Genetic Algorithm for Decentralized PI Controller Tuning of a Multi-Span Web Transport System Based on Overlapping Decomposition

Genetic algorithm for decentralized PI controller tuning of a multi-span web transport system based on overlapping decomposition

Nicola Ivan Giannoccaro[1], Giancarlo Manieri[2], Paolo Martina[3] and Tetsuzo Sakamoto[4]

***Abstract*— The use of multi-span web transport systems often requires dedicating a particular effort for defining a control system able to protect the integrity of the web. In web handling processes there is a clear need for controlling web velocity and tension. As for any control problem, the best results are achieved when there is a clear understanding of the controlled process. Starting from a new mathematical model called *Transfer Matrix Model* for a more accurate description of a web transport system, an optimization study for identifying parameters of the PI controllers for a decentralized multi-input and multi-output system (MIMO) has been proposed. The possibility of using PI controllers for each section is attractive considering that an overlapping system decomposition may permit to take into account the mutual interaction of the neighbor sections reaching specified performance objectives. Finally, by using a nonlinear interpolation of the trend of a preliminary database of values obtained through genetic algorithm, a self-tuning strategy is proposed to estimate optimal PI parameters under certain conditions, avoiding the long identification process and making the system flexible and adaptive. Simulations and experimental results validate and illustrate the effectiveness and the simplicity of the proposed method by considering several different set-points.**

I. INTRODUCTION

As a real industrial example of large-scale systems, large-scale web transport system with many different processes has been studied in the decentralized control framework. The presence of tension terms in the roller velocity dynamics and, conversely, roller velocity terms in the tension dynamics lead the web transport system to be an interacting large-scale system, as demonstrated in [9]. Given measurements of all states variables, the system can be controlled by multi-variable control methods. Numerous attempts have been presented with promising results. Though powerful, multi-variable control has its limitations. Being centralized, the control scheme must be completely redesigned if the system is changed in some way. For example, adding one process to the system may force the system to be redesigned since the system dimension has been grown by one. Furthermore, failure in a section of one web tension zone can lead to catastrophic failure in the overall control system. Decentralized structure can alleviate these problems associated with centralized control structure. Although widely applicable to the industry, traditional decentralized control structure also introduces other issues. In the decentralized control case, the interconnections between segments are usually neglected for control design purposes. However, the interconnections are affecting the subsystems. Furthermore, an extra degree of freedom that models the dynamics is added in the overlapping decentralized framework, which adds to the complexity of implementation. The aim of this work is to provide a novel approach for PI controllers parameters identification by means of genetic algorithm based on overlapping decomposition and to propose, by using a nonlinear interpolation of the trend of a preliminary database of values obtained through the same genetic algorithm, a self-tuning strategy to estimate optimal PI parameters under certain conditions and experimental tests, avoiding the long identification process and making the system flexible and adaptive.

[1,2,3] are with Department of Innovation Engineering, University of Salento, Lecce, Italy. Email: ivan.giannoccaro@unisalento.it; Tel. +390832297813

[4] is with Department of Control Engineering, Kyushu Institute of Technology, Tobata, Kitakyushu, Japan. Email: sakamoto@cntl.kyutech.ac.jp

II. THE CONSIDERED WEB TRANSPORT PLATFORM

The realized system, located in Sakamoto Laboratory at *Kyushu Institute of Technology - Kitakyushu, Fukuoka - JAPAN*, already introduced in [1] [2] is composed of four main sections strongly interlaced each other and 12 rolls placed on a mechanical frame at different heights, realized in order to represent a large transport system similar to many industrial ones. The system has been completely renewed at the end of 2015, substituting all the rolls and their bearings with new ones with high performances (low weight and low friction). The new system scheme is depicted in Fig. 1: the transport system is driven by 4 servomotors, the first is referred to as the *unwinder section*, the second to as the *lead section*, the third to as the *draw-roll section*, the last to as the *winder section*.

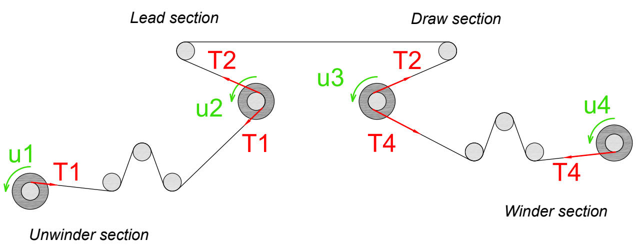


Fig. 1: Scheme of the web transport system

The mathematical model of the web transport systems is based on three laws, in the Laplace domain, applied at each section between two consecutive rolls [2]:

- Conservation of mass
- Torque balance
- Voigt-Model

$$\varepsilon_k(s) = \frac{1}{L_k \cdot s} \cdot [v_{k+1}(s) - v_k(s)] \tag{1}$$

$$s\,J_k\,\omega_k(s) = r_k\,[T_{k+1}(s) - T_k(s)] + u_k(s) - C_k(s) - k_{fk}\,\omega_k(s) \tag{2}$$

$$T_k(s) = P(s) \cdot s \cdot \varepsilon_k(s) \tag{3}$$

III. Overlapping decomposition

Since 1998 it was demonstrated [3] that a decentralized controller based on overlapping decomposition permits considering some mutual interactions between subsystems. The resultant control system has better control performance compared with a decentralized controller based on disjoint decomposition and it makes the controller design simpler. Following this approach, the realized web handling system (Fig. 2a) has been divided in 4 overlapped sections (Fig. 2b) and the new input controls (Fig. 2c) are calculated through the N^{-1} matrix.

