Algorithm for optimizing a PID controller model based on a digital filter using a genetic algorithm

Ruslan V. Petrosian[1], Ihor A. Pilkevych[2] and Arsen R. Petrosian[1]

[1]Zhytomyr Polytechnic State University, 103 Chudnivsyka Str., Zhytomyr, 10005, Ukraine
[2]Korolyov Zhytomyr Military Institute, 22 Myru Ave., Zhytomyr, 10004, Ukraine

Abstract
The widespread use of digital signal processing distinguishes the current stage of development of science and technology. However, there are many developments for continuous signal processing. Such developments include methods for tuning the PID controller, so improving the digital PID controller model remains relevant. The problem of constructing a model of a digital PID controller, which can be used in robotic systems based on microcontrollers and programmable logic integrated circuits, is considered. It is proposed to use digital filtering methods as the basis for the regulator. The digital filter coefficients are calculated using a genetic algorithm. This approach makes it possible to improve the accuracy of the model, to ensure the calculation of the PID controller coefficients using classical methods for an analog PID controller. The software has been developed in the Python programming language that implements the proposed method. The modeling demonstrated the effectiveness of the developed model.

Keywords
digital filter, PID controller, genetic algorithm, model optimization algorithm

1. Introduction

The problem of effective control of technological processes, robotic systems, aircraft and other technical means remains relevant for many industries. For this purpose, regulators are used in many areas of science and technology. The most popular is the PID controller [1].

In recent years, the role and importance of computer technology in the life of modern society has increased dramatically and continues to grow, therefore, modern technical means are mostly implemented on the basis of microprocessors and microcontrollers, and many problem solutions are adapted to work in digital devices [2]. The PID controller did not escape its fate either [3].

Controller tuning can be done in several ways, including obtaining controller parameters in analytical form [1, 4, 5]. However, most of these methods are designed for analog PID control and are not suitable for digital because its model does not exactly match the PID controller, and, accordingly, the optimization of the digital PID controller model is relevant.

doors-2023: 3rd Edge Computing Workshop, April 7, 2023, Zhytomyr, Ukraine
e_rvs@ukr.net (R. V. Petrosian); igor.pilkevich@meta.ua (I. A. Pilkevych); xckaytz@gmail.com (A. R. Petrosian)
https://ztu.edu.ua/teacher/224.html (R. V. Petrosian); https://ieeexplore.ieee.org/author/37089181628 (I. A. Pilkevych); https://scholar.google.com/citations?user=2gvWqA4AAAAJ (A. R. Petrosian)
0000-0002-0388-8821 (R. V. Petrosian); 0000-0001-5064-3272 (I. A. Pilkevych); 0000-0003-0960-8461 (A. R. Petrosian)

2. Theoretical background

In general, the control system for any object has the form shown in figure 1.

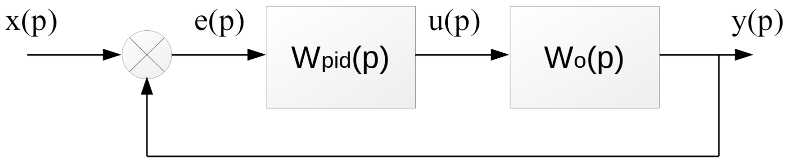


Figure 1: Control system model: $W_{pid}(p)$ – PID control transfer function, $W_o(p)$ – control object transfer function, $x(p)$ – input action, $y(p)$ – output signal, $e(p)$ – error signal, $u(p)$ – control signal.

The time-domain control algorithm is implemented in accordance with the expression (1) [1]:

$$u(t) = K_p e(t) + \frac{1}{T_i}\int_0^t e(t)dt + T_d \frac{de(t)}{dt}, \tag{1}$$

where $K_p,\ T_i,\ T_d$ – proportional factor, constant of integration and constant of derivation of the controller, respectively.

In some cases, the following expression is used (2):

$$u(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{de(t)}{dt}, \tag{2}$$

where $K_p,\ K_i,\ K_d$ – PID controller coefficients.

Thus, the PID controller includes three components: proportional, integral and differential. The proportional component generates a control signal counteracting the deviation (mismatch) of the output signal from the set value. The greater the mismatch, the greater the impact on the control object. If the output signal is equal to the set value, then the error signal is zero, and therefore the control action of the proportional component is zero. The integrator is used to eliminate the static error. The differentiating component takes into account the rate of change of the output signal, which allows you to get better control of the object by predicting the output value of the signal [1, 3].

There are several groups for assessing the quality indicators of object management: direct, root, frequency, integral. In practice, direct quality indicators have found the greatest application. This is due to the fact that direct indicators of the quality of object management are determined directly by the transient characteristic [1]. The following quality indicators can be distinguished:

- steady-state output value;
- static error;
- regulation time;
- overshoot;
- attenuation rate;
- etc.

The choice of control quality indicators depends on the task in which the PID controller is used.

To ensure the required performance of regulation, it is necessary to calculate the coefficients of the PID controller.

There are many methods for calculating quality indicators. One of the first methods for calculating the parameters of PID controllers was proposed by Ziegler and Nichols [6]. This technique does not give very good results, but it is very simple, therefore it is still often used in practice. After calculating the parameters of the regulator, manual adjustment is required to improve the quality of regulation.

