The use of closed-loop control systems in Botball

Software Engineering

Adrian Pandjaitan, adrian.pandjaitan@gmail.com
Team Talentehaus
St. Pölten, Austria

***Abstract*—In robotic applications control theory plays an important role.This paper provides an insight in the history of closed-loop controls and defines the differences between them and open-loop controls. The most common types of controllers ranging from P-only elements to the more complex PID controllers are also described. Finally this paper shows possible use cases by implementing controls in our line following robot for the PRIA Open tournament.**

I. Introduction

The game board of the Botball tournament offers various assistances for robots to determine their position and to navigate on the game field. Robots which do not solely rely on their internal sensors like encoders or gyroscopes yield better results in general, as they are more flexible and less error-prone. For example instead of hard coding the way the robots will drive, the black and blue lines on the ground provide simple reference points in order to create a basic line follower. However for implementing this strategy, the use of closed-loop control systems like PID controllers is inevitable. As those systems can respond to outside changes, like the bot losing the line to follow, they are far superior to simple open-loop controllers which do not possess this feature. [1]

II. Development and state of the art

Closed-loop controls are not a new or solely technical phenomenon. This can be seen at warm-blooded animals for example, which manage to maintain a constant body temperature regardless of the ambient temperature.

In technical applications, closed-loop systems have already been used by the ancient Greeks, who developed containers which could keep liquid on a certain level constantly. However as this system was mainly used in water clocks and oil lamps to improve the lighting quality, it was mostly viewed as unnecessary luxurious and soon started to be forgotten. [2]

A. Industrial revolution

It was not until the industrial revolution that this principle of self-regulatory control systems was taken on again. When the Frenchman Denis Pupain designed the pressure cooker he encountered the problem of exploding pots due to too high pressure. By implementing a pressure valve though, he was able to keep the pressure on a sustainable level and therefore invented a controller which is still in use today. With the emergence of steam engines the concept was further developed. To keep the revolution speed of such a machine constant, even under load, centrifugal governors were used. They were made up of two metal spheres which spun around a spindle and were pulled out because of the centrifugal force. Over a special mechanical mechanism the spheres would then close the air inlet valves more and more so that the engine speed would stay on a safe level. When a load was added the rotation speed would drop quickly and therefore the throttle valves would be opened again. The engine would then power up until the point where the air inlets would be closed again resulting in the originally wanted engine speed. [2]

B. Use cases today

In automation engineering closed-loop control systems play a crucial role. Popular examples include closed-loop stepper motors, which need to prevent step loss so as to provide high repeatability. But the use of these control systems is not restricted to robots or machines. Even simple home automation systems, like ambient temperature control use closed-loop controllers. However, the main focus nowadays mostly moved away from physical feedback controllers to make way for electric and digital controllers. [3]

In modern day applications, closed-loop control systems consist of a controlled variable, which is the parameter the controller should keep at a certain value. This level is defined through an index value. Therefore the controller has to constantly monitor changes of the control variable. If it notices an offset it will then generate a corrective variable so as to counteract this deviation. [4]

III. Concept

A. Distinction between closed-loop and open-loop systems

Both open-loop controllers and closed-loop controllers are part of automation engineering. However, there are many important differences to take into account when deciding on one method.

1) Open-loop control systems: The goal of open-loop control systems is to change a certain parameter. For example, this can be a pressure valve, which can be opened and closed or the status of a light bulb, which can be switched on and off. Open-loop controllers may receive feedback about the condition of the observed parameter. However, in contrast to closed-loop controls, this response is not used to influence the parameter directly but rather to switch the system off in case of unsafe or dangerous states. In the case of a motor, this could happen if it is turning too fast. After the open-loop controller has been informed about this issue, it will switch the motor off, to be regulated by itself and the laws of physics. [5]

Popular examples for open-loop controls include simple radiators. Normally they have a ruler which manipulates the power output. Nevertheless, a desired temperature cannot be defined, hence the heating element will theoretically heat up infinitely.o

2) Closed-loop control systems: On the other hand, closed-loop controllers offer much improved control over the regulated parameter. To keep the value constantly at the desired level, a closed-loop controller has to permanently monitor the actual state of the system and compare it to the index value. Therefore these controllers have some kind of sensors or other, also physical feedback methods, like a centrifugal governor or pressure valve. If the desired value and the controlled value differ too much, the controller will take measures to approach the index value. The strategies by which this can be done vary greatly. Common examples include proportional controllers, integral action controllers and combined PID controllers. [5]

To visualize such a control loop, so called block diagrams can be used. As the most basic control loops only use very few variables and functions, it is easily possible to create such a scheme. As can be seen from figure 1, a simple control loop

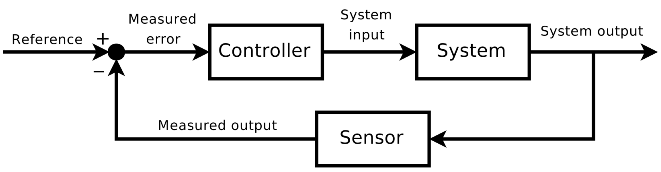

Fig. 1. Control loop block diagram

has a sensor, which measures the output of the system. This output is compared to the reference value and thus the error is calculated. The controller then generates an input in order to make the system approach the desired value. This loop repeats until the reference value is reached.

