Multi-Objective PID-Controller Tuning for a Magnetic Levitation System using NSGA-II

Gerulf K.M. Pedersen and Zhenyu Yang
Esbjerg Institute of Technology, Aalborg University
Esbjerg, Denmark
gp@cs.aaue.dk, yang@cs.aaue.dk

ABSTRACT

This paper investigates the issue of PID-controller parameter tuning for a magnetic levitation system using the non-dominated sorting genetic algorithm (NSGA-II). The magnetic levitation system is inherently unstable and the PID-controller parameters are hard to find using conventional methods. Based on four different performance measures, derived from the step response of the levitation system, the algorithm is used to find a set of non-dominated parameters for a PID-controller that can stabilize the system and minimize the performance measures.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Control theory*; J.6 [**Computer Applications**]: Computer-aided engineering—*Computer-aided design*

General Terms

Design

Keywords

PID Control, Parameter tuning, Multi-objective optimization, NSGA-II

1. INTRODUCTION

Magnetic fields have been found useful for a variety of different applications ever since they were discovered, and the number is still growing. Many of todays electrical applications contain technologies that utilize magnetic fields. Some such applications include electrical motors, cathode ray tubes and even magnetic levitated vehicles. The work presented in this paper is very much related to the latter, namely magnetic levitation.

The use of magnetic levitation is becoming more and more popular. Not only has it found its way into the transport sector with the magnetic levitated vehicles, but also for research into fusion reactors where magnetic levitation is used for controlling the extremely hot plasma in the reactors. Stabilization of motor shafts using electromagnetic bearings is also a growing field which takes advantage of the lower friction when compared to regular ball-bearings. However, one of the difficult aspects when it comes to magnetic levitation is the fact that magnetic fields are non-uniform. In fact, they are highly non-linear and the strength of a magnetic field not only depends on the distance to the source but also on the orientation of the field at the source and at the target.

In this paper a one-dimensional levitation system will be investigated. For controlling the system a simple and well-known controller will be used such that an item can be levitated at a given height and be moved in the vertical direction. However, because of the complexity of the system and also the simplicity of the controller it is quite difficult to find good controller parameters analytically. It is usually possible to use either empirical methods or human intuition to find a set of parameters, but it will most likely have relatively low performance if at all possible. This is where the use of an evolutionary algorithm comes into the picture.

Some examples where genetic algorithms have been used for finding parameters for PID controllers include [5] and [8] who considered the case of optimal control of robotic manipulators. Another example is found in [9] where genetic programming was used to find a set of optimal control parameters to two different problems, an auto-regressive exogeneous system and a continuous stirred tank reactor system. In that case the result was compared to a known set of optimal parameters. Also, in [1] several different approaches, including genetic algorithms, simulated annealing, neural networks, and combinations of those, for tuning of a PID controller for a one-axis magnetic bearing. These examples have all been based on optimization of a single objective whereas the multi-objective approach have yet to be investigated.

A multi-objective evolutionary algorithm is capable of giving a set of parameters that most likely will be equally suited at meeting some specified performance measures. Thus, by using such a multi-objective algorithm it will be possible to find several set of controller parameters that all should be relatively good at controlling the vertical movement of the one-dimensional magnetic levitation system.

This paper will first give a description of the magnetic levitation system followed by a description of the simple PID controller. The performance measures of the magnetic levitation system will then be described and those perfor-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
GECCO'06, July 8–12, 2006, Seattle, Washington, USA.
Copyright 2006 ACM 1-59593-186-4/06/0007 ...$5.00.

mance measures will later be used as objectives for the chosen multi-objective algorithm, which is the non-dominated sorting genetic algorithm (NSGA-II) [2]. A short description of NSGA-II and the different parameters used for running the algorithm along with some implementation issues will then be presented. Finally, the results obtained will be given followed by a conclusion.

However, let us first start out with a description of the magnetic levitation system.

2. MAGNETIC LEVITATION SYSTEM

In this section a brief description of the physical system, which is the foundation for this investigation, will be given along with the corresponding mathematical model. In connection with that, some of the coefficients for the mathematical model will also be calculated based on preliminary measurements of the physical system. So, let us first take a look at the physical system.

2.1 Physical System

A one-dimensional levitation system was constructed in our laboratory using an aluminum frame as shown in figure 1. Slits were milled on the sides of the framework for ease of mounting and for allowing adjustment of an optical sensor system. The electromagnet device consists of a solenoid and an iron core composed of thin steel plates riveted together. The levitating object is a table tennis ball with a diameter of 39.5mm. A small NIB (neodymium, iron, boron) magnet was glued to the top inside the ball and an M4 nut was glued to the bottom acting as counterweight to the NIB magnet. An optical sensor system for measuring the distance between the solenoid bottom and the levitating ball was developed using two LEDs (IR333-A) and a photodiode array (Hamamatsu 16-element Si photodiode array, type S5668-1). The sensor system is mounted inside an aluminum housing with a milled slit facing to the possible operating range as shown in figure 2. Such a geometry makes the sensor system less sensitive to background light.

