Component limits – Evaluations on sensor reliability and overall accuracy

Samuel Vergeiner, *Team "Wuggeis Buam"*

Abstract

The scope of this paper is to evaluate the capacities of the sensors which are contained in the Botball kit. In addition to that the engines were tested and strained to their maximum. The conditions for the measurements were kept continuous throughout the entire process to ensure their reproducibility.

INTRODUCTION

In the first chapter we take a close look at the electric engines. In this chapter ways to run the engines are shown and described. Additionally, their maximum speed and momentum are evaluated. In the second chapter the servo-engines were tested. We determined their maximum modulation. In the third chapter the sensors were examined. We evaluated their maximum potential and tried to show some general problems with the sensors.

1. ENGINES

Each set contains five electric engines which can be run through different methods. These methods are discussed below.

All the following measurements were made with fully charged batteries, as this impacts results significantly. For measuring we used a "PAN Multitacho Drehzahlmesser" as shown below.

Variant 1:

motor(#port, % velocity);

With this command the engine runs with the given percentage of the maximum velocity. Therefore, the maximum value which can be given this way is 100. In the following test we tried to obtain the maximum speed of the engines without load. The values are merely means, as the results fluctuated strongly.

A simple program was used to measure the following:

Experiment at maximum speed (1000ticks per second):

Engine 1:	67.6 1/min
Engine 2:	65.0 1/min
Engine 3:	67.5 1/min
Engine 4:	66.4 1/min
Engine 5:	65.9 1/min

The values fluctuated within a range of 6 to 8 1/min.

Standard deviation: 0.98

Variant 2:

motor_power(#port, % power);

In contrast to variant 1 instead of the velocity the percentage of the maximum power is given. The engines cannot, of course, be run at more than 100%. Even operation at 100% or little below that value is not recommended as this could lead to dangerous overheating of the components form which irreversible damages can occur.

Experiment at maximum speed:
Engine 1: 69.2 1/min
Engine 2: 67.4 1/min
Engine 3: 68.4 1/min
Engine 4: 69.3 1/min
Engine 5: 68.4 1/min

The values fluctuated within a range of 4 to 5 1/min.

Standard deviation: 0.69 U/min

As can be seen above, different results followed the different ways of running the engines.

In the next experiment the maximum momentum was evaluated. As it is very likely that the engine that is strained could be destroyed we limited our testing to only one engine.

To determine the maximum momentum the engine was strained with growing momentums through applying weights at a persistent distance from the axis until the engine stopped revolving entirely.

The momentum measured was: 370Nmm.

Variant 3:

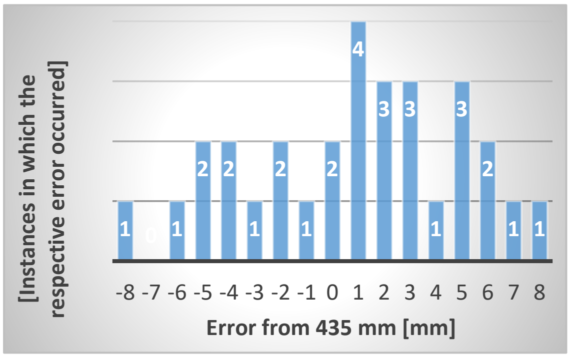
mav(#port, #velocity);

With this command the engine revolves with the velocity given in “ticks per second”.

Variant 4:

mrp(#port, #velocity, #position);

With this command the engine revolves a defined amount of ticks, which are given as the third value “#position”. This method is not a very reliant one. Our experiments have shown



that the distance the engine travels fluctuates. The diagram below is a result of the documentation that had to be written in the course of the Botball competition. It shows the fluctuations the travel of the engines has at a certain distance.

1.1 Conclusion

All in all there are many things to say. Enormous differences in the achieved revolutions per second via all different methods show that. If the revolutions per time are measured over a longer period of time (10 minutes) the measured value converges to a definite value, however the values still fluctuate significantly. In addition, the method which is applied to run the engines impacts the results very strongly.