An example for this type of control systems would be a temperature-controlled heating system. Alongside simple radiators it consists of a thermometer and a controller. It can be given a specific temperature to maintain. The thermometer will then measure the temperature and report it to the controller. If this element comes to the conclusion that the temperature is either too high or too low, it will switch off or turn on some radiators. After a small period of time, the desired temperature should be reached and the controller may output nothing until the temperature deviates enough again.

In robotic applications closed-loop solutions are often superior and sometimes even the only feasible solution to a problem. For instance a line following robot should always be implemented by using a PID controller, as this control system provides the necessary features so as to reliably stay on the line and not overshoot it too much, when correcting its movements.

B. Different types of closed-loop controllers

In this section the most widely spread control strategies, which are useful for Botball are described.

1) Proportional controller: A proportional controller, also known as P controller describes a linear dependency between the controlled variable and the corrective variable. It provides instant output, which is the offset multiplied by the amplification factor K_p. Disadvantages of P controllers are that if the deviation of the desired value is rather small, the output of the controller is also small. However the desired value can never be reached as while it is approached the error and thus the controller's output get consistently smaller. At some point the error is so small, that even multiplied with the factor K_p, it cannot be forced down to zero. However it is still superior to an on-off control which basically just switches a system on and off if the controlled variable falls under or over the desired level. Mathematically the P controller can be represented by the equation $Output(t) = Error(t) * K_p$. [6]

2) Integral action controller: This kind of controller, which is also called I controller, adds up all deviations over the whole time period by calculating the integral of those values. It then multiplies it with the factor $\frac{1}{T_R}$ or K_i. The formula by which it can be represented is $Output(t) = \frac{1}{T_R}\int_0^t \mathrm{Error(t)}\,\mathrm{d}t$, where T_R is the reset time, i.e. a factor which determines how heavily the error over time effects the output. Integral action controls are able to eliminate the offset completely, however their disadvantages lie in slower speeds and overshooting, which can be seen in the following figure. [7]

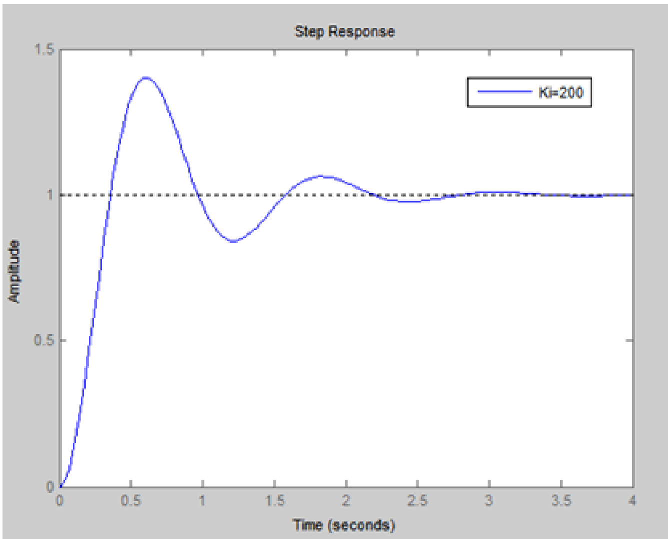

Fig. 2. Integral action controller overshoot

3) Derivative controller: D controllers are used to extrapolate how the error will evolve in the future. The time which is taken into account is defined by the variable T_v. Used correctly, the D controller has a dampening effect. However, if a very noisy variable, which is constantly undergoing small and rapid changes, is controlled, the D element amplifies this noise. It can mathematically be represented by $Output(t) = T_v \frac{\mathrm{d}}{\mathrm{d}t} Error(t)$ [8]

4) PI controller: Proportional integral action controllers combine the features of a P and an I controller. Due to the influence of the P controller they are faster than traditional I controllers, but equally as precise. [9]

5) PD controller: Proportional derivative controllers are composed of a P controller and a D element. With the D element the controller calculates the rate of change of the deviation, which is then added to the output of the P element. These controls are generally very fast, however the drawback of the P controller of never reaching the index value still exists. [10]