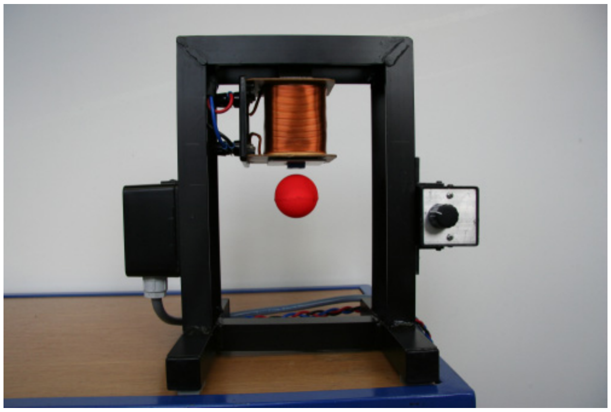

Figure 1: The physical system

With this description of the physical system in place it is now time to look at the mathematical foundation which will be used for simulations later on.

2.2 Mathematical Model

Under assumption that the used material has a linear characteristic, i.e., the magnetization density only depends on the magnetic field density [10], the magnetic flux can be approximated by a function

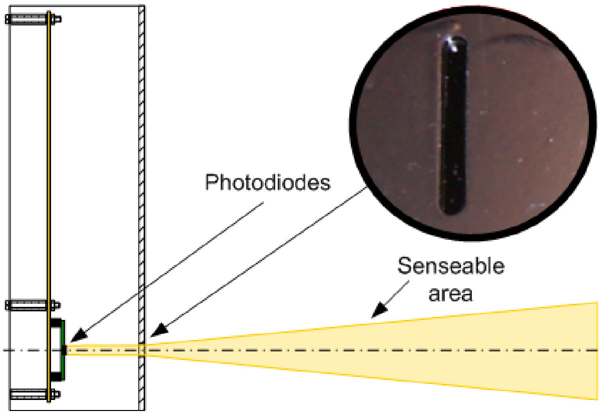

Figure 2: Geometric construction of sensor system

$$\lambda(t)\hat{=}\lambda(i(t),x(t)) = L(x(t))i(t), \tag{1}$$

where $i(t)$ denotes the current through the solenoid, and $x(t)$ denotes the displacement of the levitating object relative to the solenoid bottom. $L(x)$ denotes the total inductance which is assumed to be a function of $x(t)$. Assume $L(x)$ has the form [6]

$$L(x) = L_1 + \frac{L_0}{1+(\frac{x(t)}{a})} \ , \tag{2}$$

where $L_0 = L(0) - L(\infty)$, $L_1 = L(\infty)$ and a is a constant coefficient.

According to the electromagnetic theory [10], the magnetic co-energy, denoted as W, can be described as

$$W(t) = \int_0^{i(t)} \lambda(\bar{i}, x(t))d\bar{i} \ . \tag{3}$$

By inserting (1) and (2) into (3), we obtain

$$W(t) = \frac{1}{2}(L_1 + \frac{L_0}{1+(\frac{x(t)}{a})})i^2(t) \ . \tag{4}$$

Then, the magnetic force, denoted as $f(t)$, can be determined from the magnetic co-energy by

$$f(t) = \frac{dW(t)}{dx_a} \ ,$$

where x_a represents the force acting on the axis, which is equal to the $x(t)$ axis. Then from (4) we have

$$f(t) = -\frac{1}{2}\frac{L_0 i^2(t)}{a(1+\frac{x(t)}{a})^2} \ . \tag{5}$$

Denote the mass of the levitating object as m and the gravity acceleration as g. By neglecting the air friction, the dynamics of the levitating object can be obtained from Newton's second law as

$$m\frac{dx^2(t)}{dt^2} = mg + f(t) = mg - \frac{1}{2}\frac{L_0 i^2(t)}{a(1+\frac{x(t)}{a})^2} \ . \tag{6}$$

From circuit analysis of the electromagnet part, we have

$$u(t) = Ri(t) + \frac{d(L(x)i(t))}{dt} \ , \tag{7}$$

where R represents the coil resistance, and $u(t)$ represents the input voltage to the coil. Substitute (2) into the above (7) and we have

$$\begin{aligned}\frac{di(t)}{dt} &= \frac{aR + Rx(t)}{aL(0) + L_1 x(t)} i(t) \\ &\quad + \frac{(a + x(t))(aL(0) + L_1 x(t))}{aL_0} i(t) \frac{dx(t)}{dt} \\ &\quad + \frac{a + x(t)}{aL(0) + L_1 x(t)} u(t) \ . \qquad (8)\end{aligned}$$

Equations (6) and (8) can then be considered as the mathematical model of the magnetic levitation system which can be used for simulating the response of the physical system. In the model there are some system specific coefficients. Before it is possible to simulate the system it is necessary to determine the values of these coefficients, which will be done next.

