

# Digital Controller Design of an Industrial Sewing Station using Evolutionary Algorithms

Eric Henrique Moretti  
Polytechnic Institute of Bragança  
Campus de Santa Apolónia  
5300-253 Bragança, Portugal

Gustavo Gonçalves Coelho  
Polytechnic Institute of Bragança  
Campus de Santa Apolónia  
5300-253 Bragança, Portugal

João Paulo Coelho  
Polytechnic Institute of Bragança  
Research Centre in Digitalization and  
Intelligent Robotics (CeDRI)  
Campus de Santa Apolónia  
5300-253 Bragança, Portugal  
jpcoelho@ipb.pt

**Abstract**—Development of increasingly efficient production methods is a competitiveness driving factor for any company. Today, many of these improvements include the integration of technology-based solutions into processes traditionally operated by humans. In this context, the present work aims to report the controller performance of a prototype developed for semi-automatic sewing stations. This project was fostered by “Factory Play”, a Portuguese company that produces inflatable structures, under the technical supervision of the Polytechnic Institute of Bragança. At the present time, the sewing station travel speed is regulated by an embedded PID controller that has been previously tuned using classical methods. However, even if the overall performance is currently acceptable, additional experiments were made regarding the use of evolutionary based algorithms to attain a better dynamic response and flexibility. This article presents the results obtained using those methods where it is possible to confirm that the use of evolutionary algorithm will simplify the design process while consistently leading to a suitable solution.

**Keywords**—Industrial automation, PID controller, parameter estimation, embedded digital control, evolutionary based algorithm.

## I. INTRODUCTION

The brand “Factory Play” is associated to a Portuguese company which, among other products, design and produces inflatable structures for recreational activities and theme parks. Nowadays, its activity extends over and above the national borders: not only inside the EU region but also taking place in several other countries scattered along the globe. Due to its increasing production demand, the manufacture methods currently employed must be improved in order to become more efficient. In particular, the method currently employed to sew the inflatables is obsolete and, according to the vision of the company’s CEO, must be updated.

Just to put it in context, an inflatable structure comprises a large number of sailcloth tiles which must be joined together by means of double, triple or even quadruple seams. At the present time, this process is carried out by human-operated industrial sewing machines that are able to move along the edges of a very large working table where the tiles to be sewn are laid down. Each sewing station is operated and propelled by a company employee using muscular power. This production method presents several problems such as the dependence of the seams spacing with the speed of the sewing

station which depends on the human operator expertise. It is desirable that the sewing station linear speed be synchronized with the seesaw movement of the sewing machine which will lead to a more uniform, and operator independent, seams. Since, due to the process idiosyncrasy, off-the-shelf solutions are not commercially available, the design and implementation of a custom made prototype is currently being developed at the Polytechnic Institute of Bragança.

A first version of the electro-mechanical prototype has already been developed and documented in [1] and is presented in Figure 1. It is worth to notice that this solution was developed without any ergonomics concern in mind. The main objective of this version is to validate the instrumentation and actuation solutions and to get intuition on the overall system dynamics involved. Of course, the employee’s health and safety are two extremely important variables in the overall equation which will be addressed in a subsequent development stage.



**Fig. 1:** The actual sewing station prototype. Further details on this setup can be found at [1].

Currently, the actuation, instrumentation and control loops are already established. When in automatic mode, the movement speed of the cart depends on the sewing machine operation where an embedded digital proportional-integral (PI) controller closes the loop between the measurement speeds and the sewing station movement. In [1], the dynamics of the prototype was obtained by means of its step response and the derived mathematical model was employed to

estimate the controller parameters. Using the system transfer function, the PI gains were adjusted using a classical Bode plot reshaping technique.

Even if the closed-loop system behaviour obtained using the above method is acceptable, it requires a *know-how* that prevent its tuning by common plant operators whenever the cart dynamics shifts from the nominal point due to changes in friction or even wear and tear. For this reason, an alternative controller design technique was explored which could be used to automate the controller parameters estimation procedure. The use of evolutionary-based techniques for control systems design are well-established methods in the literature which can be used to automate the tuning process in the present reference frame.

