International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 9 Issue 12 ǁ 2021 ǁ PP. 53-57

Gene Expression Programming Todistinguish Object in Robobuilder

Matias Kristian Kelviandy
[*1]*Department of Computer System, University of Gunadarma, Depok, Indonesia*

Abstract: *The purpose of this research to study and develop the GEP algorithm in real time environments such as on devices RoboBuilder, with the main objective GEP RoboBuilder designed on the device will make the learning process from the environment with real-time data and obtained from the robot. The end goal is to design and program code required to make the device RoboBuilder behave independently, in a sense can be learned from such an environment can distinguish two objects and behave in accordance with the conditions of an existing object. Objects are distinguished based on the distance through RoboBuilder PSD sensor. The results are shown in the form of the resulting object distance detection and chromosomal evolution of the best results through GUI Visual C # and console. Objects that are detected by sensors PSD RoboBuilder must have a minimum height was associated with RoboBuilder (greater than or equal to 30 cm) and a diameter of 1cm, to maximize the visibility function PSD sensor.*

Keywords:*Gene Expression Programming, Robobuilder , Microsoft Visual Studio C#.*

Date of Submission: 15-12-2021 Date of acceptance: 30-12-2021

I. INTRODUCTION

Artificial intelligence at this time has evolved, many algorithms are created to find solutions to a complex problem, including the algorithm Gene Expression Programming (GEP). GEP is an algorithm that is included in the category of evolutionary computation, the algorithm uses a system of evolution. Evolutionary system is an optimization process that aims to improve the ability of a system to survive in an environment that changes often, using a computer-based optimization for search solutions to problems as do natural selection, reproduction, mutation and survival of an individual. GEP algorithm excellence is to have a smart decision to take a decision in finding solutions to problems. RoboBuilder is a robot that can mimic some forms, like humans (Huno), spiders (spider) and others. These robots are suitable for the development of science and logic, the robot has the advantage of having control PID (Proporional, Integrated, Derivative) in each servo has, and there is ease in controlling the movement, because it has a protocol in every movement, the programmer only needs to activate protocols for certain movements without having to set some servo used for some movement.

Based on advantages RoboBuilder and algorithms GEP, this research discusses the control RoboBuilder in detecting and distinguishing objects in an environment through PSD sensor on RoboBuilder algorithm-based GEP, the control in question is able to detect objects in an environment and knowing the distance and behavior at every condition of the object with different parameter values generated by the GEP algorithm, so it can be given the circumstances RoboBuilder allows objects even in a different environment or dynamic environments..

1.1.1 Design of Distinguishing Two Objects On GEP Algorithm

Design of Distinguishing Two Objects On GEP Algorithm requires a mapping between the sensor and the motor, in operation of robot behavior is determined by the receipt of data from the sensors, how much distance received by the sensor PSD as sensory information to drive the motor, in the mapping algorithm GEP can be used for a solution to optimize the actions taken by Robot in situations and dynamic environments. The solution is an optimization best individual of each generation produced GEP to control robot.

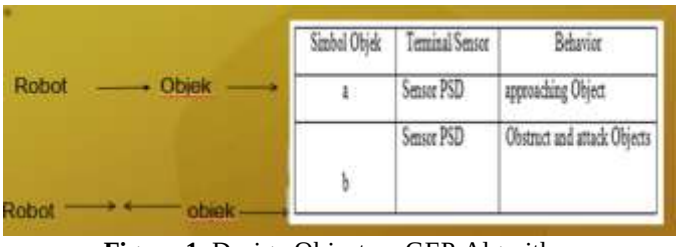

Figure 1. Design Object on GEP Algorithm

1.1.2 Design of Control Algorithm GEP

In determining the parameters in this study, is necessary to analyze the characteristics of the problem in advance, as space problems, availability of adequate mathematical model, and a solution can not be to be the most optimum but acceptable in the real world (Real Time). For these reasons, this study used a large population (1000) for one generation to the probability of the evolution of a uniform 10%, it is not required generation that a lot, about 10 generations of these parameters have reached the optimum value, and terminals used simple, just 2 terminal condition of the object (a and b), with a simple mathematical calculation functions.