2.3 Coefficient Identification

The system coefficients L_0 and a in (6) can be identified through an experimental approach [6]. A series of experiments is organized to find the currents required to levitate the object at different equilibrium positions. One such result is plotted in figure 3.

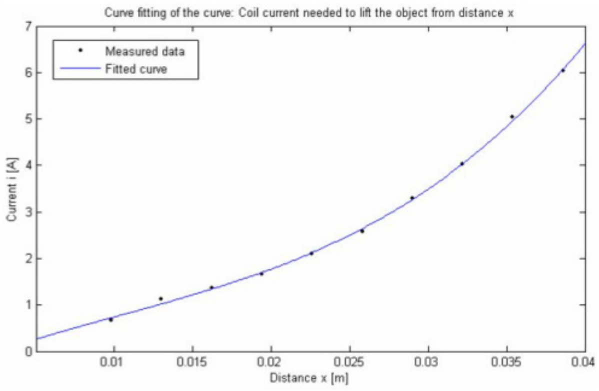

Figure 3: The equilibrium points and corresponding required currents

The test result as shown in figure 3 did not show a precise linear relationship between displacement and required current to levitate the object. However, if the levitating object only moves within a small operating range, e.g., our consideration is $0 \sim 3cm$, we can get a reasonable approximation to a linear relationship from figure 3.

The coefficients L_0 and a can be determined by using two equilibrium points located close to each other and their corresponding currents, denoted as x_1, x_2 and i_1, i_2, respectively. From (5) we obtain

$$-\frac{1}{2}\frac{L_0 i_1^2}{a(1+\frac{x_1}{a})^2} = -\frac{1}{2}\frac{L_0 i_2^2}{a(1+\frac{x_2}{a})^2} \ .$$

Then the coefficient a can be calculated by

$$a = \frac{i_2 x_1 - i_1 x_2}{i_1 - i_2} \ . \qquad (9)$$

After a is determined, L_0 can be determined based on the set (x_1, i_1) by

$$L_0 = \frac{2amg(1+\frac{x_1}{a})^2}{i_1^2} \ . \qquad (10)$$

The values of a and L_0 are calculated using different points from the measured data and are finally averaged such that the mean values can be used in our model. The obtained system coefficients are listed in table 1.

Table 1: Coefficients used in the considered system.

Description	Symbol	Value	Unit
Coil inductance	$L(0)$	$2.0 \cdot 10^{-3}$	H
Coil resistance	R	0.8	Ω
Mag. inductance	L_0	$16.7 \cdot 10^{-3}$	H
Mag. ind. coeff.	a	1.18	m
Mass of object	m	$4.16 \cdot 10^{-3}$	kg

With the model and corresponding coefficients in place it is possible to continue with a description of the desired controller for the system.

3. PID CONTROL

In order to control the magnetic levitation system such that it will be possible to manipulate the vertical position of the levitated object it is necessary to implement a suitable controller. The choice of controller that will be used for the levitation system in this paper has fallen on the Proportional, Integral, Derivate (PID) controller.

The structure of a PID controller is given by

$$u(t) = K_p e(t) + K_i \int e(\tau) d\tau + K_d \frac{de(t)}{dt} \ , \qquad (11)$$

where $u(t)$ is the output of the controller at time t, $e(t)$ is the error between the actual and the desired output, and K_p, K_i and K_d are weightings of the different terms. Depending on the choice of K_p, K_i and K_d the response of a controller to a given error signal will vary significantly.

The PID controller is a very simple controller, but the major drawback is that there is no analytical way of finding the optimal set of parameters (K_p, K_i, K_d). Empirical methods such as Ziegler-Nichols tuning [4] can be used for the tuning, but because the method has been based on empirical data for a wide range of problems it is almost certain that the parameters found could be improved. Also, due to the inherent instability of the magnetic levitation system it is difficult to use such a method for this application. Further, such a method would only result in a single controller whereas it might be desirable to be able to choose between several possible controllers. For that reason it was decided to use a multi-objective evolutionary algorithm to find several possible parameter settings that would result in good performance of the magnetic levitation system.

In the next section the different performance measures that will be used for optimization of the parameters K_p, K_i and K_d will be given.

3.1 Objective Functions

When a controller is designed for a specific system it is often desired that multiple design requirements are met. The magnetic levitation system is no exception. When evaluating the performance of the different PID-controller parameters for the magnetic levitation system a step response is simulated in Simulink and different performance measures are investigated. The performance measures that will be used in this paper are:

- Overshoot (M_p)
- Rise time (t_r)
- Settling time (t_s)
- Integrated absolute error (IAE)

An illustration of the performance measures is given in figure 4.