In work of Sablina and Markova [4], other methods for calculating the parameters of PID controllers are also considered, namely: Chien-Hrones-Reswick, Kuhn. Relay methods are also widely used [5, 7].

If the methods considered were developed relatively long ago, then the methods below are quite recent.

In [8, 9], methods for optimizing the parameters of a PID controller using a genetic algorithm are considered. In these works, the choice is analyzed: fitness functions, the main operators of the genetic algorithm, quality indicators.

The possibility of using neural networks to optimize the PID controller coefficients was considered by Kadu and Patil [10]. The main focus of the article is on the analysis of the stability of such systems.

Many works are related to the determination of the optimal parameters of the PID controller for specific control objects [9, 10, 11, 12].

A large number of works are linked to the development of the digital PID [13, 14, 15, 16, 17, 18, 19]. However, in fact, all the articles cited can be divided into two groups.

The first group of works [15, 16, 17, 18] is based on expression (3) given in [13]:

$$u\left(n\right) = K_p e\left(n\right) + K_{id} \sum_{k=0}^{n} e(k) + K_{dd}(e\left(n\right) - e(n-1)), \tag{3}$$

where T_k – sampling period, $K_{id} = K_i T_k$, $K_{dd} = K_p / T_k$.

Expression (3) is often written in a recurrent form to reduce computational costs (4):

$$\begin{aligned} u\left(n\right) = u\left(n-1\right) + K_p \left(e\left(n\right) - e\left(n-1\right)\right) + K_{id} e\left(n\right) + \\ + K_{dd} \left(e\left(n\right) - 2e\left(n-1\right) + e\left(n-2\right)\right). \end{aligned} \tag{4}$$

The second group of works [3, 14, 19] is based on the expression (5):

$$u\left(n\right) = u\left(n-1\right) + K_1 e\left(n\right) + K_2 e\left(n-1\right) + K_3 e\left(n-2\right), \tag{5}$$

where $K_1 = K_p + K_i + K_d$, $K_2 = -K_p - 2K_d$, $K_3 = K_d$.

The analysis showed that expressions (4) and (5) are practically identical (the control signal $u(n)$ depends on the last three readings of the error signal). The main difference between them is that expression (4) allows you to determine the coefficients of a digital PID controller based on an analog prototype. For expression (5), the coefficients $K_1,\ K_2,\ K_3$ must be selected when manually adjusting the control system. It may seem that these coefficients depend on the controller coefficients $K_p,\ K_i,\ K_d$, so they can be calculated, but this is not the case. This is easy to see if you pay attention to the fact that these coefficients $K_1,\ K_2,\ K_3$ do not take into account the sampling rate.

Taking into account the above, it follows that digital and analog PID controllers are not considered as different entities, therefore, methods for calculating the controller coefficients are considered regardless of whether it is digital or analog. However, as the analysis has shown, there are at least two implementations of a digital PID controller, so the calculation methods must take into account the structure of the controller. In addition, as will be shown later, the digital controller (4) is not a complete analogue of the classic analog PID controller. Thus, the problem of the algorithm for optimizing the digital PID controller is relevant.

3. Results and discussion

This section will discuss the implementation of a digital PID controller. The controller will be based on a digital filter [20]. The method for calculating the filter coefficients will be performed using a genetic algorithm [21, 22].

3.1. Digital filter

Digital signal processing is used wherever it is necessary to perform tasks such as filtering, compressing, recovering, controlling, measuring a signal: audio, video, or any signal coming from any source [23].

Filtering is the most common digital processing task, which is implemented using digital filters: filters with a finite impulse response (FIR filters); filters with infinite impulse response (IIR filters). In general, a digital filter is understood as a hardware or software implementation of a mathematical algorithm, the input of which is a digital signal, and the output is another digital signal modified by the filter.

The main operations of information filtering include: noise suppression, smoothing, prediction, differentiation, signal separation, etc.

The main advantages of digital filters over analog filters:

- may have parameters that are impossible to implement in analog filters, for example, linear phase response;
- do not require calibration, because their performance does not depend on the destabilizing factors of the external environment, for example, temperature;
- input and output data can be saved for later processing;
- accuracy of digital filters is limited by the capacity of the filter coefficients.
- can be easily rearranged to filter a different frequency range, for example, by changing the data sampling rate.

In general, the digital filter is described by the following expression (6):

$$y\left(n\right)=\sum_{k=0}^{K-1} b_k\cdot x(n-k)+\sum_{m=0}^{M-1} a_m\cdot y\left(n-m\right), \tag{6}$$

where $b_k,\ a_m$ – filter coefficients; $x\left(n\right),\ y(n)$ – input and output signal; $K,\ M$ – number of filter coefficients $b_k,\ a_m$ respectively.

Expression (6) is also called IIR filter. Such a filter is often used when you need to perform filtering with a minimum number of arithmetic operations. If all the coefficients a_m are equal to zero, then such a filter is called an FIR filter. In this case, the digital filter will be described by the following difference expression (7):

$$y\left(n\right)=\sum_{k=0}^{K-1}h(k)\cdot x(n-k), \tag{7}$$

where $h\left(k\right)=b_k$ – impulse response of an FIR filter.