6) PID controller: PID controllers combine all the just described features in order to create a universally usable control system. Therefore it is also one of the most widely spread controllers. By adjusting the P, I, and D elements it is possible to create a control system which possesses all the advantages of the single elements while cutting out the disadvantages. Its equation consists of the combined equations of all other controllers $Output(t) = Error(t) * K_p + \frac{1}{T_R} \int_0^t \mathrm{Error(t)} \,\mathrm{d}t + T_v \frac{\mathrm{d}}{\mathrm{d}t} Error(t)$ [4]

IV. Implementing controllers in software

As controls normally can be expressed through equations with only a few variables (a P element for example only needs the controlled variable, the measured variable, an actuating variable as output and the factor Kp by which it should be multiplied), its implementation in a program is not very difficult. For the more complex controllers there are even libraries for most programming languages available such as ivPID, which is a small open-source Pyhton library for creating PID controllers, created by the Turkish company IVMECH MEKATRONİK. [11]

A. Deciding on a particular control strategy

When deciding on which controller to use, there are several factors to take into account. While PID controllers are suitable for almost every purpose, sometimes the may be unnecessarily complex. In many cases the simpler P controllers suffice, especially in cases where the desired value has not to be matched completely. If precision is needed, but speed is not important, then an integrated action controller would already be enough. Therefore, when designing a control loop the actual controller should be chosen carefully.

B. Example of a simple line follower for the Open tournament

For the open tournament was a robot which heavily relies on line following created. Therefore it had implemented a system which emulates a closed-loop controller into the software. For this purpose we decided to settle on a PID controller as it was best suited for our use because of the benefits mentioned above.

We then went on to create a program for the Wallaby which implements these features. As sensors we used two color sensors to provide feedback to the controller in order to distinguish between the white floor and the black lines. The output of the Wallaby is used to adjust the rotation speed of the two motors we use for the wheels. This means that the actuating variable is the signal to the motors. The controlled variable however is not the revolution speed of the motors, but rather the deviation of the bot from the black line. Therefore the bot will start to turn in a small curve when an obstacle forces it to go off the line. Due to the integral action part of the control, if the line is not found instantly, the curve will start to get bigger because the offset is summed up. Eventually the bot will turn back to the line. However, as the bot will come in at a high angle, the bot will overshoot the line. The integral action element then will start to add up the offset for the other side, but this time the deviation will be a little bit smaller, hence the bot will reach the black line faster. This will happen a few more times and eventually the bot will drive on the line again perfectly.

V. Conclusion

There are many different types of control systems. Ranging from a simple proportional element to the much more complex PID controller, each comes with its advantages and drawbacks. The example outlines that the use of closed-loop controls makes sense almost always, even on a small scale like our robot. Even though it would be possible to use a less intricate open-loop controller by just telling the robot to follow the black line if it has got one under it, our solution is much more resistant against unforeseen obstacles which could lead to the robot to lose its way.

References

[1] *2018 Botball Game Review.* Website. Available at: http://webspace.pria.at/ecer2018/GameReview.pdf; Accessed: 16th March 2018.

[2] F.L. Lewis: *A brief history of feedback control.* Website. Available at: http://www.uta.edu/utari/acs/history.htm; Accessed: 16th March 2018.

[3] Circuit Specialists: *Advantages of a Closed-Loop Stepper Motor System.* Website. Available at: https://www.circuitspecialists.com/blog/advantages-of-a-closed-loop-stepper-motor-system/; Accessed: 16th March 2018.

[4] University of Michigan: *Introduction: PID Controller Design.* Website. Available at: http://ctms.engin.umich.edu/CTMS/index.php?example=Introduction&section=ControlPID; Accessed: 16th March 2018.

[5] Vance VanDoren: *Open- vs. closed-loop control.* Website. Available at: http://www.controleng.com/single-article/open-vs-closed-loop-control; Accessed: 16th March 2018.

[6] *Proportional-Only Control.* Website. Available at: http://iamechatronics.com/notes/general-engineering/443-proportional-only-control; Accessed: 16th March 2018.

[7] The Industrial Automation and Mechatronics Portal: *Integral (Reset) Control.* Website. Available at: http://iamechatronics.com/notes/general-engineering/445-integral-reset-control; Accessed: 17th March 2018.

[8] Jaques Smuts: *Derivative Control Explained.* Website. Available at: http://blog.opticontrols.com/archives/153; Accessed: 04th April 2018.

[9] Controlguru: *Integral Action and PI Control.* Website. Available at: https://controlguru.com/integral-action-and-pi-control/; Accessed: 17th March 2018.

[10] Matthew Kelly, Brad Saund: *PD Controller.* Website. Available at: http://www.matthewpeterkelly.com/tutorials/pdControl/index.html; Accessed: 16th March 2018.

[11] IVMECH MEKATRONİK: *ivPID.* Website. Available at: https://github.com/ivmech/ivPID; Accessed: 16th March 2018.

LIST OF FIGURES