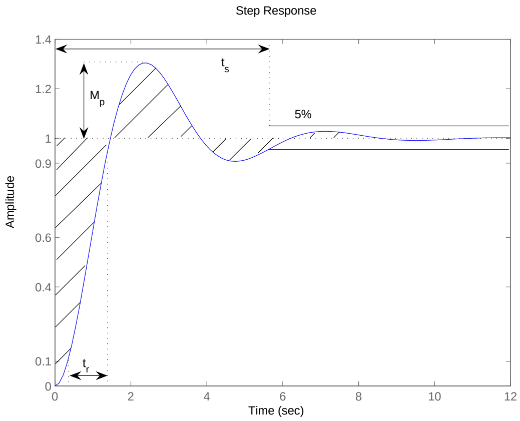

Figure 4: Performance measures obtained from a step response.

Overshoot is a percentage indicating how much the system response to the step exceeds the following steady-state value. Rise time is the time it takes for the response to rise from the pre-step value to the post-step steady-state value and is measured as the time it takes the system going from 10% to 90% of the step size. The settling time is the amount of time elapsed from the step was issued until the post-step system value is within 5% of the steady-state value. The last performance measure, the integrated absolute error, represents the area between the two signals, namely the reference and the system response.

Each of these performance measures will be included as objectives to be minimized as their inter-dependence will depend highly on the system under consideration. For this case the system under consideration is the non-linear magnetic levitation system.

With the performance measures in place it is now time to look at the algorithm that will be used to solve the presented multi-objective optimization problem.

4. NSGA-II

The choice of algorithm fell on the non-dominated sorting genetic algorithm (NSGA-II) [2] as it is a widely used and capable algorithm. The principle behind NSGA-II is that the non-dominated solutions that usually occur for multi-objective optimization problems are all treated as equals. This allows the algorithm to evolve a set of non-dominated solutions that are all equally well suited for solving the specific problem given the performance measures specified. By using this algorithm for tuning of a PID-controller for the magnetic levitation system, it will be possible to obtain a varied set of different solutions that should perform well with regard to minimization of all the specified performance measures.

In the following some of the detailed parameters for the execution of the algorithm will be presented.

As is obvious from section 3 it is desired to find 3 parameters (K_p, K_i, K_d). Different combinations of parameters will then result in different adherence to the given performance measures and it will be possible to distinguish how different parameters affect the different performances.

Many of the NSGA-II run-time parameters used for this problem are the same as the NSGA-II default values. Those default parameters are summarized in table 2.

Table 2: Parameters used for running NSGA-II.

Representation type	Real values
Crossover probability	0.9
Mutation probability	0.33
SBX parameter	10
Mutation parameter	50
Rigid bounds	1 (yes)

The choice of real valued representation was made to ensure that the precision of the parameters would not be compromised by a choice of precision, which can happen for binary representations. A crossover probability of 0.9 ensures a good mixing of genetic material and a mutation probability of 0.33 ensures that on average one parameter of each individual will be mutated. The mutation probability can also be expressed as $\frac{1}{n_{param}}$ where n_{param} is the number of parameters in an individual which for this application is 3.

When it comes to the simulated binary crossover parameter (SBX) and the mutation parameter it was decided to use the default values of 10 and 50 respectively since they provide a reasonable distribution of solutions for the different operations. The rigid bounds are maintained because at least one of the boundaries for the controller parameters must be fixed. This choice is based on insight of the magnetic levitation system.

Some of the parameters not mentioned in table 2, such as the ranges for K_p, K_i and K_d, require a little insight into the problem and those will be discussed a little further along with some implementation issues.

4.1 Implementation

One of the issues that needs to be resolved before the problem can be solved is to identify the ranges of K_p, K_i and K_d. In fact, those values are dependent on the implementation of the simulation and the representation used. This is especially an issue because the controller implemented in the simulation is a discrete version of the one given in (11). The discrete version that is implemented is given by

$$u(k) = u(k-1) + K_p\left(1 + \frac{T}{T_i} + \frac{T_d}{T}\right)e(k) - K_p\left(1 + 2\frac{T_d}{T}\right)e(k-1) + K_p\frac{T_d}{T}e(k-2) \ , \quad (12)$$

where T is the sample time of the discrete-time system, $T_i = \frac{K_p}{K_i}$ is the integration time and $T_d = \frac{K_d}{K_p}$ is the rate time. In fact, in the implementation it is these K_p, T_i and T_d that are found. From insight into the system it can be determined that due to displacement being defined positive in a downwards direction K_p must be negative, as a negative error signal should result in a positive increase of the control signal and vice versa. The range was originally set to [-200,0], but was later modified to [-1000,0] as some initial testing showed a tendency of K_p to converge at the lower limit -200. The ranges for the integration time T_i and the rate time T_d are both set to [0,15] based on the fact that a negative value should not be allowed as it, similar to the choice of K_p range, would give response in the wrong direction and any higher values would be infeasible since the dynamics of the resulting controller would become too slow. In the simulation a sample time T of 1ms has been set corresponding to a sample frequency of 1kHz.