This paper follows from [1] and describes the steps and results obtained through the use of evolutionary-based algorithms (EA) for proportional-integral-derivative (PID) controller tuning in the context of the sewing station prototype. The article is divided into seven sections. Section II present the overall system architecture and Section III describes the system mathematical model. Section IV is devoted to describe the evolutionary-based approach to the controller design problem. Comparative results between the use of genetic (GA) and particle swarm optimisation (PSO) algorithms are presented in Section V and Section VI shows the effective results obtained after deploying the controllers in the hardware support. Finally, the last section will be devoted to present the final conclusions and the direction of future work.

## II. PROTOTYPE DESCRIPTION

In order to make this article as self-contained as possible, this section present an overall portrayal of the system to be controlled. Interested readers can find a more thorough description of the prototype at [1].

The diagram in Figure 2 represents a complete overview of the instrumentation, actuation and control loops actually implemented.

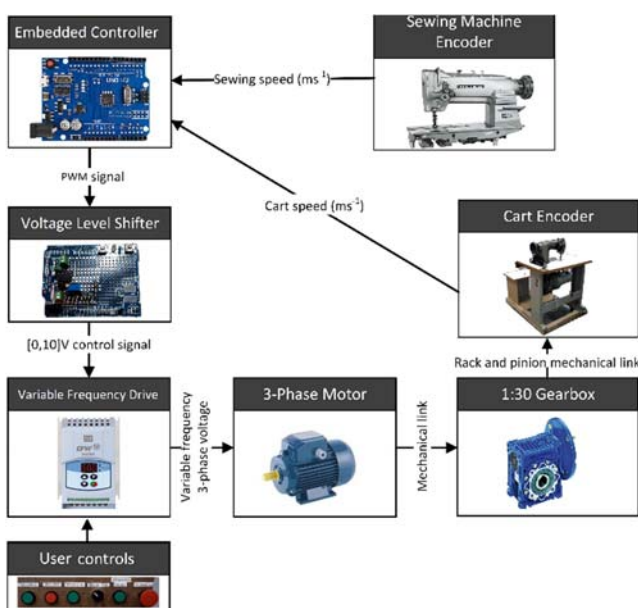


Fig. 2: The sewing station instrumentation, actuation and control loops.

The digital pulses coming from the incremental encoders, installed at the sewing machine and cart electrical motor, are read by the microcontroller. From whose signals, both the cart and sewing machine speeds are computed. The former will be the system state variable and a scaled version of the latter will be used as the set-point. This information will be fed to an embedded digital controller which will be responsible to regulate the cart speed. The digital controller computes the control signal to be applied to the analog voltage input of the motor frequency inverter. This analog signal is generated, using a pulse-width modulation (PWM) strategy and fed to the variable-frequency drive (VFD). The motor's VFD takes this input and generates a proportional voltage/frequency to the induction motor which, in turn, propels the sewing station.

The embedded digital controller plays a fundamental role in keeping the set-point tracking accuracy in the presence of several disturbances. One of such disturbances is the human operator's weight which will change after each working shift. Moreover, overshoot is not allowed and the cart speed rising-time must be kept near one second. In [1], a PI controller has found to be suitable to solve all those performance conditions. This controller was designed using a classical approach where a Bode plot reshaping technique was applied. However, reshaping the controller gains with this technique is very cumbersome and requires expertise which are not within the grasp of common technical staff. For this reason, the use of evolutionary-based algorithms was considered as tools to automate the controller design procedure. This tuning must be carried out if the system begins to exhibit performance degradation due to several factors such as aging.

Regardless of the controller design method, a mathematical model of the process to be controlled is often required. For this reason, the following section will be devoted to describe the dynamical behaviour of the system to a step input and how this information was used to derive the system's mathematical model.

## III. SYSTEM DYNAMICS

A system mathematical model is at the heart of most of the controller design techniques. Moreover, the performance of the controller deployed in the field strongly depends on this model accuracy. In the present case, this model could be obtained by *first principles* but a more empirical approach was taken. In particular, the step response of the system was measured and then fitted to canonical model structures.

Figure 3 shows the response of the sewing station cart to a step input. This data was obtained by applying a sudden change from 0 to 10V to the input of the VFD while the cart speed was measured. In fact, the data presented in the above referred plot was obtained by averaging the data acquired from ten independent experiments leading to an increased signal-to-noise ratio. It is also worth to notice that the speed axis was normalized by the open-loop steady state speed which, in this case, is equal to 0.16 m/s.