Parameter	Pengaturan GEP
Generasi Maksimum	10
Ukuran populasi	1000
Jumlah *Gene*	2
Head size	10
Fungsi Matematis	+, -, /, *, Q(akar kuadrat)
Terminal	2 (a dan b)
Probabilitas Mutasi	10%
1-Point Recombination	10%
RIS Transposition	10%
randomly seeded runs	100

Figure 2. Design of Control Algorithm GEP

```
if (res2 > 15 & res2 < 30)
{
    ObjA = true;
    Console.WriteLine("Objek A: " + ObjA + "\n");
    checkBox1.Checked = true;
    pcr.runMotion(4); //jalan maju
    PauseForMilliSeconds(1000);
    rtb.Text = richTextBox1.Text;
    rtb.Text = rtb.Text + "objek A terdeteksi\n";
    richTextBox1.Text = rtb.Text;
    a = "a";
    A = (a);
    GEPProses();
}
```

Figure 3. Control Object A

```
else if (res2 < 15)
{
    ObjB = true;
    Console.WriteLine("Objek B: " + ObjB);
    checkBox2.Checked = true;
    pcr.runMotion(3);
    PauseForMilliSeconds(1000);
    pcr.runMotion(3);
    PauseForMilliSeconds(1000);
    pcr.runMotion(11);
    rtb.Text = richTextBox1.Text;
    rtb.Text = rtb.Text + "objek B terdeteksi :serang\n"  ;
    richTextBox1.Text = rtb.Text;
    b = "b";
    B = (b);
    GEPProses();
}
```

Figure 4. Control Object B.

1.2 THE SIMULATION

The simulation design of the software GEP C # contains a set of coding phases of the program to create an algorithm GEP is composed. C # programming language is a language that is object-oriented, therefore there are some parts of the stage of making programs such as the declaration of an object, of making the knot, manufacture equality of each genome, as well as the manufacture of the population by creating an iteration shaped generations, up to decide on a chromosome best of the best generation (optimal). The stages are divided from various files (extension .CS)

- **Genome**

Genome.cs an initial step of forming functions to form a GEP algorithm, as a function of the length gene, mutation, crossover and the conditions of a fitness chromosomes, which determine to be productive or not.

- **Simpul**

Simpul.cs is the stage of making a knot in the form of a tree-based representation , starting from the maximum number of tree nodes with full conditions, as well as beginning to enter the functions, both arithmetic function or object (a and b), forming an equation of the function and operation to generate the output value

- **Pers Genome.**

Press, genome.cs the next stage of simpul.cs that equation has been formed made a genome that is packaged in the form of an array list, and start entering the calculation of fitness with genes that have been formed in an equation

- **Population**

Populasi.cs is the final stage of formation GEP algorithm, at this stage an array of genes that form, has undergone a process of mutation, crossover and recombinasi thus forming a population consisting of several genes in the form of an array list, so as to form a shape-generation iterations and take the best genes from the population at best (optimum), one of the stops generation is no longer generated if repair genes or in other words have a fixed value in a number of the same fitness value

II. RESULT AND DISCUSSION

Generation test on GEP to determine how the GEP algorithm processing in a single iteration. In one iteration of the form generation, there are populations that number has been set in the default settings, the 1000 population. The bigger the population of fewer iterations / generations to achieve the optimum value, but the diversity of genes is not too much, but if the population size a little, then the iteration will be more and take a long time, but many have a diversity of genes.

