Ground Barcode Reading for Warehouse Robotic Systems

Cyril Monette - 2179734
Department of Engineering Mathematics
University of Bristol

Massittha Maskasem - 2185253
Department of Engineering Mathematics
University of Bristol

***Abstract*—It is normal to write the abstract last. First, try to concisely state the problem/question/challenge under investigation. Second, state what you found out. The abstract should help a reader decide if they are interested in reading the whole paper.**

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

Really good. :)

I. Introduction

Motivated by higher efficiency and the removal of all human errors, the mainstream use of robots in warehouses ~~came way sooner~~ arrived earlier than anybody expected [1]. With currently high numbers of emerging competitors besides the well-known Amazon robot (ex-Kiva robot), ~~this is sign~~ that the smart warehouse field is not even yet at its peak. Put this at start to make clearer

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 opt for the reading of information ~~under~~ in the form of a single bar-code 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 bar-code is not done with a bar-code 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 lengths of code. It is as of yet unclear what the range of speeds and information density are for data to be successfully picked up.

In this optic, the present research paper makes use of a single reflectance sensor on Polulu's 3π+ robot to quantify what the limits of a linear black and white bar-code reading operation are in terms of reading speed and physical bit width. For this purpose, the robot hovers a bar-code repeatedly alternating black and white bits at various constant speeds and sampling the bit below at a constant sampling frequency. above a?

A. Hypothesis Statement

1) The first hypothesis that is made concerning the bit-reading of the robot is that none of the potential reading errors come from the ground sensors misinterpreting a bit. Instead, all reading errors are due to 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.

2) Furthermore, it is expected to find the same results for an increased bit size as for a reduced robot speed, as they both should lead to better reading performances and are intimately linked to the sampling frequency at which the 3π+ is reading the code. This link between the temporal and spatial parameters is something that will be further explored as it constitutes a central hypothesis to make code-reading successful.

3) Because the wheels of the 3π+ can be driven with a PID, the motors' lower power dead zones should not be a limit for minimum speed as the closed-loop PID relays on actual motor rotation measured by external encoders. Instead, at lower and medium speeds, the wheels' PID is expected to be the limiting factor in delivering wheel speeds with low precision with regards to the target speeds. This error in effective robot speed then induces biases in the sampling frequency and leads to low efficiency code-reading.

4) If however the robot's speed is high, parasitic temporal lags translate into important parasitic spatial shifts which eventually becomes 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 on the 3π+ mobile robot, running several tests of code-reading at different velocities and for various bit sizes.

II. Experiment Methodology

A. Overview of Method

The experiment is divided into three phases: calibration, line following, and barcode reading.

In the first phase, the calibration of the robot's reflectance sensors is conducted to alleviate effects of ambient light on

Hypotheses are good, but I feel could be presented in a more sequential manner. 2. seems like a good first hypothesis - you could then go on to describe ones relating to errors etc...

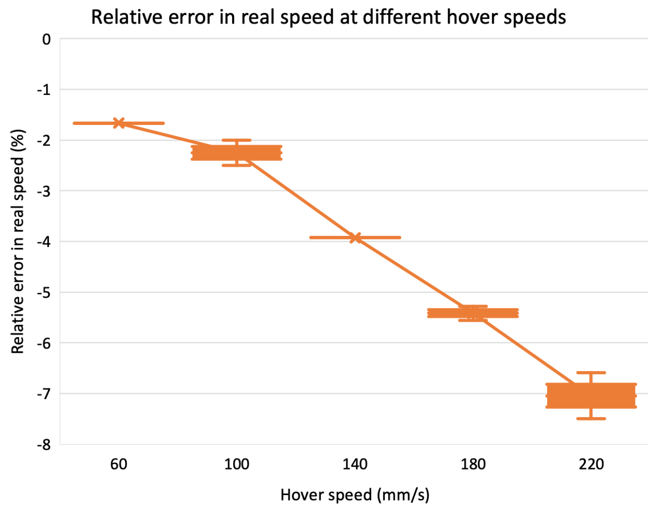


Fig. 1: TO BE REPLACED by image of: Description of the overall experimental setup

the sensors. During this phase, the robot runs on alternating black and white surfaces, allowing it to record measurements for the darkest and brightest surfaces to calculate calibration coefficients that enable a normalised format for every future measurement. 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 tape at a set distance to allow it 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 any human noise related to the robot placement.