The simulation is performed by applying a step in the displacement reference signal from 0.0025m to 0.007m. The system is initially at rest at 0.0025m and when the step to 0.007m is applied 0.5s into the simulation, the resulting dynamics can be observed. In order to ensure that the dynamics have time to settle within the scope of the simulation a total of 10s is simulated, thus providing the system dynamics 9.5s to settle after the step has been applied.

The simulation itself is performed in Simulink which is part of Matlab. Because of this and the fact that the NSGA-II version used is written in C, a simple handshaking protocol using files, similar to the one described in [7], is used to ensure that the communication between the programs would be possible.

Due to the computational complexity of the simulations it was desired to keep the number of total evaluations to a minimum such that the results could be obtained within a reasonable amount of time. This meant that a population size of 50 individuals was chosen along with a maximum number of generations of 150. It is evident that the population size is low when compared to the three-dimensional search space that needs to be covered by the algorithm, but by allowing the algorithm to run for 150 generations it is expected that the algorithm will be capable of finding a reasonable set of well performing solutions even though the chance of premature convergence is relatively high.

Besides from the use of the 4 objectives a constraint on the allowable amount of overshoot has also been formulated. This was done based on preliminary testing which shoved that the overshoot could increase beyond 1800% which is very unreasonable. Thus, a constraint was put on the overshoot such that only values below 100% was allowed. This constraint would also help to contain the individuals near the area deemed most useful on the non-dominated front, namely the region with no excessive overshoot. So, despite the low number of individuals it is expected that the coverage of the non-dominated front will be adequate.

With all of the implementation issues in place we can take a look at the results obtained.

5. RESULTS

The results when optimizing PID controller parameters for the magnetic levitation system using NSGA-II turned out to be quite interesting. In order to obtain a better understanding of the results and the system in general, the discussion of the results will also take the distribution of controller parameters in the space spanned by K_p, T_i and T_d into account. This is similar to the concept of innovization introduced in [3] although the concept will only be used as a guideline for the current investigation.

In figure 5 a pairwise combination of performance measures for the last generation is shown.

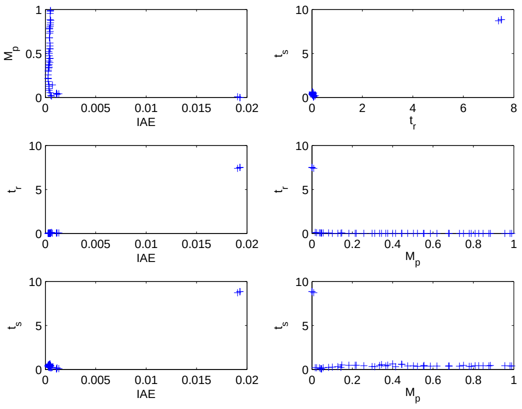

Figure 5: Pairwise combination of performance measures for last generation.

The interesting thing to notice from this result is that there are some significantly isolated points among the final non-dominated set. In fact, they are so significantly different from the rest that it is hard to conclude anything else from the figure. Also, on the distribution of solutions for the space spanned by K_p, T_i and T_d in figure 6 there are a few solutions, those having large T_d values near the top, which stand out.

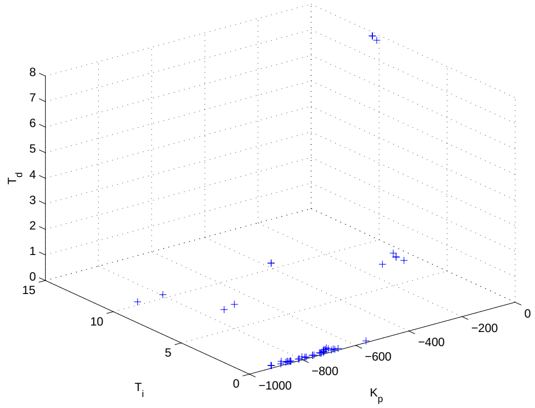

Figure 6: Plot of parameters K_p, T_i and T_d for last generation.

The remainder of individuals shown in figure 6 in fact have T_d values much smaller than 1. Not surprisingly are the outliers from figure 6 the same individuals that cause the outliers of figure 5.