The amplitude-frequency response (AFR) of such a filter will have the following form (8):

$$H(\omega)=\sum_{k=0}^{K-1}h(k)\cdot e^{-j\omega n} \tag{8}$$

where ω – circular frequency.

In many digital signal processing applications, the use of FIR filters is preferable because they have the following advantages:

- filter group delay constant (linear phase FIR filters);
- FIR filters are always stable.

For FIR filters to be linear phase, the impulse response must be symmetric or antisymmetric [20]. In this case, four types of FIR filters are possible (table 1).

Table 1
Types of linear phase filters and their characteristics.

Filter type	Impulse response	Number of impulse response coefficients	Amplitude-frequency response
I	symmetrical	odd	$H(\omega)=\sum_{k=0}^{(K-1)/2}a(k)\cdot\cos(\omega k)$
II	symmetrical	even	$H(\omega)=\sum_{k=1}^{K/2}b(k)\cdot\cos(\omega(k-1/2))$
III	antisymmetric	odd	$H(\omega)=\sum_{k=1}^{(K-1)/2}c(k)\cdot\sin(\omega k)$
IV	antisymmetric	even	$H(\omega)=\sum_{k=1}^{K/2}d(k)\cdot\sin(\omega(k-1/2))$

Here $a\left(0\right)=h\left(\frac{K-1}{2}\right),\ a\left(k\right)=2h\left(\frac{K-1}{2}-k\right),\ c\left(0\right)=0,\ c\left(k\right)=2h\left(\frac{K-1}{2}-k\right),\ k=1,\ 2,\ 3,\ \ldots,\ \frac{K-1}{2}$, $b\left(k\right)=2h\left(\frac{K}{2}-k\right),\ d\left(k\right)=2h\left(\frac{K}{2}-k\right),\ k=1,\ 2,\ 3,\ \ldots,\ \frac{K}{2}$.

An example of an antisymmetric FIR filter with an even number of coefficients is a differentiating filter, the AFR of which corresponds to figure 2.

3.2. Genetic algorithm

Genetic algorithm is a heuristic algorithm, which is a kind of evolutionary algorithms, with the help of which optimization problems are solved using methods of natural evolution, similar to natural selection [21, 22].

The range of tasks solved using the genetic algorithm is very wide:

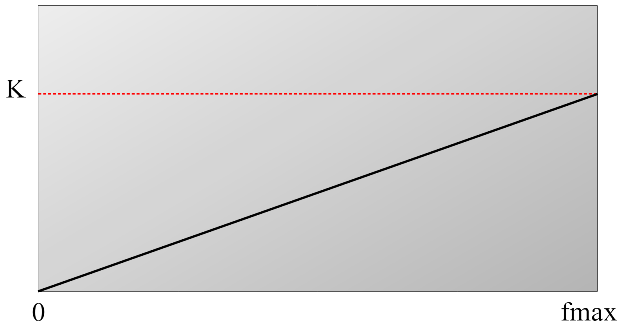


Figure 2: AFR of the differentiating term.

- numerical optimization problems;
- traveling salesman tasks;
- scheduling;
- function approximation;
- artificial neural network training;
- etc.

The key concept of a genetic algorithm is an individual that encodes a possible solution to a problem. An individual is characterized by a chromosome or a set of chromosomes. The atomic unit of a chromosome is a gene (most often encoded by one bit). When solving the problem, a population of individuals is created. Each individual is assessed by the degree of fitness, which is determined in the task by the fitness function. Thus, individuals are determined that are better adapted to the "environment" (have the best solution).

The genetic algorithm is iterative, therefore, at each iteration, a new population of individuals is generated, which has better fitness than the previous one. This process continues until the desired results are achieved, or the number of iterations exceeds the threshold.

The peculiarity of the genetic algorithm is that the set of solutions is immediately improved, unlike many other optimization algorithms.

To create a new population, genetic operators are applied to current individuals: crossing, mutation, selection.

Crossover is an operator that applies to two parents. Most often, each of them is divided into two parts at the same random gene position. Formed individuals are a combination of the first and second parts of chromosomes from different parents (figure 3). The considered option is called the one-point crossing method. There are other crossover methods: multipoint, uniform, etc.

Mutation is an operator that makes a change in a gene at a random position on the parent chromosome. The mutation is designed to reduce the likelihood of optimization at the local maximum. There are the following mutation methods: bit inversion, exchange, permutation, etc.

Selection is an operator aimed at selecting individuals in accordance with a certain criterion. There are various selection methods: roulette method, tournament selection, ranking method, etc.

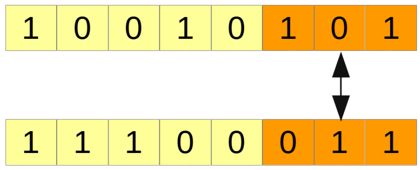


Figure 3: Crossover operator.

The following sequence describes how the genetic algorithm works:

1. Generating an initial population;
2. Calculation of the fitness of chromosomes;
3. Selection of initial chromosomes (solutions) with the best fitness values for creating a new population;
4. Performing the crossing operation;
5. Performing a mutation operation;
6. Calculation of the fitness of chromosomes;
7. If the stop condition is met, return the chromosome with the best fitness value, otherwise go to step 3 to process the new population.

