode reader nor with a camera, but with basic reflectance sensors. The reading of these codes happen in a linear way, at different reading speeds and for varying physical sizes of bits. It currently unclear what the range of speeds and information density are for data to be successfully picked up, yet this could enable warehouse robots to be used at their highest efficiency, saving space with smaller bit sizes and saving time with higher reading speeds, while ensuring reading success.

In this view, the present research makes use of a single reflectance sensor on Polulu's $3\pi+$ robot to quantify the efficiency of a linear black and white barcode asynchronous reading operation in terms of reading speed and physical bit size.

### A. Hypothesis Statement

1) It is expected to find the same results for an increased bit size as for a reduced robot speed, as both should lead to better reading performances and are intimately linked to the robot's barcode-sampling frequency.

2) It is hypothesised that none of the reading errors come from the ground sensors misinterpreting a bit. Instead, all reading errors are expected to come from a spatio-temporal parasitic shift of the robot with regards to the bit. Because this parasitic shift is adding up at every bit, the sample picked up by the robot inevitably ends up being from a wrong bit.

3) By driving the robot's wheels with a closed-loop PID controller that relays on actual motor rotation measured by external encoders, the motors' lower power dead zones should not be a limit for minimum speed. Instead, at lower and medium speeds, the wheels' PID is expected to be the limiting factor in delivering wheel speeds with limited accuracy with regards to the target speeds, hence inducing biases in the sampling frequency.

4) At high speed, parasitic temporal lags translate into important parasitic spatial shifts, which is thought to eventually become the limiting factor for successful code-reading. It is thus hypothesised that high reading speeds result in much less reliable code-reading.

These hypotheses are investigated through a structured experiment, running several tests of barcode-reading at different velocities and for various bit sizes.

## II. Experiment Methodology

### A. Overview of Method

The experiment is conducted in the environment described by Figure 1 and is divided into four phases: Calibration, line following, barcode reading, and data retrieving. The setup consists of flat regions alternating black and white strips (bits) of equal widths, called *ladders* in the rest of the report, separated by a constant black line.

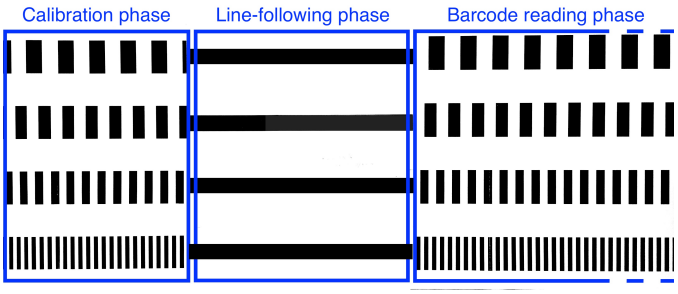


Fig. 1: Description of the overall experimental setup, divided in three physical phases.

In the first phase, the calibration of the robot's reflectance sensors is conducted to alleviate effects of ambient light on the sensors. During this phase, the robot is manually positioned on the left ladder facing the line, after what it advances and records measurements for the darkest and brightest surfaces for calibration. The functioning of the reflectance sensors and the calibration algorithm are further described in Section III.

Following the calibration, the robot performs a line following behaviour on a straight black line at a set distance to prepare for barcode reading. This allows the robot to be travelling at a constant velocity and in a straight line when it reaches the start of the bit reading phase, as well as to remove all inevitable human noise related to the robot placement in the calibration phase.

Next comes the barcode reading phase, where the robot moves over a barcode repeatedly alternating black and white bits of equal widths and samples the ground's reflectance at a fixed frequency, distinguishing black from white bits. As the robot moves over the ladder, the bits' brightness, the positions of the bits' edges and the position at which the bit's sampling is taken are recorded for each bit. The robot is programmed to expect an alternation of brightness and stops whenever it reads the same values in two consecutive readings, assessing its success automatically. Having the specific pattern of a ladder ensures no one bit can be read incorrectly yet still return the right value, as bits' neighbours are of different brightness for every bit.