In order to get an understanding of why such a phenomenon takes place it was decided to simulate the step response of one of these outliers. The step response of the individual with the values $K_p = -10.0706$, $T_i = 10.3086$ and $T_d = 7.9326$ can be seen in figure 7.

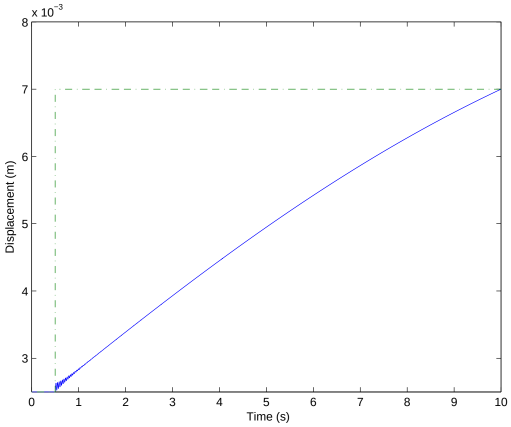

Figure 7: Step response from a solution amongst the outliers.

In the figure it is obvious that the algorithm took advantage of the finite simulation time. In order to obtain very small overshoot values the system was tuned to reach the step threshold specifically near the end of simulation such that any continuation of that trajectory would not be penalized in the performance measure. As this was obviously an exploit used by the algorithm it was decided to remove the outliers from the final generation and re-plot the remaining solutions such that a set of good controller parameters could be found.

With the outliers removed, the pairwise combination of performance measures for the final generation is shown in figure 8.

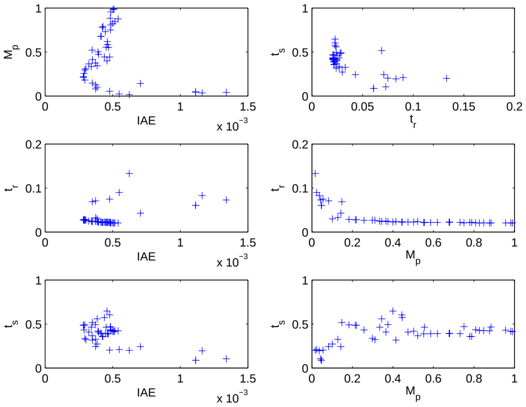

Figure 8: Pairwise combination of performance measures for last generation without outliers.

Several interesting features can be seen from this final set of performance measures. The first thing that can be noticed from the top left plot is that the overshoot is almost independent of the integrated absolute error. Only for very small overshoot values can it be seen that the integrated absolute error for some individuals increase significantly. Otherwise, it can be seen that the integrated absolute error only increases slightly whenever the overshoot increases.

The plots in the middle of figure 8 show that most of the solutions found have almost the same rise time t_r, but for small overshoot values it can be seen that the rise time increases along with the integrated absolute error.

From the bottom plots of figure 8 it can be seen that the settling times t_s also have a tendency, even not as clearly as the rise time, to be around the same values. Not surprisingly does the settling time decrease with a decrease in overshoot, but this also causes the integrated absolute error to increase.

The plot in the top right of figure 8 shows that there are some trade-offs between the rise time and the settling time which is also quite evident by comparing the two plots in the middle right and bottom right.

From these observations it can be seen that there is a clear tendency amongst the solutions to have similar values with the only exception being the overshoot which varies rather uniformly over the entire range from 0-100%. This uniform distribution gives an indication of a good spread of solutions along the non-dominated front when it comes to the overshoot performance measure.

In order to confirm the observations made from looking at the performance measures it was decided to also take a look at the distribution of K_p, T_i and T_d for the case where the outliers have been removed. This is illustrated in figure 9.

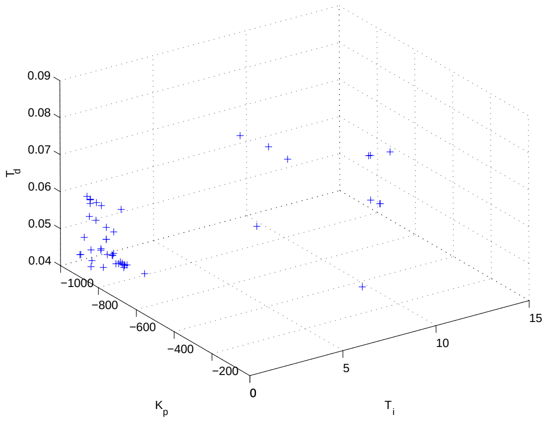

Figure 9: Plot of parameters K_p, T_i and T_d for last generation.

In the figure it is quite evident that there is a large grouping of individuals for small values of T_i and K_p values below -600. Comparing the performance of the individuals from that group with each other show that they generate the responses that have 20% and higher overshoot, whereas the individuals with higher T_i values generate the responses with very small overshoot.