As mentioned above, the genetic algorithm refers to heuristic search algorithms, so it is necessary to adjust the hyperparameters. To make sure that the hyperparameters used made it possible to obtain a solution close to optimal, it is essential to control changes in chromosome fitness from generation to generation.

3.3. Development of a PID controller model

Let's define the transfer function of the analog PID controller. For this, it is necessary to perform the Laplace transform of formula (2) with zero initial conditions $u\left(0\right)=0$. As a result, we get the following expression:

$$W_A\left(p\right)=K_p+K_i\frac{1}{p}+K_dp, \tag{9}$$

where p – Laplace operator.

If in expression (9) we substitute $p=j\omega$, then we obtain an expression for the frequency response of the analog PID controller (10):

$$W_A\left(\omega\right)=K_p-K_i\frac{j}{\omega}+jK_d\omega. \tag{10}$$

Let's look at the frequency response of the regulator at $K_p=10,\ K_i=1,\ K_d=1$ (figure 4).

The AFR of the analog PID controller shows that the integrating component has an effect in the low-frequency range, and the differentiating component in the high-frequency range.

Now let's compare the AFR of the differentiating components of the analog and digital PID controllers.

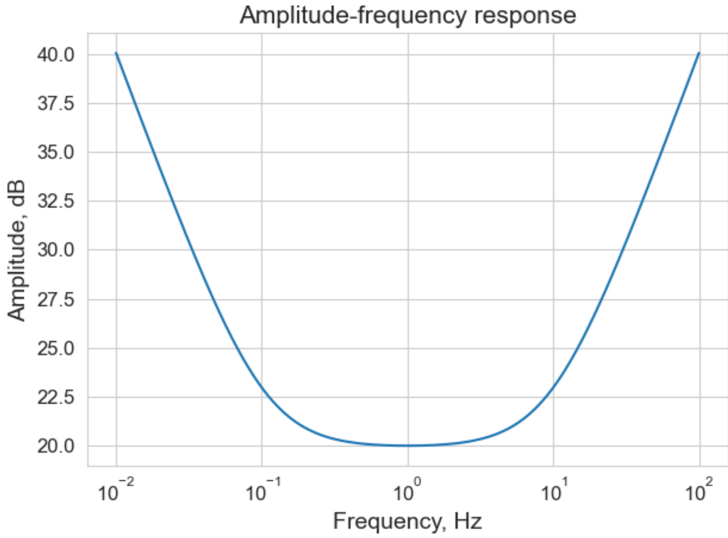


Figure 4: AFR of the PID controller.

First, let's write the frequency response of the derivative component of the analog PID controller. It can be seen from formula (10) that its frequency response is determined by the expression (11):

$$D_A(\omega) = K_d\omega. \tag{11}$$

Now let's determine the AFR of the differentiating component of the digital PID controller. From expression (3) it can be seen that in the time domain the differentiating component has the following form (12):

$$d_D(n) = K_{dd}\left(e(n) - e(n-1)\right). \tag{12}$$

In the brackets of expression (12) there is a digital filter of the first order, therefore the frequency response can be determined from formula (8). As a result, we will have the following expression (13):

$$D_D(\omega) = 2K_{dd}\sin\left(\frac{\omega T_k}{2}\right). \tag{13}$$

Figure 5 shows the AFR of both regulators. For convenience of comparison, the frequencies are normalized (the sampling rate is taken equal to one), and the coefficients are taken equal to: $K_d = 1$, $K_{dd} = 1$.

Obviously, the AFR of the differentiating component of the analog PID controller corresponds to figure 2, but the digital one does not. Such a term works as a differentiating one only for 1/3-1/4 of the initial interval, however, in this area its influence is minimal, because the integrating and proportional components prevail there. Figure 6 shows the relative error. As you can see from the figure, the maximum error is more than 35%.

To eliminate this drawback, it is necessary to set a new model of the differentiating component. One of the options is the use of digital processing methods, namely the use of the previously mentioned FIR filter with a linear phase.

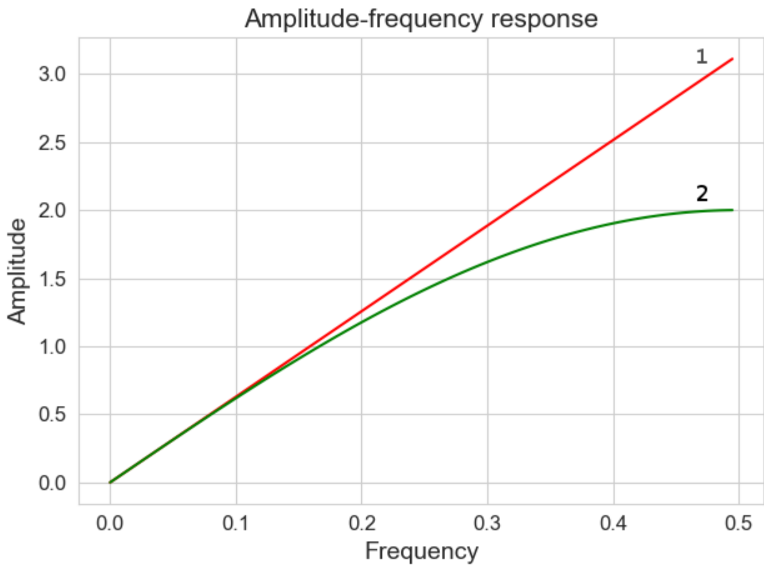


Figure 5: AFR of the differentiating component of PID controllers: 1 – analog; 2 – digital.