Lastly, when the robot auto-assesses a misread, the data-retrieving phase starts: the robot stops moving and the data can be retrieved once plugged into a computer.

### B. Discussion of Variables

- **Controlled Variables**: The physical sizes of the bits are fixed by the ladders, which are programmed to have bits of similar size and later printed. The control of the reading speeds is handled by a PID controller, and the speed is ensured to be constant thanks to the line-following phase. Moreover, every trial is carried out under the same diffused light, where any temporal distortion in light intensity is filtered out through calibration before each run, each of them lasting between 5 and 15 seconds. Printers ensure that colours are homogeneous all over the experiment environment, with no difference in black or white shade, offering meaningful credibility to calibration. Lastly, rechargeable batteries were used for the robot and were recharged between every series of experiments on one ladder to eliminate all effects linked to battery level.
- **Independent Variables**: The experiment consists of four repetitive runs on every unique combination of hover speed and bit size, which are changed between 4, 8, 12 and 16 mm and 60, 100, 140, 180 and 220 mm/s respectively, which adds up to a total of eighty trials. Because bit size relates to a change in environment and hover speed a change in software, these variables are independent from each other and can easily be tuned individually.
- **Dependent Variable**: The change in bit-reading abilities will be assessed by observing the number of bits that can be read by the robot. Associated metrics are described in Section II-C.

### C. Discussion of Metrics

Because the number of bits that can be read correctly determines the complexity of the information that can be transmitted in real conditions, this will constitute the main metric evaluated during the experiments. By taking the mean of the number of bits read over runs with similar parameters, we can quantify the performance of the robot for a given hover speed and bit size. A disadvantage of this metric is that the ladder has a fixed full range, thus reduced sensitivity when combining trials with very low and very high success.

It is desirable to observe to what extent the robot can pick up information from the barcode both accurately and precisely. While the accuracy is rated by the aforementioned metric, the precision can be assessed by looking at the position within each bit at which the readings are taken (Figure 2). A Sampling Position Score (SPS) can be derived from that position, where 0.5 is the ideal central sampling position, values close to 1 translate a sampling close to the next bit's edge and scores close to 0 translate a sampling close to the previous bit's edge.

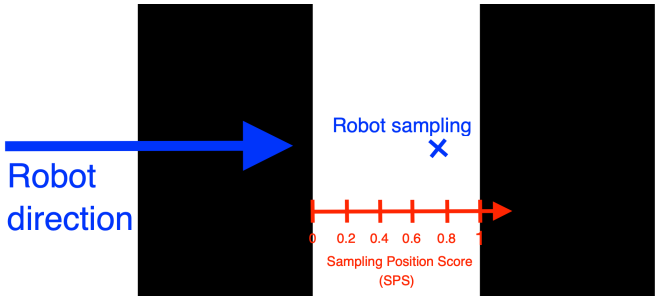


Fig. 2: Sampling Position Score (SPS) within a bit, determined by where the robot takes the brightness sample (about 0.78 in this example)

## III. Implementation

This section provides a description of how the previously described experiment methodology is effectively carried out, both from a hardware and software point of view.

### A. Robotic implementation and limitations

The robot used for the experiments is the 3$\pi$+ robot [4] from the Polulu brand, equipped with an ATmega32U4 microcontroller [5].

*1) Reflectance sensors:* The 3$\pi$+ robot has five reflectance sensors orientated towards the ground, amongst which the three central ones will be used for line-following and the unique central sensor for barcode reading. The functioning of these is relatively simple and can be understood from Figure 3. An infrared LED shines the surface underneath the robot, and based on the flux of light the photodiode perceives, a capacitor is discharged in a given amount of time. Depending on the surface's reflectance, the perceived light is altered which varies the discharging time as indicated in Figure 4. Darker surfaces register a longer capacitor discharging time than brighter surfaces do, which enables the sensors to characterise the hovered surface and will be used throughout the experimentation for the line-following as well as to discriminate the barcode bits' levels.