At first glance, the response seems to be consistent to the one of a first order system with a time constant of around 0.3 seconds. Using the measured time constant, and taking into consideration the canonical form of a first-order dynamical system, the transfer function becomes:

$$G(s) = \frac{3.953}{s+3.593} \quad (1)$$

where  $s$  denotes the *Laplace* operator.

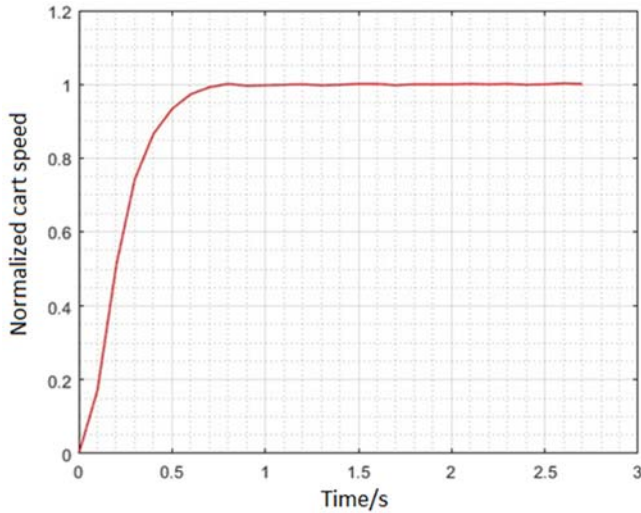


Fig. 3: Open-loop sewing station cart speed response to a step-type input signal.

The comparison between the measured step response and the one obtained by simulation is presented in Figure 4.

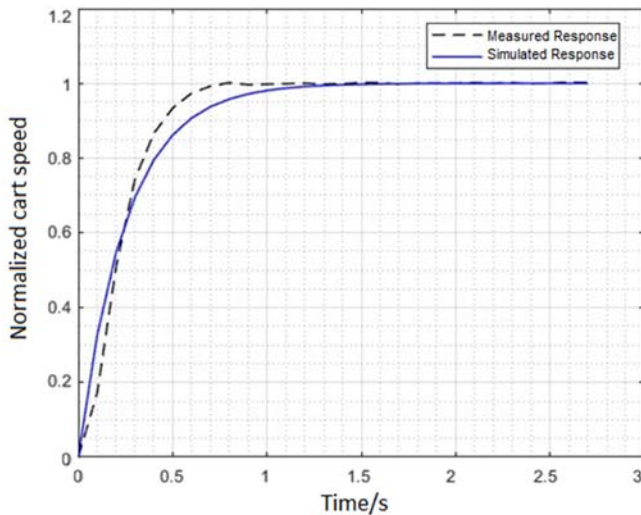


Fig. 4: Simulated and measured step response assuming a first-order dynamical model.

The relative simulation error using this model is around 16% which has been considered too high. For this reason, the model order was increased and its parameters derived from two performance indexes: the overshoot value and the peak time. The former was found to be  $t_p = 0.9$  s and the latter  $M_p = 1.0012$ .

As is well known, the canonical form of a second order system is given by:

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (2)$$

where  $\omega_n$  is the system's natural frequency and  $\zeta$  is the damping ratio. Those variables are related to  $t_p$  and  $M_p$  by:

$$\zeta = \sqrt{\frac{\ln(M_p - 1)^2}{\pi^2 \ln(M_p - 1)^2}} \quad (3)$$

$$\omega_n = \frac{\pi}{t_p \sqrt{1 - \zeta^2}} \quad (4)$$

Replacing the measured overshoot and peak time in (3) and (4), leads to  $\zeta = 0.9$  and  $\omega_n = 8.04$ . Hence, the transfer function (2) becomes:

$$G(s) = \frac{64.58}{s^2 + 14.48s + 64.58} \quad (5)$$

The measured and simulated step response, assuming the second-order transfer function presented in (5), is shown in Figure 5.