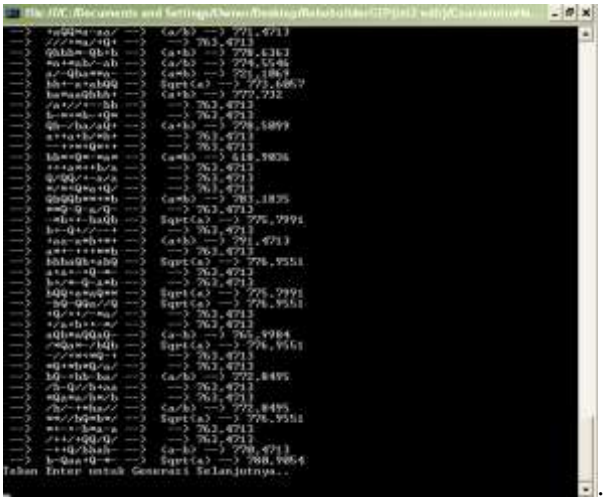

Figure 5. Iteration/Generation 1

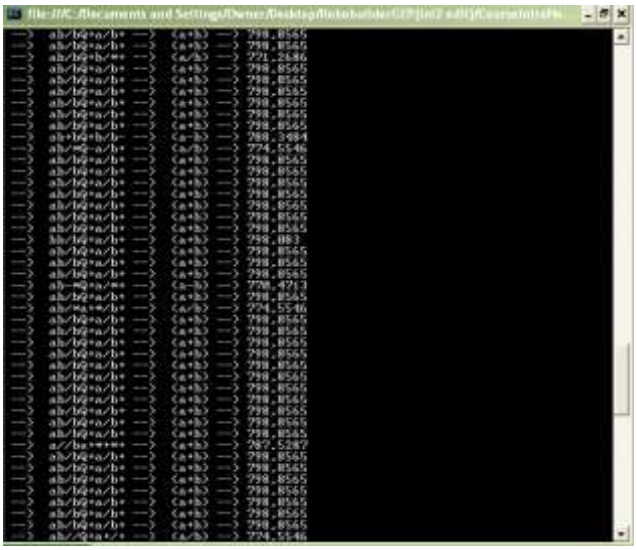

Figure 6. Iteration/Generation 10

In the figure 6 shows iteration / generation 10, in this generation, there is one gene dominates the population, namely genes :
ab/bQ+a/b+-> (a+b) -> 798.8565…….…. (4.1)

Figure 7. Detection Object A

Figure 8. Detection Object B

III. CONCLUSION

1. Algorithm Gene Expression Programming (GEP) in general can provide value parameter control in the work system real time as in environmental Robot (RoboBuilder), the value of this control can be used a robot to distinguish any condition of the object by giving each parameter value of the results of the process of evolution algorithm GEP against any condition of the object is encountered, the objects defined in the form of a variable (stationary object) and the variable b (moving objects) is used as a parameter GEP as a terminal and perform the process of evolution as mutation, crossover, and recombination, so that if one of the objects or the second object is detected or not known, each state will get a parameter value that is unique and different from other conditions, this makes the value parameter control to be able to recognize any existing conditions
2. Testing RoboBuilder PSD sensor response capabilities in detecting the distance has approximately 1 cm accuracy when compared with measurements. Object detection is affected by the object characteristics such as height and diameter of the object, wherein the detected object must have a minimum height of more than 30cm (height equivalent to RoboBuilder) with a 1cm diameter or width of the object (the length of two eye-shaped sensor RoboBuilder PSD). RoboBuilder PSD sensor capabilities in the detection distance is within the range of 10-50 cm with the limitations contained in visibility can only be detected in parallel to the object position sensor PSD (zero degrees)

REFERENCES

[1]. C. Ferreira. Gene Expression Programming: A New Adaptive Algorithm for Solving Problems. J.Complex Systems, 13(2):87-129, 2001
[2]. C. Ferreira. Gene Expression Programming: Mathematical Modelling by an ArtifficiaI ntelligence (2nd edition). Springer, 2006
[3]. D. K. Laszlo.Evolution of Intelligent Agents: A New Approach to Automatic Plan Design. In Proceedings of the IFAC Workshop on Control Applications of Optimization, pages 237-243. Elsevier, 2003
[4]. L. Zielinski, J. Rutkowski. Design Tolerancing with Utilization of Gene Expression Programming and Genetic Algorithm
[5]. Zhou,C.; Xiao, W.; Tirpak, T.M.; and Nelson, P.C. 2001.Discovery of Classification Rules by Using Gene Expression Programming. Technical Report, Dept. of Computer Science, University of Illinois at Chicag0
[6]. URL :http://code.google.com/p/robobuilderann/