The line sensors will be calibrated during the calibration phase, which will lead to a normalisation of the discharging time lengths as per the following formula:

$$t_{normalised} = S_{calibration} * (t - t_{white})$$

Where $S_{calibration} = \frac{1}{t_{black} - t_{white}}$, $t_{black}$ and $t_{white}$ are the calibration factor, the recorded time durations for the darkest and brightest surfaces respectively.

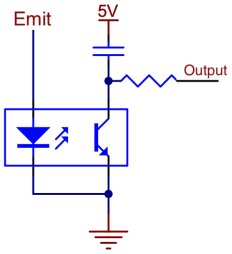


Fig. 3: Reflectance sensors' electrical layout, which includes an IR LED and photodiode discharging a capacitor

*2) Intrinsic speed characterisation:* A characterisation of the robot's actual speed compared to the input speed was carried out after the fact. The results, displayed in Figure 5, show a negative static error, meaning the robot runs slower than expected, with increasing errors for increasing speeds. This may come from intrinsic physical errors that plug into the robot's distance processing, such as uncertainty in the robot's wheels diameter and wheel axis length.

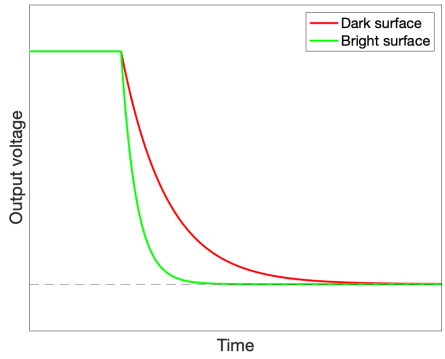


Fig. 4: Reflectance sensors' output voltage over time for different surfaces, showing how dark surfaces lead to longer discharging times

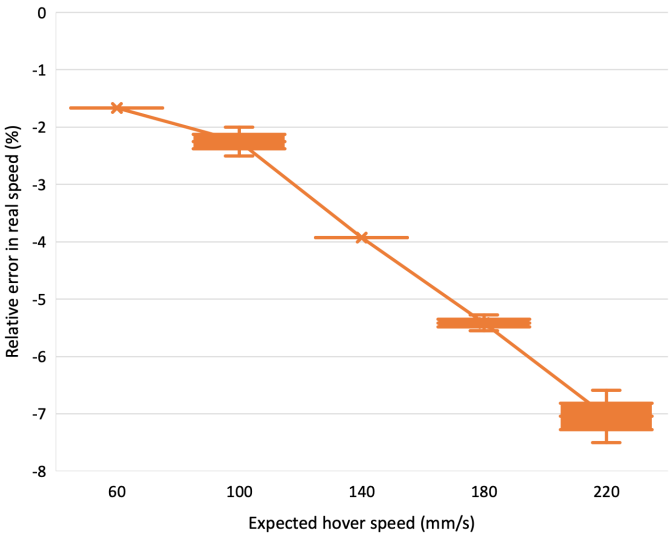


Fig. 5: Robot's real-speed relative error regarding expected hover speed, showing an increasing error as speed increases

### B. System architecture

As previously discussed in Section II, the robot will undergo different phases during the experimentation, in the following order: Calibration, line-following, barcode-reading and data-retrieving. These are implemented in the system architecture under the form of a Finite State Machine (Figure 6), and the various background-tasks' functioning are briefly described below.

In the line-following phase, a Nested PID Control [6] is implemented on the three central line sensors to control the heading direction of the robot, and a PID controller is used on each wheel to monitor their rotational speeds. The feedback signal from the line sensors determines the speed at which each wheel is driven. As the error approaches zero, the speeds of both wheels reach the same constant value, hence maintaining a straight path at a fixed velocity.