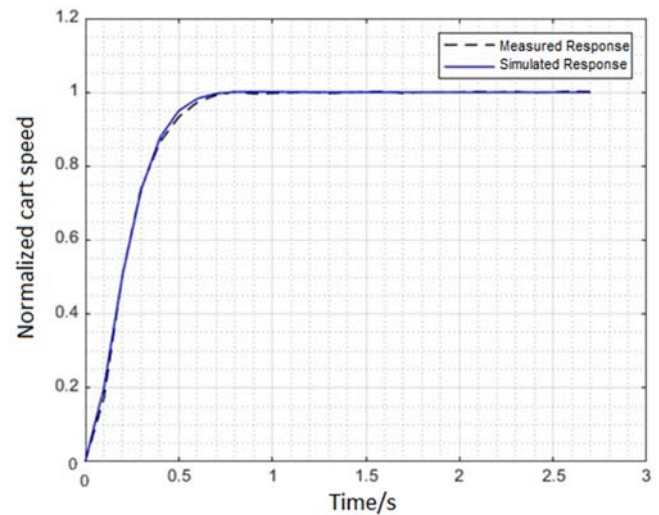


Fig. 5: Simulated and measured step response assuming a second-order dynamical model.

As can be seen, the increase model order leads to a better model accuracy. In this case, the relative error was lowered down to less than 2.5% which is within an acceptable range. For this reason, the controller design process, which will be described in the following section, will consider this higher order model.

#### IV. CONTROLLER DESIGN WITH EA

Since the late 1980's, evolutionary-based algorithms plays an important role as a numerical tool for solving different types of engineering problems. The name "evolutionary algorithms" is a generic reference to a class of stochastic optimisation methods that are commonly biologically or naturally inspired [2-3]. Nowadays, there are myriads of different types of such algorithms but genetic algorithms (GA) and particle swarm optimisation (PSO) are still two of the older and most well established methods.

GA's are a class of search technique based on natural selection mechanisms [4-5]. The natural selection theory has its foundations on the deviations between organisms of the same population. According to Darwin's theory of evolution, the population elements that best adapts to the surrounding environment will be those who's probability of surviving and procreate will be higher, passing its genetic information to their offsprings. On the other hand, PSO is a stochastic optimisation method inspired on the social behaviour of organisms [6-7]. The algorithm searches a multidimensional space by adjusting the trajectories of individuals toward the positions of their own previous best performance and the best previous performance of their neighbours. Each particle is treated as a point in a  $n$ -dimensional space, and adjusts its

dynamics according to its own experience and that of its pairs. Full enumeration of the advantages and disadvantages of stochastic search techniques over more deterministic approaches is outside the scope of the current document. Let's just say that control systems design is one of the engineering areas that have benefited with the dawn of EA. In particular, the use of such techniques for PID controllers design can be tracked down to more than twenty years ago and remain an up-to-date research subject [8-11]. PID based control loops are ubiquitous and can be found in several industrial plants and automotive systems, just to name a few. In the context of industrial sewing machines, this type of three-term controllers has already been used. For example, [12] has applied a PID strategy to the speed control of a brushless direct-current motor. As already referred in the beginning of this paper, a PI controller was already designed and deployed in the current sewing station prototype. The design method employed was based on a classical Bode diagram reshape approach. On the other hand, this paper will deal with the design of a PID controller whose parameters will be estimated using two distinct evolutionary-based algorithms according to a given fitness function. Defining the correct shape of this function is fundamental since it is the solution primary steering factor which, at the end, dictates the performance of both the controller and the search algorithm. The following subsection will be devoted to characterise the fitness function employed in the PID controller design stage.

A. The Fitness Function

The time-domain canonical form of a PID controller is given by:

$$u(t) = K_p \left( e(t) + \frac{1}{T_i} \int_{-\infty}^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right) \quad (6)$$

where  $u(t)$  is the control signal and  $e(t)$  is the error signal computed as the difference between the desired set-point and the actual measured system output. The parameters  $K_p$ ,  $T_i$  and  $T_d$  are positive real-valued controller coefficients denoted as the proportional gain, integration time and the derivative time respectively.

