Artificial Intelligence in robotics

Isaak Gruber
HTL St. Johann
Higher Technical Collage for Mechatronics
St. Johann i. Pongau, 5600, Austria
isaak.gruber@htl-saalfelden.at

Matthäus Miller-Aichholz
HTL St. Johann
Higher Technical Collage for Mechatronics
St. Johann i. Pongau, 5600, Austria
matthaeus.miller@htl-saalfelden.at

Joshua O'Dwyer
HTL St. Johann
Higher Technical Collage for Mechatronics
St. Johann i. Pongau, 5600, Austria
joshua.odwyer@htl-saalfelden.at

Thomas Prommegger
HTL St. Johann
Higher Technical Collage for Mechatronics
St. Johann i. Pongau, 5600, Austria
thomas.prommegger@htl-saalfelden.at

David Schett
HTL St. Johann
Higher Technical Collage for Mechatronics
St. Johann i. Pongau, 5600, Austria
david.schett@htl-saalfelden.at

Felix Schober
HTL St. Johann
Higher Technical Collage for Mechatronics
St. Johann i. Pongau, 5600, Austria
felix.schober@htl-saalfelden.at

Abstract - Artificial intelligence is increasingly being used in robotics these days. With the help of AI, systems with cameras and sensors can improve themselves without further software revisions and are also capable of recognizing patterns in processes. Furthermore, AI can and is already being used in the programming of processes. But could AI revolutionize this field of application in the future?

I. INTRODUCTION

As robots become more and more autonomous in the future, artificial intelligence (AI) will become more and more important in the programming of systems. With new interaction models, they are even able to generate text and voice responses that appear almost human, making them easier partners in everyday life. Machines can now recognize patterns in huge amounts of data, learn from experience, and visually understand their surroundings. But how does this work and where does it get the information from? Artificial intelligence is already widespread, is increasingly used in industry and will also become more common in everyday life. Using an AI can be a big advantage for us in the Botball competition, too. This paper will explain the principles underlying AI and where the information it uses comes from. We will also explain how artificial intelligence is constantly improving, and at the end, we will test this ourselves in an experiment.

II. HOW IS AI USED IN ROBOTICS

A. AI in programming robots

1) Genetic programming

In our paper we want to describe a method that an AI often uses depending on the application and which is called Genetic Programming.
Genetic programming (GP) is a method for automatically creating a functioning computer program for difficult problems. [4]

A population of randomly generated programs, which can also be seen as two random data sources, that have been optimized through evolution over multiple generations is the starting point for GP. [5] Natural selection serves as the foundation for the process, which states that the most effective programs endure and procreate. Mechanisms including gene duplication, crossover (combination of programs), and reproduction (copying programs) are all part of this process. [4]
Complex issues like symbolic regression and the automatic construction of electronic circuits can be resolved with GP. But of course, also in the field of robotics it can be used to detect problems in codes and modify them using the different operators. [5]

In robot programming, the computer program that computes the solution to some specific problem generally takes sensor values or system state variables as inputs and produces an external action as output. It is inefficient and unnatural to represent program hierarchies of general size and shape in terms of the fixed-size character strings found in conventional genetic Abbreviation and Acronyms. [1]

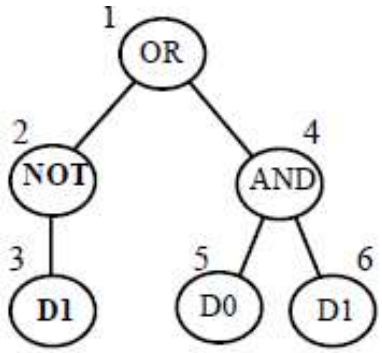
In genetic programming, a solution is sought by examining all possible combinations of existing functions. These functions can be computer programs, for example. The programs are gradually and repeatedly assembled from the available functions to find the best solution.
For example, the programming language LISP, with its symbol S-expressions, is a very practical choice to describe these functions of an artificially written program. [5] These S-expressions used in LISP is a direct mapping of the structure of the parse tree that is generated internally by at most compilers. [1]

2) Depiction of AI written programs

In genetic programming there are two central operations, the crossover- and reproduction operator:

- Reproduction copies next-generation programs based on their suitability. The more successful a program is, the more likely it is to be passed on.
- Crossover combines two pre-existing programs by randomly choosing parts to exchange. This creates new programs that adopt characteristics of both parent functions. This allows the AI to identify new solutions.

These processes allow for the gradual improvement of programs through natural selection and variation. [1]



[1] Fig. 1: Two parental computer programs shown as trees with ordered branches.

The figure above shows a syntax tree of a computer program used in genetic programming. The internal nodes represent functions (logical operators, in this case: OR, NOT and AND), while the leaves of the tree show the terminals (input data: D0 and D1).
Logical expression of the tree:

$$D1 \vee (D0 \wedge D1)$$

Trees are connected by the root OR: On the left, NOT negates the D1 input, while on the right, AND combines the D0 and D1 inputs. This results in the following function:

- The expression is true (1) if either D1 is false (0) or both D0 and D1 are true at the same time (1).