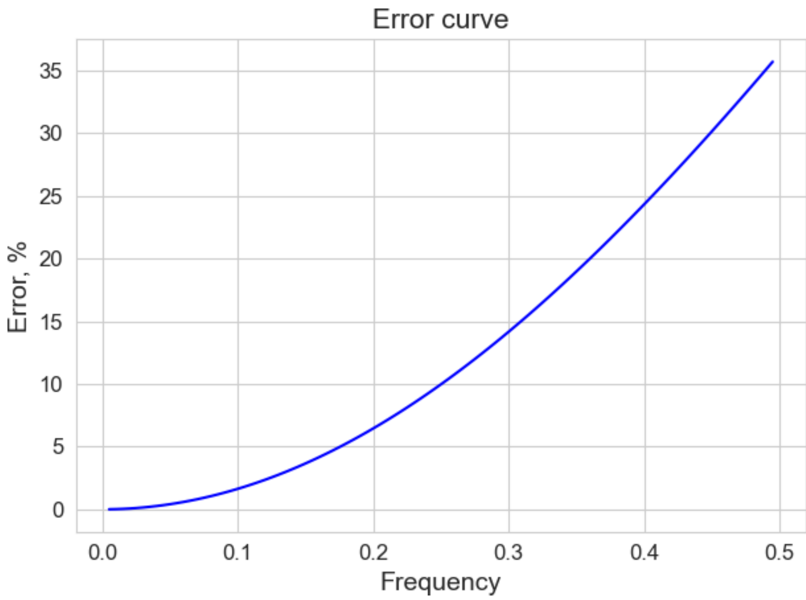


Figure 6: Relative error of the differentiating component of AFR of the digital PID controller.

From table 1 it follows that there are 4 types of such filters. Filters of types III and IV have an imaginary part of the AFR. However, the type III filter cannot always be used as such. The reason is that the value of the transmission coefficient at the maximum frequency will be equal to zero $H(\omega_{max}) = 0$ regardless of the filter coefficients (table 1). Such a filter can be used as a differentiating filter only in the initial section. We need to use the entire range, so for our task the best solution would be to use a type IV filter.

From expression (7) and table 1, it follows that the differentiating component for a digital PID controller can be represented in the form of expression (14):

$$y\left(n\right)=\sum_{l=0}^{L-1}h(l)\cdot\left(x\left(n-l\right)-x(n+l-2L+1)\right), \tag{14}$$

where L – number of independent coefficients.

In this case, taking into account expression (3), the algorithm for implementing a digital PID controller will be described by the expression (15):

$$u\left(n\right)=K_pe\left(n\right)+K_{id}\sum_{k=0}^{n}e(k)+K_{dd}\sum_{l=0}^{L-1}h(l)\cdot\left(e\left(n-l\right)-e(n+l-2L+1)\right). \tag{15}$$

For simplicity, we will call it PPID. By analogy, we can write an expression similar to the recurrence formula (4) or recurrently recorded only an integrating part (this will reduce the likelihood of overflow and reduce the number of arithmetic operations) in the following way (16):

$$u\left(n\right)=K_pe\left(n\right)+I\left(n\right)+K_{dd}\sum_{l=0}^{L-1}h(l)\cdot\left(e\left(n-l\right)-e(n+l-2L+1)\right), \tag{16}$$

where $I\left(n\right)=I\left(n-1\right)+K_{id}e\left(n\right)$ – integrating component.

To obtain the final model of the PPID controller, it is necessary to determine the coefficients $h(l)$, where $l=1,\ 2,\ 3,\ \ldots,\ L-1$.

To synthesize the filter (14), a fitness function is required. The synthesis of the filter with the best uniform approximation will be performed in the form of the problem of minimizing the weighted Chebyshev norm (17):

$$e=max\left(W\left(\omega\right)\left|H\left(\omega\right)-\widehat{H}(\omega)\right|\right)\rightarrow min, \tag{17}$$

where $H\left(\omega\right)$, $\widehat{H}(\omega)$ – AFR of the approximated and approximating filters, respectively, $W\left(\omega\right)$ – weight function.

3.4. Experiments

To test the model of the digital PPID controller, we will carry out a number of experiments. Let us synthesize a digital FIR filter (14). There are many methods for their design in the scientific literature [20]. The most widely used are the classical methods for calculating FIR filters: weighing method; frequency sampling method; least squares method; best uniform approximation method. The first two are not optimization methods, but are fairly easy to use. The third and fourth methods are referred to as optimization methods. The fourth method allows obtaining the best results, but, as a rule, it is impossible to determine analytically the function of the best uniform approximation. However, in this case, the synthesis of the FIR filter will be carried out using the genetic algorithm [24], which will allow us to obtain some advantages, for example, when searching for the values of the filter coefficients, we will take into account the effect of quantization.