Designing a PID controller boils down to calculating the values of its three degrees of freedom such as to have the desired closed-loop dynamic response. When using an optimisation algorithm as the design engine, there is one major decisions that must be made which concerns on how to mathematically describe the system performance. Within a classical optimisation framework, this mathematical description is known as the objective function and usually is known under the name of fitness function when dealing with EA terminology. If  $f(\theta)$  is such a function, then the controller design problem can be equated as the following constrained optimisation problem:

$$\begin{aligned} \min_{\theta=\{K_p, T_i, T_d\}} & f(\theta) \\ \text{s.t.} & K_p \in [L_p, U_p] \\ & T_i \in [L_i, U_i] \\ & T_d \in [L_d, U_d] \end{aligned} \quad (7)$$

where  $L_p$ ,  $L_i$  and  $L_d$  concerns the lower bound and  $U_p$ ,  $U_i$  and  $U_d$  the upper bound of the decision variables  $K_p$ ,  $T_i$  and  $T_d$ .

In the current work, the fitness function  $f(\theta)$  is defined as a weighted sum of several distinct terms as can be seen below:

$$f(\theta) = [k_{os} \quad k_e \quad k_{bw} \quad k_c \quad k_{\Delta}] \begin{bmatrix} f_{os}(\theta) \\ f_e(\theta) \\ f_{bw}(\theta) \\ f_c(\theta) \\ f_{\Delta}(\theta) \end{bmatrix} \quad (7)$$

where  $k_{os}$ ,  $k_e$ ,  $k_{bw}$ ,  $k_c$  and  $k_{\Delta}$  are positive weighting coefficients and  $f_{os}(\theta)$  is a function that represent the closed-loop overshoot,  $f_e(\theta)$  denotes the mean square error,  $f_{bw}(\theta)$  depends on the closed-loop bandwidth,  $f_c(\theta)$  is the control signal maximum value and  $f_{\Delta}(\theta)$  denotes de difference between the settling time and the rise time. Further details of each function are provided in the following subsection.

B. Description of the Fitness Function Terms

There is not an exact and single figure-of-merit that can be used to describe the desired closed-loop response. From empirical tests, it was found that the system should exhibit a settling time of near one second, an overshoot close to zero, minimum steady-state error and a control signal within the dynamic range of the VFD. In this context, the fitness function (7) was devised and takes into account all the above referred conditions.

The first term of (6) will penalize the fitness function for responses with large overshoot. In particular,  $f_{os}(\theta)$  is the simulated closed-loop overshoot value obtained from a step input response. The weight associated to this component has been defined to be  $k_{os} = 0.3$ . The second term,  $f_e(\theta)$ , denotes the mean square value of the steady-state error assuming also a step input. The weighing factor used for this term was  $k_e = 100$ .

Since the desired closed-loop bandwidth must be around 2 rad/s, the penalty function  $f_{bw}(\theta)$  was added which is equal to the absolute difference between the actual simulated bandwidth and the desired value of 2 rad/s. The weight of this term, in the overall fitness function, was empirically found to be equal to  $k_{bw} = 0.001$ .

Due to the saturation phenomena caused by the VFD input limits, the controller must be able to provide a control signal that are within the VFD operating range. For this reason, a fourth term was added to (6) which penalizes the performance if the amplitude of the control signal exceeds the drive operating limits. In this case,  $f_c(\theta)$  refers to the maximum value of the control signal and its weighting coefficient was defined to be  $k_c = 1.8$ .

The fifth, and last term, aims to promote monotone responses during the transient period. Hence, the term  $f_{\Delta}(\theta)$  computed as the difference between the settling time and the rise time was added. This function has proven to mitigate the referred condition for  $k_{\Delta} = 0.5$ . Besides the fitness function definition, other operating conditions must also be defined. For example, the decision variables interval and the software package used. Both questions will be answered in the next subsection.

C. Operational Conditions

The simulations and obtained results documented in the following section, were carried out and within the MATLAB® numerical computation environment. Moreover, only *off-the-shelf* functions from its Global Optimisation Toolbox (GOT) were used. In particular, the *ga()* and *particleswarm()*.The



only custom computer code developed was the one for implementing the fitness function.

Regarding the dynamic range of the controller gains, they were coarse to be within the interval that spans from 0.001 up to 100. Section V will present the controller design results obtained by each of the two EA executed under the above referred simulation conditions.