The barcode reading phase is triggered by the detection of the line's end. From there on, a timer's interrupts will

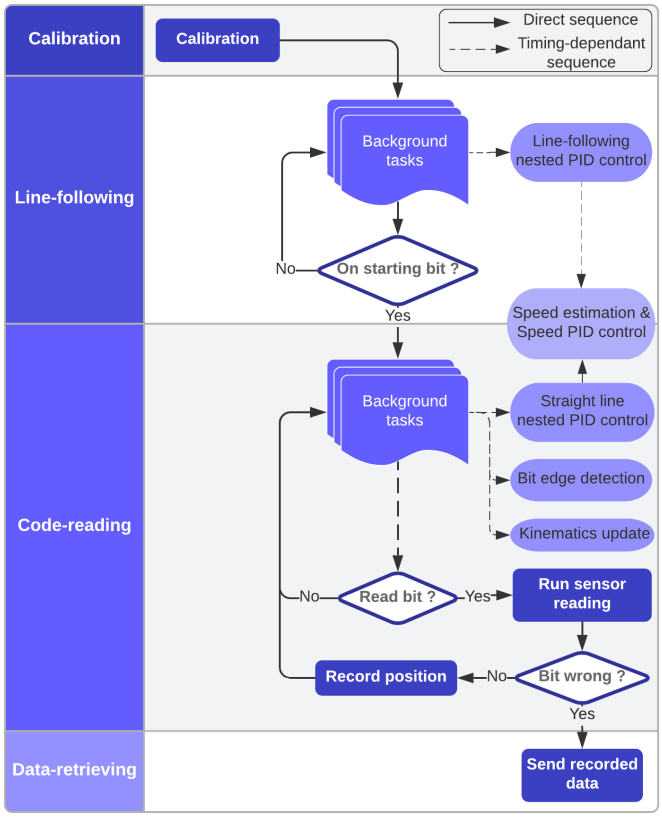


Fig. 6: Software architecture, constituted of a four-state Finite State Machine

command when to sample a bit's brightness with the foremost reflectance sensor, at a frequency depending on the hover speed and the bit sizes, as per the following formula:

$$f = \frac{V_{hover}}{W_{bit}} \tag{1}$$

The reading was designated to approximately take place at the centre of each bit by setting the timer's counter at half of its maximum count by the end of line following phase.

In the barcode reading phase, PID controllers for both heading direction and wheel speeds continue to maintain straight path at a controlled speed. To ensure following a straight path, the robot uses interoreceptive kinematics coupled to the wheels' encoders to keep an overall null rotation. The kinematics implementation also allows the robot to compute its position on an arbitrary plane and to register the position of the bits' edges as well as the position at which the sampling happens, which is then used to compute the Sampling Position Score (previously described in Section II-C) with the following formula, which assigns a value between 0 and 1:

$$SPS = \frac{x_{sampling} - x_{previous\ edge}}{x_{next\ edge} - x_{previous\ edge}}$$

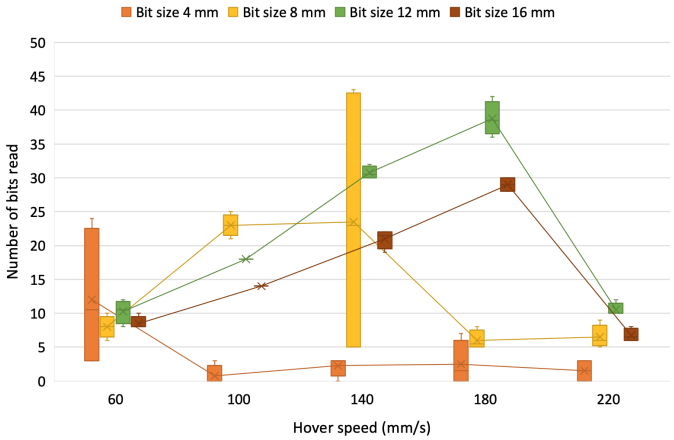


Fig. 7: Number of bits successfully read by the robot at different hover speeds. For instance, the 100 mm/s and 8 mm trials delivered between 22 and 24 successful bit readings