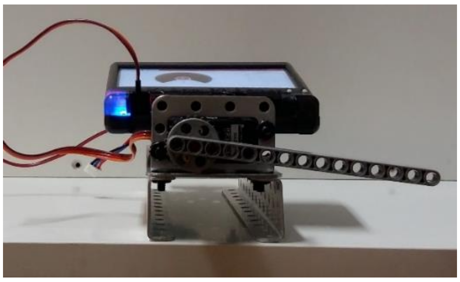
2. Servo-engines

The servo-engines work similarily to the conventional engines. In addition to the power supply another port is needed to transmit the signal for the position.

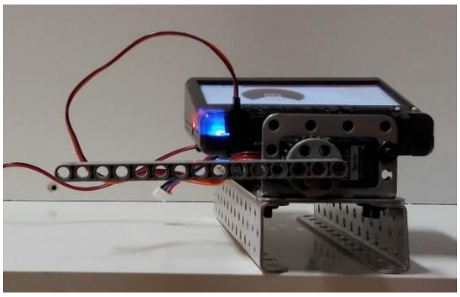
The port has to be connected with the orange cord oriented towards the screen of the controller. Using this orientation the servo-engine rotates counter-clockwise. If the cord is oriented differently the servo-engine does not work at all.

The transmission of the signal works via a 10bit lane, which translates to 2048 different position for the engine.

The different achievable positions are shown below.
Position 0:



Position: 2047:

 Thus, the servo-engine is capable of half a rotation with 2048 different steps.

The small servo-engine works basically the same way.

The commands to run the servo-engines are straightforward. The following command moves the servo-engine to the given position.

enable_servos();

This command activates the servo-ports. The signals are transmitted to the actors.

set_servo_position(#port, #position);

With this command the servo-engine on the specified port moves to the given position.

disable_servos();

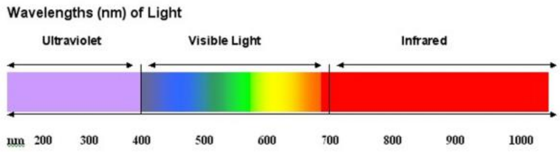
This commands deactivates the servo-ports. Commands following this command are not transmitted to the actors, but are still acknowledged by the compiler.

3. Sensors

This kit contains a lot of sensors which enable the robot to react to its environment. These sensors can for example be used for the robot to follow a coloured line on the floor or to dodge an obstacle.

4.1 Light sensor

The light sensor is an analog sensor. This sensor is used to start the robot at the beginning of a game. To achieve that, the robot is irradiated by a light. The sensor reacts only to light with a wavelength of 780nm to 1mm, hence to light in the infrared spectrum.



For this reason it is required to use incandescent light, as LEDs do not emit light in this spectrum.

wait_for_light();

This command stops the program until light is registered by the sensor.

4.2 Reflectance sensor

This sensor is, again, an analog sensor. It emits light in the infrared spectrum and measures the amount that is reflected back at it. The amount of the light measured is depended upon many factors such as surface texture, colour and the distance to the object.

If the sensor is mounted at a certain distance relative to the ground, it is easily possible to distinguish a black area from a white area. This function can be used to follow a line on the game table.

However, this sensor is subject to some restrictions. The distance to the measured object is a decisive variable for reproducible results. Furthermore, light plays a very dominant role. The following experiments were conducted to evaluate the sensor's reaction to those cross sensitivities.

The tests were carried out in a distance range of 5 to 50mm and provided us with the following conclusions:

At distances below 15mm no distinction between different colours was possible, even white and black which should produce very different values were not distinguishable.

The best results were made at a distance of about 20mm. Even distinction between different shades of colours such as light blue and dark blue were possible. As a consequence we can conclude that 20mm is the best distance to mount this sensor at.