### V. SIMULATION RESULTS

The simulation results presented in this section were carried out using the 64-bits version of MATLAB® R2018b and the version 4.0 of the GOT.

The first set of experiments concerns the use of GA as the PID controller design method. In the MATLAB® environment, the function *ga()* was executed with a population size of 100 individuals, a crossover probability of 80% and the default uniform mutation strategy.

A total of 300 generations per trial was considered with an early stopping mechanism consisting of 50 consecutive stall generations. After 100 trials, the algorithm has converged to the following average solution:  $K_p = 0.169$ ,  $T_i = 0.0721$  and  $T_d = 0.0914$ . The closed-loop step response using these gains is shown in Fig. 6.

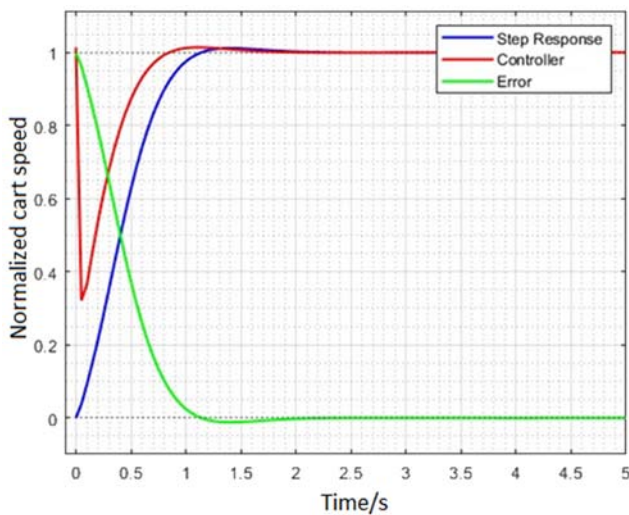


Fig. 6: Closed-loop step response simulation using the parameters found with the GA.

It can be seen that the system’s rise time is equal to 0.7 seconds, the settling time is around 1.5 seconds and the overshoot is lower than 1.5%.

Besides GA, the PSO algorithm was also considered in this work as the controller design method. The MATLAB®’s *particleswarm()* function was used for this set of tests where a population of 100 particles was considered. Both the social and self-adjustment gains were made equal to 1.49. The evolution is carried out during a maximum of 300 epochs per trial with an early stopping criteria fired if the solution stall during 50 consecutive epochs. The average results found, after the execution of 100 trials, was:  $K_p = 0.092$ ,  $T_i = 0.0231$  and  $T_d = 12.15$  and the system step response obtained using these parameters is represented on Fig. 7. In this case, the rise time is below 0.5 second and the settling time is under one second. Moreover, the overshoot observed from this simulation is zero.

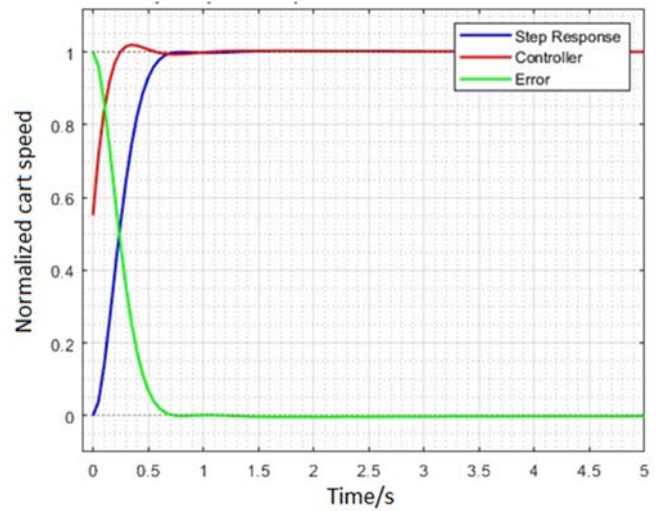


Fig. 7: Step response from the parameters found with the Particle Swarm Optimization.