Next comes the barcode reading phase, where the robot moves over a barcode alternating between black and white strips (bits) of equal widths and samples the ground's reflectance at a fixed frequency and distinguishes 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. To assess accuracy, the robot is programmed to expect an alternation of brightness and stops whenever it reads the same values in two consecutive readings, assessing his success automatically. Having the specific pattern of alternating bright and dark bits ensures no one bit can be read wrong yet still return the right value, as bits' neighbours are of different brightness for every bit.

Feels like this section might be better somewhere here!

Its

repeating

incorrectly

The experiment was done on barcodes of four different bit sizes: 4, 8, 12 and 16 millimetres. Each bit size has was tested using five speeds: 60, 100, 140, 180, and 220 millimetres per second. Four repetitive trials were conducted for each speed and bit size, which adds up to a total of eighty trials.

Lastly, the data-retrieving phase takes place when the robot records two consecutive bits of the same value, meaning that it has failed to recognise the correct bit colour. The robot then stops moving and the data can be retrieved once it is plugged into a computer.

Heading of robot? With the initial line-following?

B. Discussion of Variables

- **Controlled Variables**: Bit width and reading speeds, light conditions, batteries
  The sizes of the bits are fixed by the ladders which are printed. The control of the reading speeds will be explained in the following section. Every trial is carried out under the same ambient light. Batteries used are rechargeable and are fully charged before performing the experiment.
- **Independent Variable**: Bit width and reading speed
- **Dependent Variable(s)**: Number of bits read correctly, time of reading

C. Discussion of Metric(s)

Because the number~~s~~ of bits that can correctly be read determine~~s~~ the complexity of ~~an~~ the information that can be transmitted in real conditions, this will constitute the main metric~~s~~ evaluated during the experiments.

It is desirable to observe to what extent the robot can pick up information from the barcode accurately and precisely. While the accuracy is rated by the number of bits of information that can be read correctly, the precision can be assessed when looking at the position within each bit at which the readings are taken (Figure 2). A score can be derived from that position, where 0,5 is the ideal central sampling position, values close to 1 translate a late sampling, close to the next bit's edge and scores close to 0 mean an early sampling, close to the previous bit's edge.

don't think you need this

be consistent: In the graph you use a point

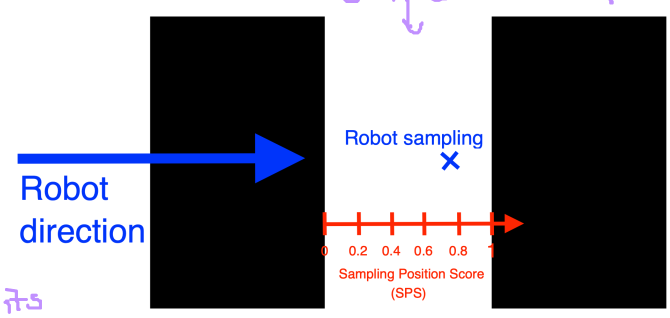


Fig. 2: Sampling Position Score (SPS) within a bit — In this example the robot samples later on with an SPS of ---.

describe a bit more fully so you don't have to spell all the info out the main

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 3π+ robot [4] from the brand Polulu, equipped with an ATmega32U4 microcontroller [5].

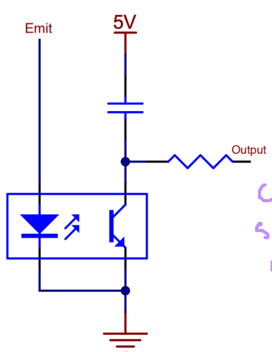
1) Reflectance sensors: The $3\pi+$ robot has 5 reflectance sensors orientated towards the ground, amongst which the three central ones will be used for line-following and the unique central sensor for barcode reading. The functioning of these is relatively simple and can be understood from Figure 3. An infrared Organic LED (IR OLED) shines the bottom surface beneath 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 described in Section II, 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.