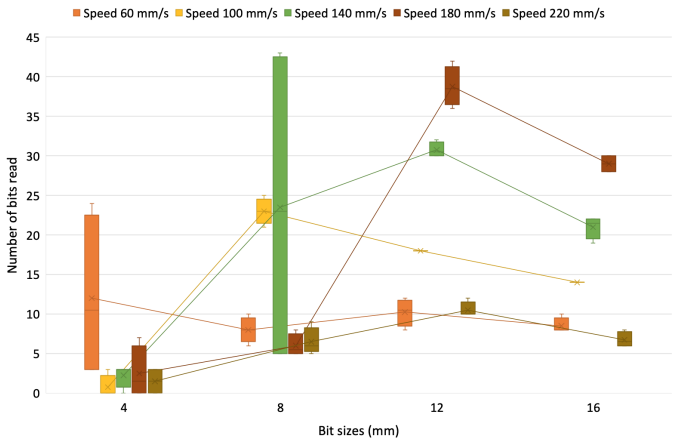


Fig. 8: Number of bits successfully read by the robot for different bit sizes. For instance, the 140 mm/s and 12 mm trials delivered between 30 and 32 successful bit readings

## IV. Results

### A. Main experimental results

The results obtained from the experiments are presented in Figure 7 and 8, showing interesting nonlinear behaviours. From these results, we can observe the following:

1) Low speeds and small bit sizes lead to poor bit reading, as do very high speeds.
2) There exists an optimal hover speed and bit size, respectively around 180 mm/s and 12 mm, maximising the number of bits that can be correctly read.
3) There is a great uncertainty at a hover speed of 140 mm/s and a bit size of 8 mm, which can be explained by results found in Section IV-B.

### B. Sampling Position Score (SPS) results

For each unique combination of hover speed and bit size, a log of each bit's SPS was recorded along the robot's path and retrieved to form the insightful graphs in Figure 9 and 10.


Fig. 9: Evolution of bits' SPS along runs at different hover speeds but constant bit size (16 mm)

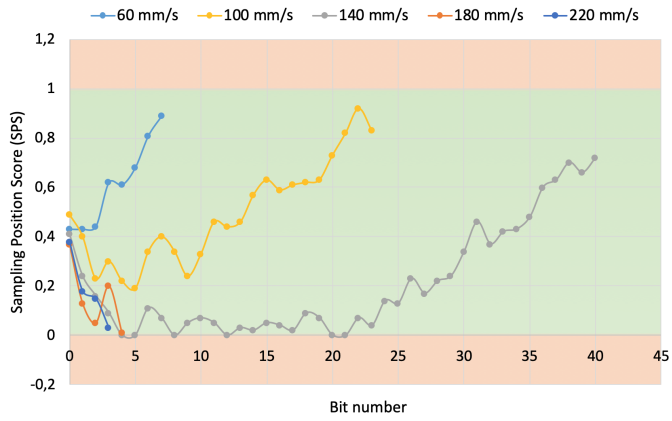


Fig. 10: Evolution of bits' SPS along runs at different hover speeds but constant bit size (8 mm)

As a reminder, an SPS approaching 0 or 1 leads to a failure of the bit-reading as this corresponds to either trying to read the previous bit ($SPS \leq 0$) or next bit ($SPS \geq 1$). These graphs provide very helpful insight on the robot's interaction with the ladders, but for now the following can be observed:

1) The SPS approximates a parabola over the course of the run, indicating that there are error sources both pushing the SPS towards 0 - translating either temporally fast or spatially slow robot behaviours - and towards 1 - translating either temporally slow or spatially fast behaviours. The failure to read a bit thus has a non-linear origin as the failing can be due to hitting both the lower and upper limit of SPS.
2) The difference in initial SPS gradient indicates that the cause of the SPS's decrease seems to be more pronounced at higher speeds, or the cause of the SPS's rise more pronounced at lower speeds.
3) Smaller bit sizes results in an overall drop in SPS, sometimes enabling many more bits to be read.
4) Oscillations, particularly visible in Figure 9 at 180 mm/s, come from the sensors' intrinsic different dynamics when detecting black and white bit edges.
5) The great uncertainty of the 8 mm bit size and 140 mm/s speed runs can be understood by looking at the grey parabola in Figure 10. Because the SPS comes very close to 0 early on before increasing, this sometimes leads to only five bits read and sometimes all the way to forty depending on small intrinsic variations.