In order to better understand the differences between the resulting controllers a simulation of a typical controller from the cluster and one from the more isolated area was simulated and the results are shown in figure 10 and figure 11.

Figure 10 shows the response of a typical controller from the cluster. The top plot shows the response for the entire simulation time and the bottom plot shows the simulation from 0.4s to 1.4s. The corresponding performance

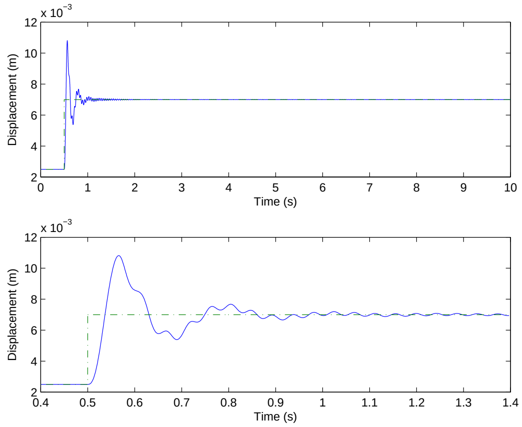

Figure 10: System response for $K_p = -811.2746$, $T_i = 0.0208$ and $T_d = 0.0626$ resulting in a rather oscillatory response.

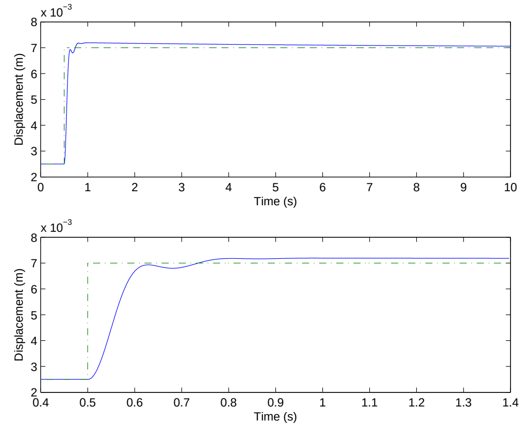

Figure 11: System response for $K_p = -47.0373$, $T_i = 8.0471$ and $T_d = 0.0882$ resulting in a rather smooth response.

measures for this individual are IAE=$5 \cdot 10^{-4}$, $M_p = 84.82\%$, $t_r = 21ms$ and $t_s = 0.425s$. The latter 3 performance measures can also roughly be observed in the figure. The interesting thing about this controller is that the system response consists of a fast oscillation on top of a slower one. The fast rise time for this controller is most likely due to this fast oscillation overlaid on the response, and the main factor in that oscillation is the size of K_p which is very aggressive towards positional errors. However, this aggressiveness of the controller might come at a price. The simulations here have been performed under ideal conditions without noise and it could be expected that the controller would be rather poor at handling the noisier conditions of the real system.

Figure 11 shows the response of one of the controllers not in the main cluster. The top plot shows the entire simulation and the bottom plot shows the simulation from 0.4s to 1.4s. The corresponding performance measures for this controller are IAE=$1.3 \cdot 10^{-3}$, $M_p = 4.27\%$, $t_r = 73ms$ and $t_s = 0.106s$. It can be seen that the decrease in overshoot comes at the price of a longer rise time and a larger integrated absolute error. The controller did have a rather fast settling time, but as can be seen from the figure, the controller never fully reaches the new set-point within the simulation running time. The main reason for this behavior of the controller is the large value of T_i which takes a long time to average out the error accumulated just after the step was applied. Contrary to the controller from figure 10, the controller shown in figure 11 is expected to behave better for non-ideal conditions as the somewhat slower response is expected to be less sensitive towards any noise that might exist on the real system.

So far, the results have solely been based on simulations. In order to prove that the method would also apply to the real system, some tests were performed on the physical system. However, it turned out that the parameters that worked well for the theoretical case did not perform too good for the actual system as can be seen in figure 12.

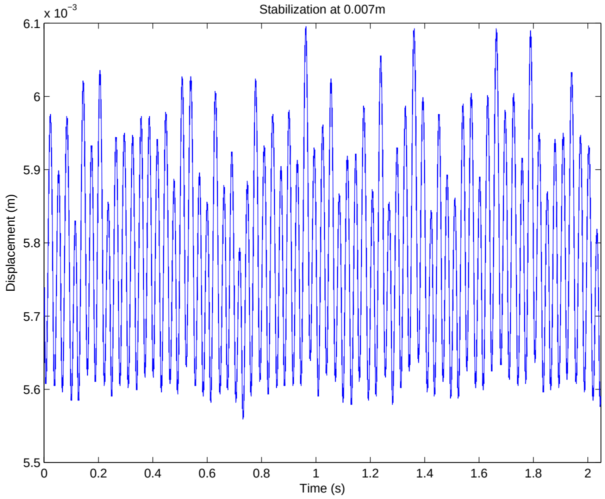

Figure 12: System response for the physical system with $K_p = -47.0373$, $T_i = 8.0471$ and $T_d = 0.0882$ when stabilizing at 0.007m.