If you are tight on space this isn't a key figure



is this the same as calibration? If so be consistent with terms

Fig. 3: Reflectance sensors' electrical layout

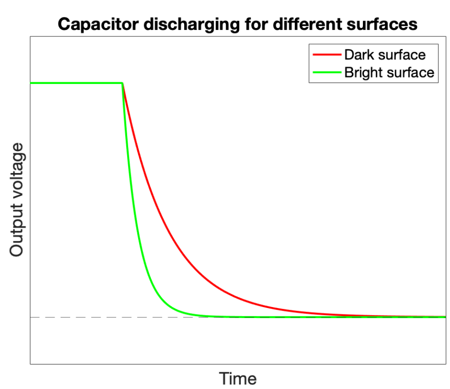


Fig. 4: Reflectance sensors' output voltage over time for different surfaces

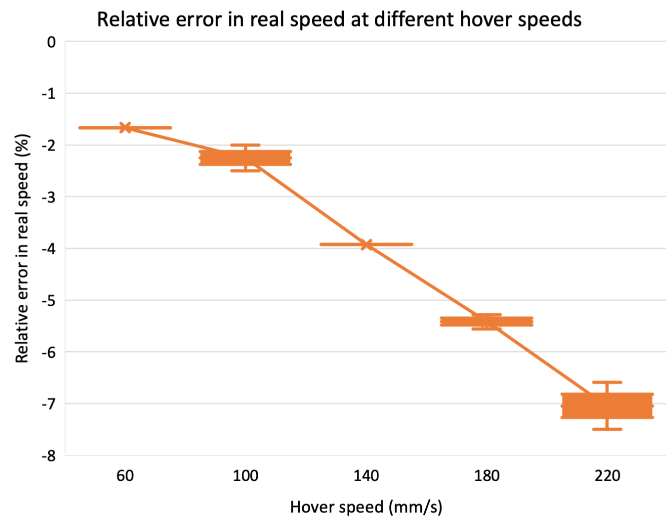


Fig. 5: Robot's real speed is increasingly lower than demanded speed

2) Intrinsic speed characterisation: to fill in

B. System architecture

As previously discussed in Section II, the robot will undergo different phases during the experimentation, in the following order: Initialisation, line-following, code-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-Control is implemented using a PID controller on the three central line sensors, controlling the heading direction of the robot following, and a PID controller on each wheel, monitoring the wheels' rotational speed. 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 command when to sample a bit's brightness, 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 counter at half of its maximum count by the end of line following phase. Only the central front reflectance sensor of the robot is used for bar-code reading.

In the barcode reading phase, PID controllers for both heading direction and wheel speeds continue working 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 this isn't very clear