When solving a problem with a genetic algorithm, it is necessary to isolate the phenotype that determines the real object. In our case, the filter coefficients that will form an individual will act as a phenotype (figure 7).

h(0)	h(1)	h(2)	. . .	. . .	h(L-1)

Figure 7: The structure of the individual.

Fitness function will be described by expression (17). The AFR of the differentiating part of the analog PID filter (11) at $K_d = 1$ (figure 5, graph 1) will act as the AFR of the approximated filter. The AFR of the approximating filter will be determined by a type IV filter (table 1).

The simulation was carried out using the Python programming language. To implement the genetic algorithm, it is necessary to tune the hyperparameters. In our case, they will have the following form:

POPULATION = 100 # number of individuals in the population
SURVIVOR = 0.2 # survival probability
MUTATION = 0.1 # possibility of mutating of an individual
GENERATIONS = 250 # maximum number of generations

Below is the fitness function code in the Python programming language, which corresponds to the expression (17), where *prototype* is an instance of the *PrototypeFIR class* of the filter being approximated (figure 2); *fir* – an instance of the *Fir1T class* approximating the filter (7).

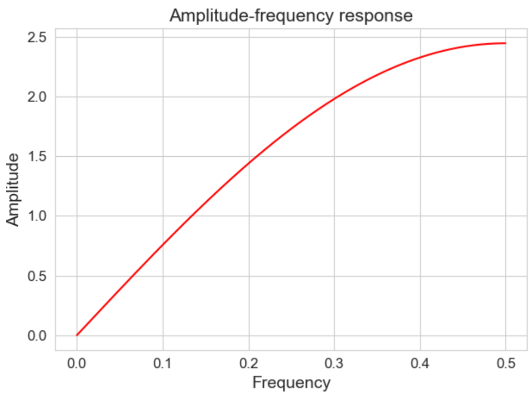
```
def fitness(individual): # fitness function
  fmax = prototype.getSamplingFrequency() / 2
  fir = fir1t.Fir1T(fmax, individual)
  emax = 0
  for fi in prototype.getReferencePoints():
    e = abs(prototype.getGain(fi) -
      fir.getGain(fi))*prototype.getWeight(fi)
    if e > emax:
      emax = e
  return  emax,
```

Figure 8 shows the synthesis of a differentiating component PPID controller.

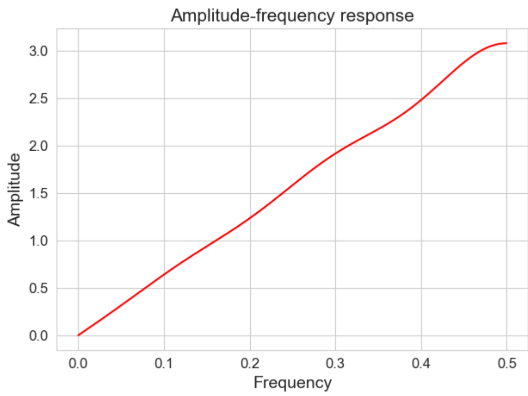
Table 2 shows the calculated coefficients of filters of different orders, and also indicates the approximation error of the differentiating component of the PPID controller.

If we compare the proposed model of the PPID controller with the original digital PID controller (3) at $L = 1$, we can see that the expressions differ only by the factor $h(0)$ (table 2). However, due to him, the error was reduced by 15%. Figure 9 shows the relative error of this PPID controller.

Figure 10 shows the tendency of decreasing the error with an increase in the number of coefficients $h(l)$.



a)



b)

Figure 8: AFR of the differentiating component of the PPID controller: a) $L = 1$; b) $L = 6$.

Table 2
FIR filter coefficients for the developed model of a digital PID controller.

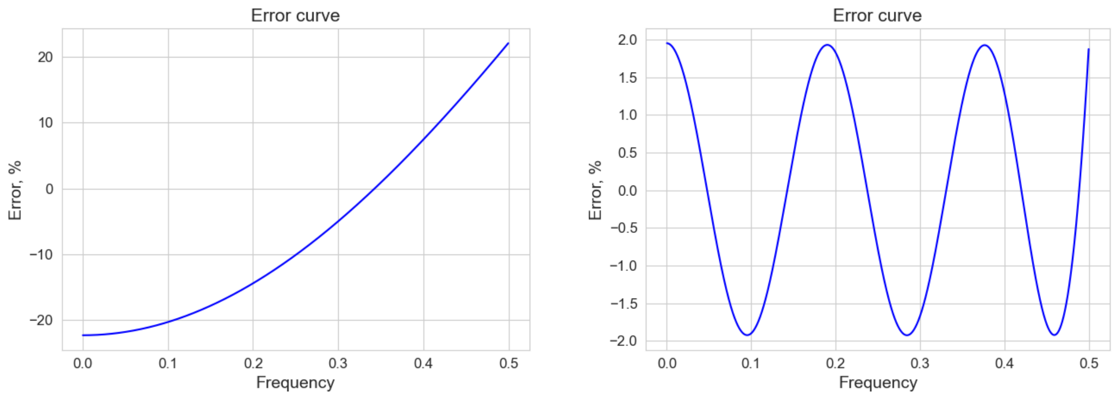
L	h(l)	Error, %
1	1.22323099	22.32
2	-0.1310637, 1.30919328	8.40
3	0.05080995, -0.16309586, 1.28291046	4.77
4	-0.02893289, 0.06681262, -0.14717449, 1.27747904	3.25
5	0.01963494, -0.03861118, 0.05499173, -0.14416749, 1.27554355	2.44
6	-0.01467901, 0.02619673, -0.0291035, 0.05288374, -0.14306129, 1.27468666	1.95
7	0.0116237, -0.01945604, 0.01826656, -0.02751572, 0.05212775, -0.14253636, 1.27425729	1.62
8	-0.00959958, 0.01532948, -0.01266609, 0.01698343, -0.02694503, 0.05174926, -0.14223351, 1.2739692	1.38
9	0.00816843, -0.01257872, 0.00936974, -0.01156839, 0.01650426, -0.02665169, 0.0515316, -0.14203773, 1.27377827	1.20
10	-0.00710013, 0.01063027, -0.00727018, 0.0084291, -0.01118041, 0.01626373, -0.026466, 0.05138662, -0.14191183, 1.27366718	1.07
11	0.00627216, -0.00918011, 0.00583902, -0.00643919, 0.00810167, -0.01099786, 0.01613495, -0.02635945, 0.05128535, -0.1418134, 1.2735769	0.96