From the obtained results it is possible to confirm that both the GA and the PSO can be used as PID design tools for the problem addressed in this work. Even if both solutions are acceptable in terms of overall performance, the one obtained by the PSO outperforms the solution delivered by the GA in all the referred figures-of-merit. However, it was the solution generated by the latter that was chosen to be deployed in the hardware controller. This choice was mainly due to its higher rise-time which produces lower cart accelerations and leading to a more comfortable feeling at the human operator level. The experimental results obtained from deploying this PID controller into the prototype will be presented ahead.

### VI. EXPERIMENTAL RESULTS

With the PID parameters obtained from the previous section, a digital controller was deployed into the microcontroller. Since the design was made in the continuous-time domain, a discretization step must be performed prior to its firmware implementation. Two different conversion methods were tried in order to obtain the difference equations: the trapezoidal and the bilinear methods [13]. Figure 8 show the obtained results using the former technique.

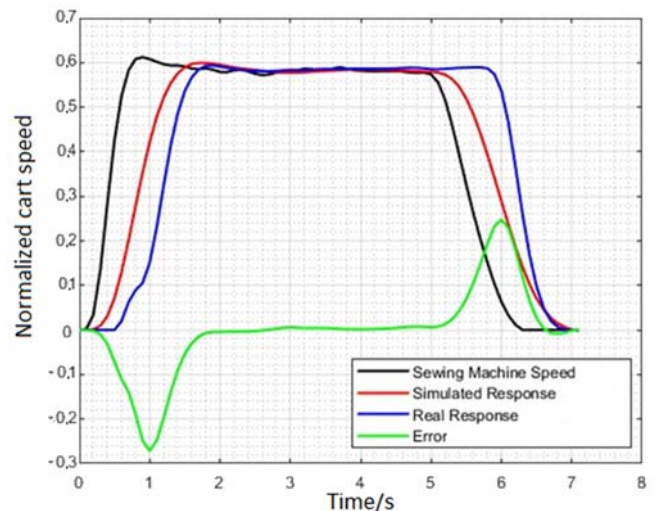


Fig. 8: Closed-loop system response using the trapezoidal approximation as the controller discretization technique. The blue line concerns the measured speed, the red line the simulation and the black line is the set-point.

This plot concerns the closed-loop system response to a scaled version of the sewing machine speed. This signal is the set-point input and is represented by the black line in Figure 8. In the same plot, it is possible to observe the actual measured cart speed represented by the blue line. Moreover, an off-line simulation, using the mathematical model described in Section III, was also carried out and the results plotted using the red colour. The mismatch between the measured and simulated response is represented in green.

The second discretization method was explored with the aim of improving the matching between the actual and simulated dynamic response. Figure 9 present the results obtained when using the bilinear discretization method.

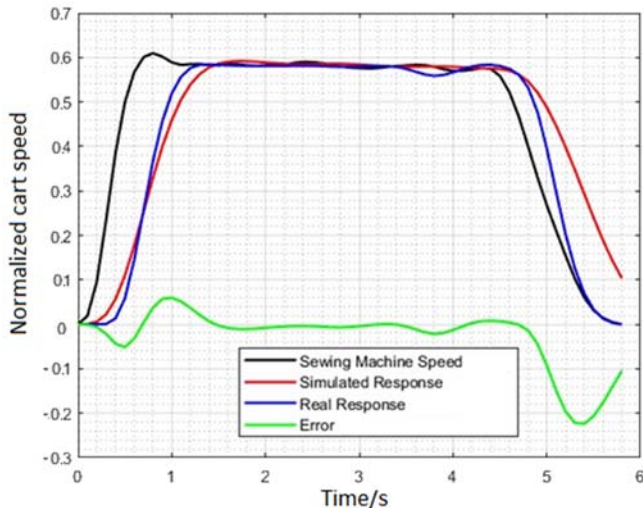


Fig. 9: Closed-loop system response using the bilinear approximation as the controller discretization technique. The blue line concerns the measured speed, the red line the simulation and the black line is the set-point.

Comparing both results, it is possible to conclude that, while the bilinear discretization method leads to a controller with better transient response, it will remain more oscillatory during steady-state than the response obtained using the trapezoidal approximation.

From the human operator point-of-view, the oscillation amplitude observed, when using the bilinear discretization technique, are unnoticed. And since it was the solution that exhibit better dynamic response, the bilinear discretization method will be the proper choice if, in the future, this continuous-to-digital translation remains a necessary step.