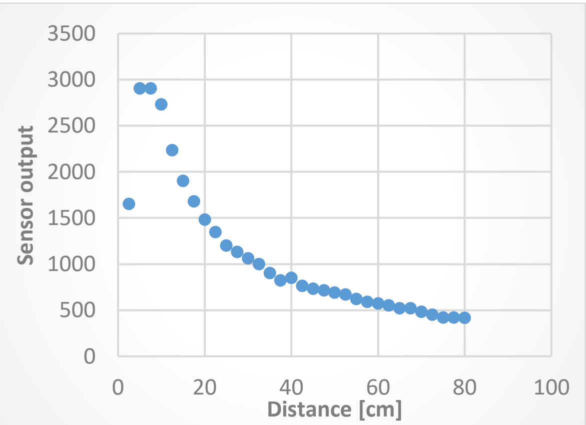
If the distance is now enlarged further, it gets progressively more difficult to differentiate different colours. However, a distinction between black and white is possible up to a distance of 50mm. Increasing the distance further beyond the 50mm mark leads to indistinguishable values for each colour and makes the overall use of this sensor redundant.

At last, we researched the cross sensitivity of the sensor to different light sources. As it was proposed earlier, light emitted by LEDs does not impact the measurement at all. If a light source such as a standard light bulb is used, the results become useless.

4.3 Distance sensor

The distance sensor is also an analog sensor. It emits infrared light, detects the angle at which it is reflected back and calculates the distance through triangulation. The range within the sensor works reliable is given as 5cm to 80cm. In our experiments we tried to confirm those values.

As seen in the diagram below, the sensor is indeed only reliant within the given distance range. At distances smaller than 5cm the measured values are not coherent with the distance; at distances greater than 80cm the sensor does sometimes output values from which the distance can actually be concluded, but errors are very common.



4.4 Camera

Two different cameras are contained in the set. All of the following measurements and assumptions are based on the Hercules model. The camera can be used to capture images and detect different colours in these images. The camera is also capable of measuring distances within the picture. This can be used to track objects with the pre-set colours and evaluate the robot's movements in accordance to the position of those objects.

The colour the camera tracks can be directly set in the controllers menu. The to be measured object is placed in front of the camera and selected via touchscreen. Four different colours can be tracked this way.

The following is a small code that gives the x- and y-coordinates of an object with the colour that has been set on the channel ”0”. If there were more objects of the same colour in the picture, the camera would calculate the distances based on the object that is nearest to the camera. Hence, the second ”0”.

```
Int x;
Int y;

while(1);
{
camera_open();
camera_update();
x = get_object_center_x(0, 0);
y = get_object_center_y(0, 0) ;
}
```

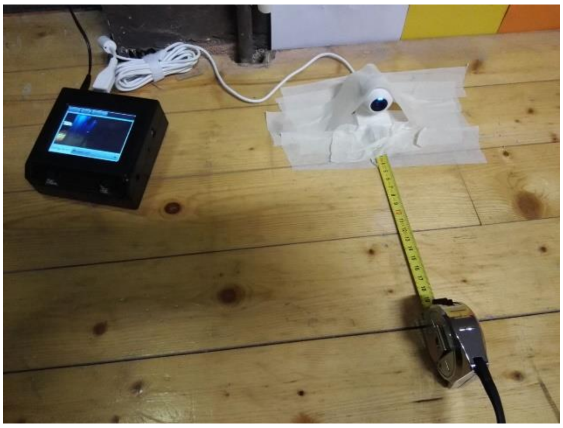
The screen is divided into 160 units on each axis; with this knowledge in mind the robot can be programmed to drive in the direction of an object with a specified colour.

With the following command the amount of tracked objects in a picture can be obtained.

```
get_object_count(#colour #id);
```

Colour tracking works very reliantly with both cameras. In the following two tests we tried to push the camera as far as possible and determined the maximum distance for colour-tracking as well as the reaction time of the camera.

In the first experiment we tried to evaluate the maximum distance the camera can detect objects. With the simple setup shown below we measured the maximum distance the camera is able to detect an object.



The objects used consistently had an area of 15cm by 7cm. We used different colours in a setting of artificial light provided by neon lights.

The results we got were quite steady. The camera works reliantly within a distance range from 0m to 9m, but fails almost every time the 9.5m mark is crossed. An additional conclusion we were able to make was that the amount of objects in the scope of the camera does not at all impact the results.

5. FINAL WORDS

All in all, it can be said, that the sensors are equipped with enough functions to enable a robot which is structured around these sensors to be universally useable in a variety of applications.