In genetic programming, this tree can be considered a program of the parent generation. Crossover allows the exchange of subtrees with other program trees. This optimizes older programs and also contributes to the creation of new programs. [1]

B. How an AI works

1) Data selection by AI

Now that we have clarified how AI is looking for ideal solutions with operators and the principle of GP, we would like to clarify where ChatGPT and other artificial intelligences get their data from, and we would also like to explain how AI is constantly learning and becoming more powerful the more you use it.

Artificial intelligences are experts at organizing and evaluating data. This can be done in a variety of ways. For example, through various internet sources, which the AI analyses and then provides an idea of a quantity, a unit, or a range, based, of course, on various criteria that the AI has already learned. [2]
But AI's such as ChatGPT also obtains its data from publicly accessible information databases. The advantage here is that the data is already structured and must not go throw different software like internet sources.[6]

In conclusion, however, it can be said that an AI is only as reliable as the sources it uses. Therefore, it is important to ensure that artificial intelligences have sufficient sources on specific topics. [6] These can then be combined by the GP operators to find the best solution or answer. [1]

2) How AI improves

Artificial intelligence can improve itself through machine learning. In this process, an algorithm learns how to solve a task by constantly repeating it. The software automatically memorizes data structures to learn them. For example, robots can learn how to grasp certain objects in order to transport them from one place to another.[7]
Users of artificial intelligence also improve their systems by pointing out errors and constantly repeating tasks.

In our case, machine learning is a crucial factor in the programming and development of our robots. By entering the correct parameters, we approximate the correct values, and ChatGPT implements the values as simply but accurately as possible. This approach makes our code coherent, and the AI also improves and explains errors and possible improvements. But the AI also makes mistakes. By constantly developing the AI and working with it, it continually improves. If it does something wrong, we point it out, and it does it better next time.

C. Programming with AI in Botball

Finally, we wanted to test an AI, in our case ChatGPT, to program a simple C program for our bot. To do this, we came up with a simple program in which our robot follows one of the black tape strips with an infrared Top-Hat sensor and stops as soon as it hits a pipe.

The following image shows the code written by us. The function of the program is the same as that of the code of the AI. It is also important to note that we did not base our programming on the artificially generated code.
When comparing the two programs, we noticed that both basically fulfil their described function and are structured similarly, but the artificially created code is much shorter. This has the advantage of saving memory.

```c
#include <kipr/wombat.h>
#include <stdio.h>

#define SMALL_TOPHAT_LEFT 0
#define SMALL_TOPHAT_RIGHT 1

#define LARGE_TOUCH_SENSOR 0

#define MOTOR_RIGHT 0
#define MOTOR_LEFT 1

void DRIVE_FORWARD();
void Follow_line();

int main() {
    while (1 == 1) {
        Follow_line();

        if(digital(LARGE_TOUCH_SENSOR) == 1)    {
            ao();
            break;
        }
    }
    return 0;
}

void DRIVE_FORWARD() {
    motor(MOTOR_RIGHT, 100);
    motor(MOTOR_LEFT, 100);
}

void Follow_line() {
    DRIVE_FORWARD();

    if (analog(SMALL_TOPHAT_LEFT) < 3800) {
        motor(MOTOR_RIGHT, 70);
        motor(MOTOR_LEFT, 100);
        msleep(200);
    }
    else if (analog(SMALL_TOPHAT_RIGHT) < 3800) {
        motor(MOTOR_RIGHT, 100);
        motor(MOTOR_LEFT, 70);
        msleep(200);
    }
    else {
        msleep(100);
    }
}
```

Fig.: 1 code written by us

With the program written by AI, the problem was that the AI didn't know the light conditions in the room. The AI set the sensor value for detecting the lines to 2000. Because this value was too low due to the relatively bright environment, the program didn't work properly on the first run because the bot didn't correct itself properly when it deviated.
While programming, we could observe that the AI is able to write easy programs, but as these become increasingly complex, such as programs with multiple different sensors, errors creep in.
This is because the artificial intelligence doesn't have specific information about the lighting conditions and the game table.

```c
#include <kipr/wombat.h>

#define LEFT_SENSOR 0
#define RIGHT_SENSOR 1
#define BUMPER 2
#define THRESHOLD 3800

void follow_line() {
    while(1) {
        if(digital(BUMPER)) {
            ao();
            break;
        }

        int left = analog(LEFT_SENSOR);
        int right = analog(RIGHT_SENSOR);

        if(left > THRESHOLD && right > THRESHOLD) {
            motor(0, 50);
            motor(1, 50);
        } else if(left < THRESHOLD) {
            motor(0, 20);
            motor(1, 50);
        } else {
            motor(0, 50);
            motor(1, 20);
        }
    }
}

int main() {
    follow_line();
    return 0;
}
```