It can be seen that $4 \leq L \leq 6$ is sufficient for solving most control problems in robotic systems.

4. Conclusion

The problem of constructing a model of a digital PID controller, which can be used in robotic systems based on microcontrollers and programmable logic integrated circuits, is considered.

The regulator is based on digital filtering methods. It is proposed to use an FIR filter with a linear phase of the IV type as a filtering device. This made it possible to fairly accurately approx-



a) b)

Figure 9: Relative error of the differentiating component of the AFR of the PID controller: a) $L = 1$; b) $L = 6$.

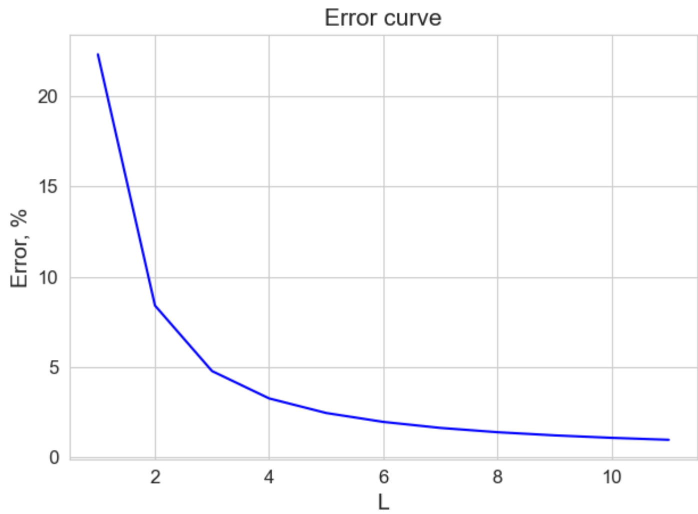


Figure 10: Changing error differentiating component with increasing L PPID controller.

imate the differentiating component. So, for a classic digital PID controller, the introduction of one coefficient has reduced the relative frequency response error by 15%. In addition, the PID controller model was developed with the ability to use ready-made methods for calculating the PID controller coefficients.

The digital filter coefficients are calculated using a genetic algorithm. The phenotype is the filter coefficients. The Chebyshev norm was used as a fitness function.

The simulation results were carried out using the Python programming language.

Data for all filters up to 21 orders (up to 11 independent coefficients) has been analyzed.

As shown in the work, for most control problems in robotic systems, it is sufficient to use filters with 4-6 independent coefficients.

Perspectives for further research consist in testing the proposed methods on a wider range of problems, studying the effects of finite bit depth, and analyzing the structure of the PID controller.

Acknowledgments

We would like to express our gratitude to our relatives who supported and helped us. We would also like to express our gratitude to all the organizers of the *doors-2023: 3rd Edge Computing Workshop*, especially Tetiana Vakaliuk.

References

[1] A. R. Petrosian, R. V. Petrosian, O. V. Pidtychenko, Optimization of the PID Controller Model Based on a Digital Filter, Vcheni zapysky TNU imeni V.I. Vernadskoho. Seriia: Tekhnichni nauky 32(71) (2021) 129–134. URL: https://www.tech.vernadskyjournals.in.ua/journals/2021/4_2021/22.pdf. doi:10.32838/2663-5941/2021.4/20.

[2] A. I. Herts, I. M. Tsidylo, N. V. Herts, S. T. Tolmachev, Cloud service ThingSpeak for monitoring the surface layer of the atmosphere polluted by particulate matters, CTE Workshop Proceedings 6 (2019) 363–376. doi:10.55056/cte.397.

[3] T. Wescott, PID without a PhD, 2018. URL: https://www.wescottdesign.com/articles/pid/pidWithoutAPhd.pdf.

[4] G. V. Sablina, V. A. Markova, Tuning a PID Controller in a System with a Delayed Second-Order Object, Optoelectronics, Instrumentation and Data Processing 58 (2022) 410–417. doi:10.3103/S8756699022040112.