In fact, any of the evolved controllers were incapable of stabilizing the physical system adequately. This illuminates the main drawback of using evolutionary computation for tuning controllers in this manner, namely that the model used in the evolutionary process must match the physical system very precisely. It is suspected that the mathematical model did not match the physical system to such a degree that the results obtained using simulation could be directly transferred to the actual system.

6. CONCLUSION

In this paper it has been investigated how PID-controller parameters for a magnetic levitation system could be found using NSGA-II. First, an introduction to the field of magnetic levitation and the basic equations was given followed by a short description of PID-control. The desired performance measures were then presented and the implementation of these into NSGA-II along with a short description of the parameters used in the algorithm was discussed. Finally, some results were obtained and they were analyzed using the innovization principle.

The first conclusion that can be made is that the algorithm succeeded in finding a set of non-dominated solutions which not only stabilized the simulation of the magnetic levitation system but also had reasonable dynamic responses. There were a few outliers that took advantage of the finite simulation time, but most of the solutions did not exploit this. Most of the remaining solutions were clustered together with the only major difference being in the choice of the K_p parameter resulting in different amount of overshoot for the system. These solutions are, however, less suited for implementation on a real system, as the aggressiveness of the controllers will make them very sensitive to noise. Apart from the cluster there were some solutions which had a slower response and a corresponding lower overshoot. The rise times for these controllers were slower but this slower response should have been beneficial for implementation on the real system where noise is an issue.

Unfortunately, it turned out that the parameters found using the simulations did not perform satisfactory for the physical system. There could be several reasons for this discrepancy, including modeling errors, approximations, noise, and sensor dynamics. In order to find a more suited set of PID parameters, it would be necessary to take another look at the simulation model to make it conform more to the physical system. Because, if the simulation model had adequately described the dynamics of the system, the evolved controllers would have reflected that and not have been overly aggressive.

Generally, it can be concluded that tuning of controller parameters using multi-objective algorithms is quite promising as it can be used when no analytical or empirical methods are applicable. There are of course both benefits and pitfalls to using such an approach, but it is the authors belief that, if sufficiently accurate models can be derived and simulated, the benefits clearly outweighs the drawbacks and we expect that the method will become more widely used in the future for other types of systems and controllers.

7. ACKNOWLEDGMENTS

The authors would like to thank Stefan Kjærgaard Greisen, Jette Rørgaard Hansen, and Niels Allan Pedersen for their assistance in connection with modelling issues and performing experiments on the system.

8. REFERENCES

[1] M. Chiaberge, J. J. Merelo, L. M. Reyneri, A. Prieto, and L. Zocca. A comparison of neural networks, linear controllers, genetic algorithms and simulated annealing for real time control. *European Symposium on Artificial Neural Networks*, pages 205–210, 1994.

[2] K. Deb, A. Pratap, and S. Moitra. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. *Parallel Problem Solving from Nature - PPSN VI*, pages 849–858, 2000. NSGA-II code available at KanGAL website: 'http://www.iitk.ac.in/kangal/'.

[3] K. Deb and A. Srinivasan. Innovization: Innovation of design principles through optimization. Technical Report 2005007, Kanpur Genetic Algorithms Laboratory (KanGAL), Indian Institute of Technology Kanpur, Kanpur, PIN 208016, India, 2005.

[4] A. E.-N. Gene F. Franklin, J. David Powell. *Feedback Control of Dynamic Systems*. 4th edition, 2002.

[5] D. P. Kwok and F. Sheng. Genetic algorithm and simulated annealing for optimal robot arm PID control. *Proceedings of the First IEEE Conference on Evolutionary Computation*, 2:707–713, 1994.

[6] V. Oliveira, E. Costa, and J. Vargas. Digital implementation of a magnetic suspension control system for laboratory experiments. *IEEE Trans. on Education*, 42(4):315–322, Nov. 1999.

[7] G. K. M. Pedersen. *Towards Automatic Controller Desing using Multi-Objective Evolutionary Algorithms.* Doctoral dissertation, Aalborg University, Dept. of Control Engineering, Aalborg, Denmark, 2005.

[8] B. Porter and C. Allaoui. Performance measures in the genetic design of digital controllers for robotic manipulators. *Proceedings of 1995 IEEE International Conference on Evolutionary Computation*, 2:509–514, 1995.

[9] D. Searson, M. Willis, and G. Montague. Chemical process controller design using genetic programming. *Genetic Programming 98*, pages 359–364, 1998.

[10] M. Woodson. *Electromechanical Dynamics, art I.* John Wiley & Sons Inc., New York, 1968.