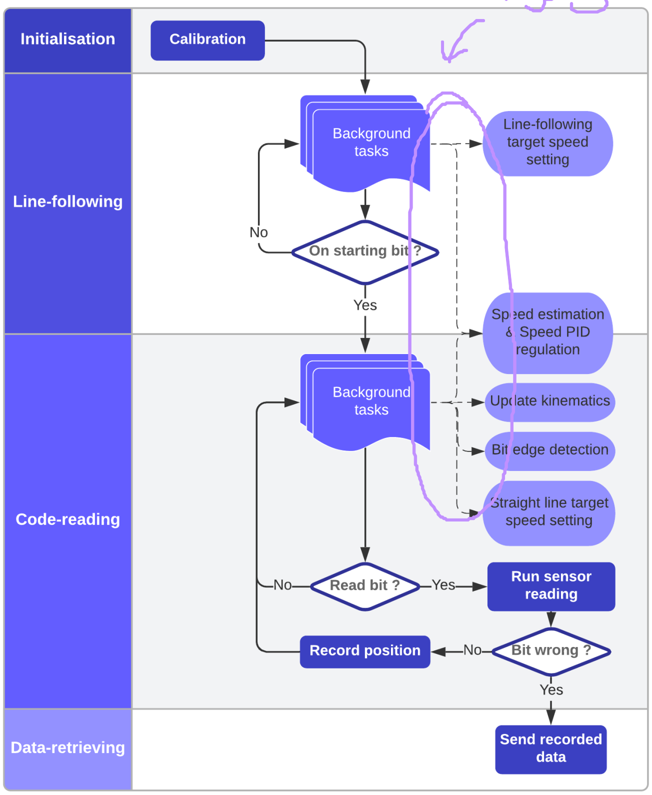


Fig. 6: Software structure

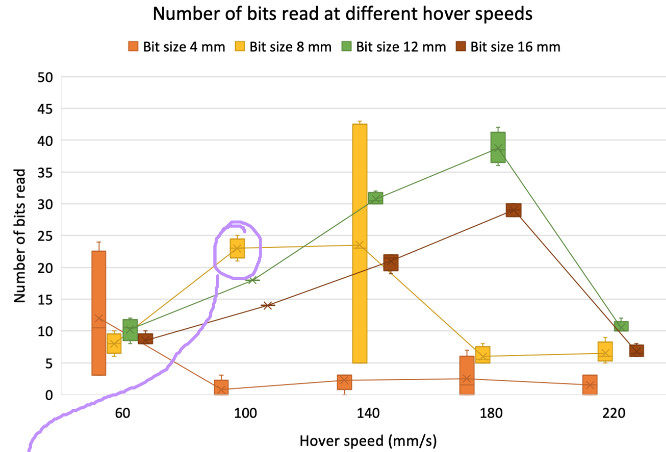


Fig. 7: Number of bits read at different hover speeds

Pick a couple examples from the graph and explain what it is showing

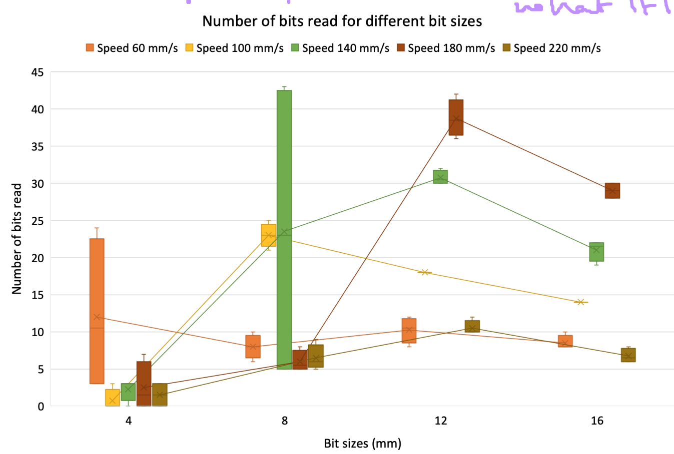


Fig. 8: Number of bits read for different bit sizes

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 ??) with the following formula:

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

IV. Results

See comments in email

The results obtained from the experiments are presented below, with some interesting unexpected speed and bit width maximums: In Figure 7 and Figure 8, we can observe the following things:

1) Low speeds and small bit sizes lead to poor bit reading.
2) There exists an optimal hover speed and bit size, respectively around 180 mm/s and 12 mm.
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-A.

A. Sampling Position Score 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 following insightful graphs. As a reminder, a SPS approaching 1 or 0 leads to a failure of the bit-reading as this corresponds to trying to read the previous bit ($SPS \leq 0$) or next bit($SPS \geq 1$). These graphs help us conclude on the following things:

space

approximates?

1) The SPS ~~describing~~ a parabola over the course of the run, this indicates that there are error sources both pushing the SPS towards 0 - leading to either temporally fast or spatially slow robot behaviours - and towards 1, which translate into 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 great uncertainty of the 12 mm bit size and 140 mm/s hover speed runs comes from the fact that the grey parabola described in Figure 10 has a minimum close to 0 around bit 5, hence sometimes leading to only 5 bits read and sometimes all the way to 40
3)

8?

you should explain what that means here, otherwise it doesn't really make sense

B. 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 explain the non-linearity in Figures 8 and 7.

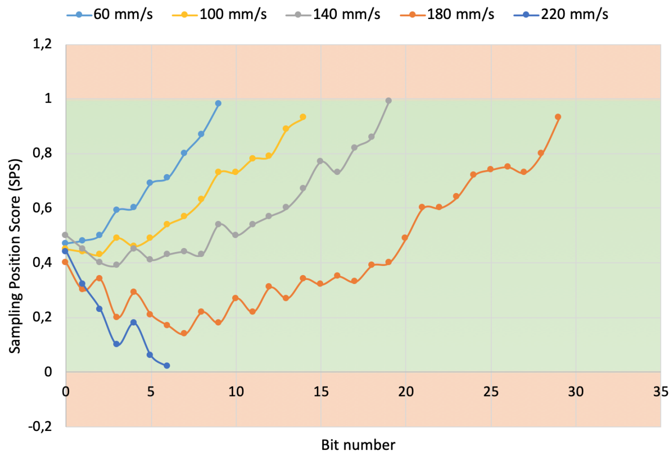


Fig. 9: SPS over runs at different hover speeds but constant bit size (16 mm)