Fig. 2: Program written by AI

We also wanted to test whether the AI could write us a program including a calibration of the Top Hat sensors. After several attempts and tweaks, this was surprisingly successful. The code works on the following principle: robot first reads a value one by one with the sensors on a white background, and then a value is read in with the sensors looking at the black tape. The resulting average

```c
#include <kipr/wombat.h>

#define LEFT_SENSOR 0
#define RIGHT_SENSOR 1
#define BUMPER 2
int THRESHOLD;

int calibrate_sensor(int sensor) {
    printf("Place sensor %d on WHITE and press enter.\n", sensor);
    while(!a_button()) {}
    int white = analog(sensor);
    msleep(500);

    printf("Place sensor %d on BLACK and press enter.\n", sensor);
    while(!a_button()) {}
    int black = analog(sensor);
    msleep(500);

    return (white + black) / 2;
}

void calibrate() {
    int left_threshold = calibrate_sensor(LEFT_SENSOR);
    int right_threshold = calibrate_sensor(RIGHT_SENSOR);
    THRESHOLD = (left_threshold + right_threshold) / 2;
    printf("Calibrated threshold: %d\n", THRESHOLD);
}
```

Fig.: 4 calibration code written by AI

value is now automatically the threshold value, from which the robot's individually controlled motors begin to align themselves automatically when deviating from the line.

With this extension, the problem can be solved by the code without calibration, because the robot now also adapts to the lighting conditions.

We also tried another experiment with ChatGPT. We wanted to test whether the AI could help us, for example, develop a strategy for grabbing the potato. We tried to give the AI as much information as necessary so that it could deliver a result. The AI first explained to us how to position the sensors and that we should place the heavy parts like the battery and Wombat controller on the base frame so that the robot is stable. ChatGPT also told us how to approach the robot, i.e. to move as slowly as possible towards the potato. It is also important to have a few landmarks and lines so that the robot knows where it is. We also received a suggestion for the best way to build it. The best system according to the AI would be a pincher gripper system that closes using servo movement. We should also install a sensor so that it can tell whether the robot has grabbed the potato. If not, it should move on to another task. ChatGPT also showed us a small test program, but it wasn't that good, which is understandable, since it couldn't create a truly good program for the Botball competition with the information it had. So, from this, we can conclude that AI's can be used to improve programs, but you must do most of it yourself, since you can't describe everything to the AI very well, especially the game table. Overall, AI provided a relatively stable strategy. The gripping system is even similar to the one we already have, but that was only possible by giving the AI as precise information as possible and correcting errors.

III. Conclusion

In conclusion, the use of artificial intelligence in robotics offers many advantages. In our paper, we first mentioned one of many methods for optimizing programs and data, so-called genetic programming. In this method, programs are optimized using different operators. We also demonstrated that AI obtains its data from public internet sources or information databases. Through machine learning, AI can improve itself independently. This also played a crucial role in our experiment with ChatGPT, as the codes were always improved by pointing out programming errors. All in all, artificial intelligence is already an important component in industry and everyday life. The question is whether AI will one day be able to completely take over human work.

References

[1] John R. Koza, James P. Rice: Automatic Programming of Robots using Genetic Programming, pages 1 – 3, available [online] at https://www.genetic-programming.com/jkpdf/aaai1992.pdf
(Accessed February 13 2025)

[2] Werner Bünnagel, Springer: pages 41-48; 2024; available [online] at https://books.google.at/books?hl=de&lr=&id=07syEQAAQBAJ&oi=fnd&pg=PR7&dq=wo+kommt+das+wissen+her+dass+KI+hat&ots=neVpJjNknS&sig=LJ6oNn0aMJjH9Qzmvwo6xB9ItB4&redir_esc=y#v=onepage&q=wo%20kommt%20das%20wissen%20her%20dass%20KI%20hat&f=false
(Accessed March 24 2025)

[3] For programming with AI we used: OpenAI: ChatGPT; available [online] at https://chatgpt.com

[4] John R. Koza: Genetic-Programming; July 8, 2007; available [online] at https://www.genetic-programming.org/
(Accessed March 7 2025)

[5] Kai Staats: Genetic Programming; available [online] at https://geneticprogramming.com/about-gp/tree-based-gp/
(Accessed March 26 2025)

[6] Felix Weipprecht, Digital-Zentral das Magazin; June 22, 2023; available [online] at https://digital-zentral.de/die-datenquelle-der-kuenstlichen-intelligenz-woher-bezieht-eine-ki-ihre-informationen/
(Accessed March 28 2025)

[7] Fraunhofer-IKS; available [online] at https://www.iks.fraunhofer.de/de/themen/kuenstliche-intelligenz.html
(Accessed March 28 2025)