## V. Discussion

### A. Error sources review and simulation

Based on the findings offered by the SPS plots, a review of the different error sources was carried out to explore the reason behind their non-linearity.

The first error source was previously detailed in Figure 5 and describes a shift in the robot's real speed compared to what is asked from it. Because this shift is negative, meaning the robot is spatially slower than expected, it is responsible for a decrease in the SPS. This decrease can be observed in Figure 9 and 10 and is particularly prevailing at high speeds, which is in accordance with Figure 5's increasing errors over speed.

While the speed error can be responsible for a decrease in SPS, the origin of the SPS increase is yet to be identified. By remeasuring the size of the bits with an increased accuracy technique, we find that a 3.13% error in all the bits' sizes was introduced in the robot's programming, which made the robot overestimate the bits' size. Through the sampling frequency calculation in equation 1, this translates into a temporal lag and thus in an increase of SPS over time, making this error a good candidate for this increase.

Exactly how these error sources - and other error sources - affect the final result is complex and non-linear as Figure 9 and 10 show. However, the linear impact of these spatial and temporal lags can be assessed mathematically when considered individually.

The spatial lag, originating from a shift in effective velocity $\Delta V$, would on its own lead to a failure after a time $t_{failure}$ that is directly related to the time needed for the velocity shift to contribute a distance of half a bit size $\frac{W}{2}$:

$$t_{failure} = \frac{W}{2 * \Delta V} = \frac{X_{error}}{V}$$

As this time is also represented by the real distance $X_{error}$ covered by the robot at its real speed $V$ upon failure, we find:

$$n_{error} := \frac{X_{error}}{W} = \frac{1}{2}\frac{V}{\Delta V}$$

$n_{error}$ thus represents the maximum number of readable bits when solely taking the spatial lag into account. $n_{error}$, plotted in Figure 11 in yellow, thus depends on the robot's real speed and velocity shift.

The temporal lag is due to the bit size encoded in the robot $W^*$ being different than a bit's real size $W$, as per $W^* = W + \Delta W$. From this, we conclude the following scale difference in $X$ and $X^*$:

$$\frac{X^*}{X} = \frac{W^*}{W} = 1 + \frac{\Delta W}{W}$$

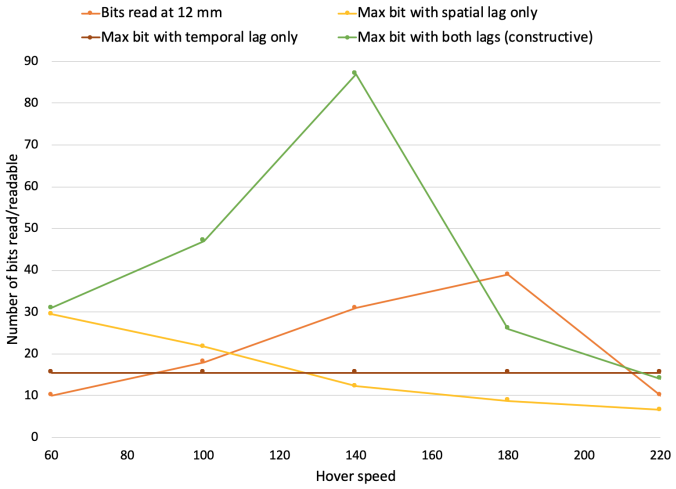


Fig. 11: Comparison between the mean number of bits experimentally read at 12 mm and the simulated theoretical limits of readable bits

Where $\alpha := \frac{\Delta W}{W}$ is known and equals 3.13% independently of the bit's size. By imposing $X^* - X_{error} = \frac{W}{2}$, which corresponds to the distance $X_{error}$ where the difference in the robot's scale and real scale inevitably leads to a reading error, we find:

$$n_{error} = \frac{X_{error}}{W} = \frac{1}{2}\frac{W}{\Delta W}$$

As $\alpha = \frac{\Delta W}{W}$ is constant, $n_{error}$ is constant over all speeds and bit sizes, as shown in Figure 11.