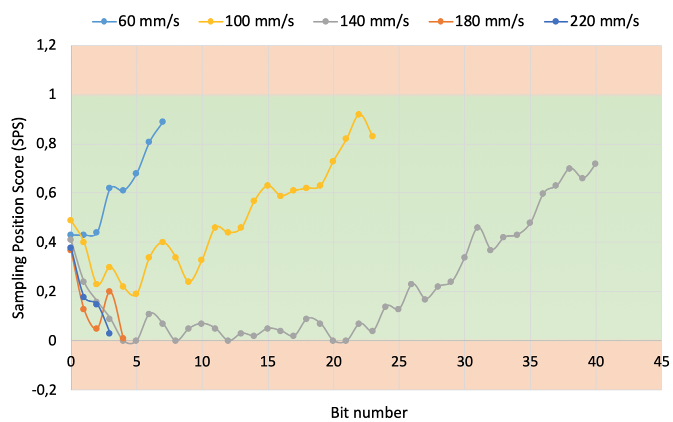


Fig. 10: SPS over runs at different hover speeds but constant bit size (8 mm)

The first error source was already previously detailed in Figure 5 and describes a shift in the robot's real speed compared to what is asked from ~~him~~ it. Because this shift is negative, meaning the robot is spatially slower than ~~what is~~ expected, it is responsible for a decrease in the SPS. This can be observed in Figure 9 and 10, specifically at high speeds where this type of error is prevailing, as Figure 5 indeed confirms.

which ones specifically in each graph

While the speed error can be responsible for a decrease in SPS, the origin of the SPS's increase is yet to be found. 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, 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 $\frac{W}{2}$:

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

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

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

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

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^*:

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

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

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

Because $\alpha = \frac{\Delta W}{W}$ is a constant, n_{error} is constant over all speeds and bit sizes, and is also plotted in Figure 11 in brown.

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 number of successful bit-readings like what was obtained during the experiments, plotted in Figure 7. However, the parabolic nature of the SPS observed in Figure 9 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.

V. Discussion and Conclusion

Here we discuss the initial hypotheses (listed below) and the directions to focus future research in order to better characterise a robot's interactions with its environment in a barcode reading operation.

1) The first hypothesis that is made concerning the bit-reading of the robot is that none of the potential reading errors come from the ground sensors misinterpreting

You don't need to repeat everything word for word here

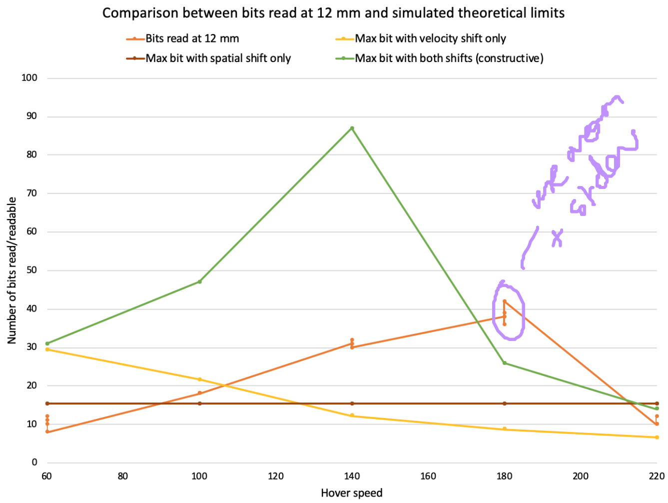


Fig. 11: Comparison of simulated maximum readable bits with spatial and temporal lags with real 12-mm bit width results

a bit. Instead, all reading errors are due to 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.

2) Furthermore, it is expected to find the same results for an increased bit size as for a reduced robot speed, as they both should lead to better reading performances and are intimately linked to the sampling frequency at which the 3π+ is reading the code. This link between the temporal and spatial parameters is something that will be further explored as it constitutes a central hypothesis to make code-reading successful.

3) Because the wheels of the 3π+ can be driven with a PID, the motors' lower power dead zones should not be a limit for minimum speed as the closed-loop PID relays on actual motor rotation measured by external encoders. Instead, at lower and medium speeds, the wheels' PID is expected to be the limiting factor in delivering wheel speeds with low precision with regards to the target speeds. This error in effective robot speed then induces biases in the sampling frequency and leads to low efficiency code-reading.

4) If however the robot's speed is high, parasitic temporal lags translate into important parasitic spatial shifts which eventually becomes the limiting factor for successful code-reading. It is thus hypothesised that high reading speeds result in much less reliable code-reading.

Deformation of a loop time at high speed may lead to parabolas Should record every loop duration to confirm this

We found an optimum bit size and hover speed nonetheless Constant time Second degree motion that accelerates the robot, because spatially unsynchronised Further temporal lag may come from the interrupts and the sampling delay, which could be further explored

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."