[5] R. P. Borase, D. K. Maghade, S. Y. Sondkar, S. N. Pawar, A review of PID control, tuning methods and applications, International Journal of Dynamics and Control 9 (2021) 818–827. doi:10.1007/s40435-020-00665-4.

[6] J. G. Ziegler, N. B. Nichols, Optimum Settings for Automatic Controllers, Trans. ASME 64 (1942) 759–765. URL: https://web.archive.org/web/20170918055307if_/http://staff.guilan.ac.ir:80/staff/users/chaibakhsh/fckeditor_repo/file/documents/Optimum%20Settings%20for%20Automatic%20Controllers%20(Ziegler%20and%20Nichols,%201942).pdf. doi:10.1115/1.4019264.

[7] S. Hornsey, A Review of Relay Auto-tuning Methods for the Tuning of PID-type Controllers, Reinvention: an International Journal of Undergraduate Research 11 (2018). URL: https://warwick.ac.uk/fac/cross_fac/iatl/reinvention/archive/volume5issue2/hornsey/.

[8] A. Mirzal, S. Yoshii, M. Furukawa, Pid parameters optimization by using genetic algorithm, 2012. arXiv:1204.0885.

[9] A. Jayachitra, R. Vinodha, Genetic Algorithm Based PID Controller Tuning Approach for Continuous Stirred Tank Reactor, Advances in Artificial Intelligence 2014 (2014) 791230. doi:10.1155/2014/791230.

[10] C. Kadu, C. Patil, Design and Implementation of Stable PID Controller for Interacting Level Control System, Procedia Computer Science 79 (2016) 737–746. doi:10.1016/j.procs.2016.03.097.

[11] T. Samakwong, W. Assawinchaichote, PID controller design for electro-hydraulic servo valve system with genetic algorithm, Procedia Computer Science 86 (2016) 91–94. doi:10.1016/j.procs.2016.05.023.
[12] M. Trafczynski, M. Markowski, P. Kisielewski, K. Urbaniec, J. Wernik, A Modeling Framework to Investigate the Influence of Fouling on the Dynamic Characteristics of PID-Controlled Heat Exchangers and Their Networks, Applied Sciences 9 (2019) 824. doi:10.3390/app9050824.
[13] P. Bhandari, P. Z. Csurcsia, Digital implementation of the PID controller, Software Impacts 13 (2022) 100306. doi:10.1016/j.simpa.2022.100306.
[14] A. Maghsadhagh, Implementation of PID Controller by Microcontroller of PIC (18 Series) and Controlling the Height of Liquid in Sources, Advances in Robotics & Automation 5 (2016) 1000156. URL: https://tinyurl.com/muac76z7.
[15] Y. Cheng, Y. Chen, H. Wang, Design of PID controller based on information collecting robot in agricultural fields, in: 2011 International Conference on Computer Science and Service System (CSSS), 2011, pp. 345–348. doi:10.1109/CSSS.2011.5974664.
[16] N. M. Mohamed, A. A. Abdalaziz, A. A. Ahmed, A. A. Ahmed, Implementation of a PID control system on microcontroller (DC motor case study), in: 2017 International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE), 2017, pp. 1–5. doi:10.1109/ICCCCEE.2017.7866088.
[17] D. Hou, PID Control on PIC16F161X by using a PID Peripheral, 2015. URL: http://ww1.microchip.com/downloads/en/AppNotes/90003136A.pdf.
[18] Atmel, AVR221: Discrete PID Controller on tinyAVR and megaAVR devices, 2016. URL: http://ww1.microchip.com/downloads/en/AppNotes/Atmel-2558-Discrete-PID-Controller-on-tinyAVR-and-megaAVR_ApplicationNote_AVR221.pdf.
[19] S. Masade, S. Parmar, A. J. Bhanushali, Speed Control for Brushless DC Motors using PID Algorithm, Whitepaper, Einfochips, 2016. URL: https://silo.tips/download/speed-control-for-brushless-dc-motors-using-pid-algorithm-whitepaper.
[20] D. G. Manolakis, J. G. Proakis, Digital Signal Processing: Principles, Algorithms, and Applications, 4 ed., Pearson Education Limited, 2006.
[21] J. Carr, An Introduction to Genetic Algorithms, 2014. URL: https://www.whitman.edu/documents/academics/mathematics/2014/carrjk.pdf.
[22] M. Mutingi, C. Mbohwa, Grouping Genetic Algorithms: Advances and Applications, volume 666 of *Studies in Computational Intelligence*, Springer Cham, 2017. doi:10.1007/978-3-319-44394-2.
[23] T. M. Nikitchuk, T. A. Vakaliuk, O. A. Chernysh, O. L. Korenivska, L. A. Martseva, V. V. Osadchyi, Non-contact photoplethysmographic sensors for monitoring students' cardiovascular system functional state in an iot system, Journal of Edge Computing 1 (2022) 17–28. doi:10.55056/jec.570.
[24] M. Mobini, Digital IIR Filter Design Using Genetic Algorithm and CCGA Method, International Journal of Mechatronics, Electrical and Computer Technology 2 (2012) 222–232. URL: https://www.aeuso.org/includes/files/articles/Vol2_Iss6_222-232_Digital_IIR_Filter_Design_Using_Gen.pdf.