But because these two error sources lead to a rise and a fall in SPS simultaneously, they cancel each other out and enable the reading of more bits than either of their limits. By simulating a constructive interaction between these two error sources in Matlab, a maximum bit number was calculated at each hover speed and is also plotted in Figure 11, in green.

Figure 11 shows that linear errors can indeed lead to non-linear abilities in bit-reading, as obtained during the experiments (Figure 7). However, the parabolic evolution of SPS over a run means the linear simulation is only a simplification of the real interactions happening between the robot and its environment, which is reminded by the experiments achieving better results than predicted by the simulation, as Figure 11 indicates at the 180 mm/s hover speed.

### B. Hypothesis review

Here we discuss the initial hypotheses in Section I-A and the directions to focus future research in order to better characterise a robot's interactions with its environment in a barcode reading operation.

Concerning *H1*, we conclude that because Figure 7 and 8 both present non-linear behaviours, the relation between bit size and hover speed is more complex than predicted. Unlike what was hypothesised, reasonably increasing hover speed improves bit reading given the spatio-temporal errors cancelling out, until it becomes the main error source and indeed leads to failure. Bit size also presents an optimum for successful bit-reading, yet an explanation cannot be provided here as both error sources studied are independent of the bit size. It is proposed that intrinsic temporal lags may be the reason behind the SPS's quadratic rise, which means the rise in SPS gets more important as time goes by, hence allowing smaller bit sizes to be more successful for given time durations. This comes from the observation that for the same hover speed (of 100 mm/s for instance), the robot's SPS's rise comes in after more bits when travelling on a 8 mm ladder (Figure 10) than on a 16 mm ladder (Figure 9), but approximately after the same amount of time.

Hypothesis *H2* is hard to verify in depth, but considering failures systematically happened when the SPS approached 0 or 1 (See Figure 9 and 10), it does indeed seem like errors in bit-reading come from spatio-temporal shifts and never from the misinterpretation of a bit's level.

*H3*'s first hypothesis concerning the motors' lower dead zones was verified, with very low speeds being reachable thanks to the closed-loop PID. However, the limited accuracy of the wheels' PID was far from being the main error source for bit-reading, given the PID indeed showed high accuracy at all hover speeds.

Lastly, *H4*'s hypotheses about temporal lags still stands, but whether or not this is the reason of failure at high speeds is debatable. While it could be one of the reasons of failure, it has also been shown that high errors in actual robot speed at high speeds also contribute to a fail in bit-reading.

## VI. Conclusion

An optimum bit size and hover speed was found for robotic barcode reading, at a hover speed of around 180 mm/s and a 12 mm bit size. The relationship between these two parameters is complex, with multiple linear and non-linear error sources giving rise to a quadratic evolution in bit sampling's precision (SPS).

Further temporal lag may come from the interrupts' desynchronisation due to the sensors' non-negligible sampling delay caused by the capacities' discharging time, which could be further explored. For instance, recording every loop duration would enable to track whether they undergo a certain deformation, which might lead to the observed quadratic rise in SPS.

## References

[1] S. Banker, "Robots in the warehouse: It's not just amazon," Jan 2016.
[2] A. Bolu and O. Korçak, "Path planning for multiple mobile robots in smart warehouse," in *2019 7th International Conference on Control, Mechatronics and Automation (ICCMA)*, pp. 144–150, 2019.
[3] P. R. Teja and A. A. N. Kumaar, "Qr code based path planning for warehouse management robot," in *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 1239–1244, 2018.
[4] "Pololu - 3pi+ 32u4 robot - standard edition (30:1 mp motors), assembled."
[5] "Atmega16u4/atmega32u4 - microchip technology."
[6] P. O'Dowd, "Ematm00532122: Pololu3pi labsheets," Nov 2021.