## VII. CONCLUSIONS AND FURTHER WORK

This paper was devoted to document the advances made in the development of a sewing station prototype to be used at the “Factory Play” company. In particular, it deals with the evaluation of EA as suitable tools for PID controller design. Due to its mathematical simplicity and parallel nature, those algorithms are easily translated into hardware and may become part of an auto-tuning strategy.

From the obtained results it is possible to conclude that, indeed, EA can be used as a controller design tool with the advantage of being easily translated to embedded systems.

Additionally, they can provide an automated method to perform the controller tuning without any particular technical expertise.

However, before becoming an integrated solution in the prototype, there are some details that must be tackled. First, EA are conceptually simple but requires a large computation framework to speed up its execution. We believe that the use of field programmable gate arrays (FPGA) can overcome this problem [14]. Moreover, the non-convexity of the actual fitness function, must be handled in order to improve the EA convergence to a systematic solution.

It was also notice that the controller performance is uneven in all operating regimes. This phenomenon can be explained by the non-linearity behaviour caused by the actual motor drive installed. Hence, an alternative VFD with vector control capabilities is being considered.

## REFERENCES

- [1] J. P. Coelho, P. Santos, T. M. Pinho, J. Boaventura-Cunha, and J. Oliveira, “Instrumentation and control of an industrial sewing station”, 2018 13th APCA International Conference on Control and Soft Computing, doi: 10.1109/controlo.2018.8514280.
- [2] M. Gen and R. Cheng, Genetic algorithms and engineering optimization. John Wiley & Sons, Inc., 2000
- [3] D. Floreano and C. Mattiussi, Bio-Inspired Artificial Intelligence: Theories, Methods and Technologies, 1st. Cambridge, MA, U.S.A.: The MIT Press, 2008, ISBN:9780262062718
- [4] Goldberg, D. E. (1989). Genetic Algorithms in search, optimization and machine learning, Addison-Wesley.
- [5] Holland, J.H. (1975). *Adaptation in natural artificial systems*, Ann Arbor, MI: University of Michigan Press.
- [6] Kennedy, J. and R.C. Eberhart (1995). Particle Swarm Optimization, *Proc. Of the 1995 IEEE Int. Conf. On Neural Networks*, pp. 1942-1948. IEEE Service Center, Piscataway, NJ.
- [7] K. E. Parsopoulos and M. N. Vrahatis, Particle Swarm Optimization and Intelligence: Advances and Applications, 1st. Hershey, PA, U.S.A.: Information Science Reference, IGI Global, 2010
- [8] De Moura Oliveira, P.B. and A. H. Jones (1998). Genetic Design of Two Degrees-of-Freedom PID Controllers for Set-Point Tracking and Disturbance Rejection, *Controlo '98*, pp. 111-118, Coimbra, Portugal.
- [9] Chou P., Hwang T. (2004) Design of PID Controllers Using Genetic Algorithms Approach for Low Damping, Slow Response Plants. In: Yin FL., Wang J., Guo C. (eds) *Advances in Neural Networks. Lecture Notes in Computer Science*, vol 3174. Springer, Berlin, Heidelberg
- [10] X. Meng B. Song (2007), Fast Genetic Algorithms Used for PID Parameter Optimization, *Proceedings of the IEEE International Conference on Automation and Logistics*, Jinan, China.
- [11] Perng, J-Wi and Hsieh, S-C (2019), Design of Digital PID Control Systems Based on Sensitivity Analysis and Genetic Algorithms, *International Journal of Control, Automation and Systems*, vol 17, 4, p.p. 1838—1846.
- [12] Yang Luo, Hui Li, and Mingyong Shen (2006). Speed control of BLDCM for industrial sewing machine based on dSPACE. In 2006 International Conference on Mechatronics and Automation. IEEE.
- [13] C. H. Houpis and G. B. Lamont, Digital Control Systems: Theory, Hardware, Software, 2nd. The McGraw-Hill Companies, Inc., 1992, ISBN: 0070305005.
- [14] J. V. József, D. Drótos, J. Turán, J. Végh (2012). Processors, FPGAs, SOCs, trends and questions. *Carpathian Journal of Electronic and Computer Engineering* 5, 149-152.