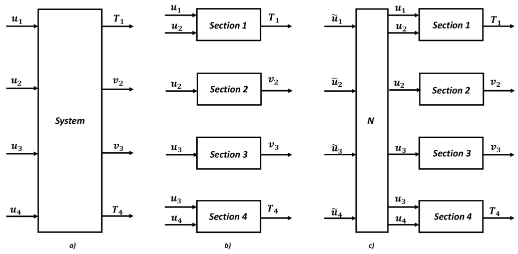


Fig. 2: Overlapped subsystems decomposition

Equations (1), and (3) have been considered to obtain (4) and (5) for T_1 and T_4 respectively. Equations of speeds have been written without the dissipative terms, C_k and k_{fk}, thus obtaining a simpler form for web speed outputs (6), (7), (8) and (9):

$$T_1(s) = \frac{P(s)}{L_1}[v_2(s) - v_1(s)] \tag{4}$$

$$T_4(s) = \frac{P(s)}{L_3}[v_4(s) - v_3(s)] \tag{5}$$

$$v_1(s) = \frac{-u_1\,r_1 + T_1(s)\,r_1^2}{s\,J_1} \tag{6}$$

$$v_2(s) = \frac{u_2\,r_2 - T_1(s)\,r_2^2 + T_2(s)\,r_2^2}{s\,J_2} \tag{7}$$

$$v_3(s) = \frac{u_3\,r_3 - T_2(s)\,r_3^2 + T_4(s)\,r_3^2}{s\,J_3} \tag{8}$$

$$v_4(s) = \frac{u_4\,r_4 - T_4(s)\,r_4^2}{s\,J_4} \tag{9}$$

Before proceeding with mathematical replacements, it was necessary to further simplify each expression. Overlapping decomposition, in fact, takes into account a part of all the interactions between two neighboring subsystems. Each

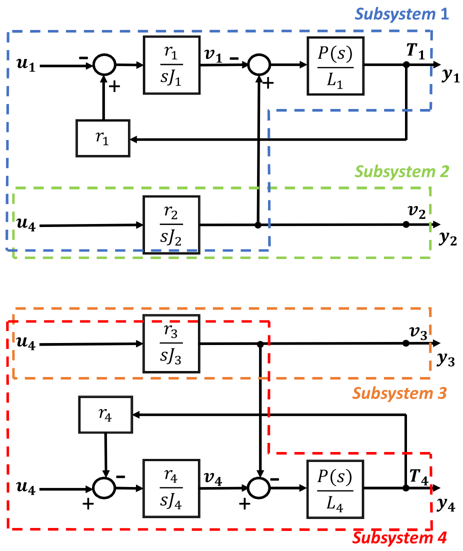


Fig. 3: System model scheme with the considered overlapping decomposition

subsystem output considers only the information contained in the colored boxes (Fig. 3).

Following this, replacing both (6) and (7) in (4) and repeating the same procedure for both (8) and (9) in (5) it was obtained:

$$T_1(s) = \frac{\frac{P(s)}{L_1}}{s + \frac{P(s)}{L_1}\frac{r_1^2}{J_1}}\left(\frac{r_2}{J_2}u_2 + \frac{r_1}{J_1}u_1\right) \tag{10}$$

$$v_2(s) = \frac{r_2}{J_2\,s}u_2 \tag{11}$$

$$v_3(s) = \frac{r_3}{J_3\,s}u_3 \tag{12}$$

$$T_4(s) = \frac{\frac{P(s)}{L_3}}{s + \frac{P(s)}{L_1}\frac{r_4^2}{J_4}}\left(\frac{r_4}{J_4}u_4 + \frac{r_3}{J_3}u_3\right) \tag{13}$$

By imposing:

$$\tilde{u_1}(s) = \frac{r_2}{J_2}u_2 + \frac{r_1}{J_1}u_1 \tag{14}$$

$$\tilde{u_2}(s) = \frac{r_2}{J_2\,s}u_2 \tag{15}$$

$$\tilde{u_3}(s) = \frac{r_3}{J_3\,s}u_3 \tag{16}$$

$$\tilde{u_4}(s) = \frac{r_4}{J_4}u_4 - \frac{r_3}{J_3}u_3 \tag{17}$$

the N^{-1} matrix was obtained as follow:

$$[N]^{-1} = \begin{bmatrix} \frac{r_1}{J_1} & \frac{r_2}{J_2} & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{r_3}{J_3} & \frac{r_4}{J_4} \end{bmatrix} \tag{18}$$

This matrix allows to switch from original input vector $u = [u_1, u_2, u_3, u_4]$ to new control inputs vector $\tilde{u} = [\tilde{u_1}, \tilde{u_2}, \tilde{u_3}, \tilde{u_4}]$:

$$\{\tilde{u}\} = \left[N^{-1}\right]\{u\} \tag{19}$$

IV. Genetic Algorithm

Genetic Algorithms (GAs) are global search methods that are based on natural population genetics. Studies by Goldberg [4], Mitchell [5], and others have demonstrated, both theoretically and experimentally, the superior performance of GAs over traditional optimization technique. It is evident that, as a natural means for optimization, the genetic algorithm approach fits well within the scope of control system design and system identification where noisy, highly nonlinear, multi-modal, and discontinuous functions of many dimensions need to be optimized GAs have already been employed for the tuning of PID controllers for SISO systems [6], and Vlachos , Williams and Gomm [7] have proposed a solution using genetically tuned, multi-loop PID controllers. In this work, genetic algorithm has been applied for the tuning of multi-loop PI controllers. Since these are parametric controllers, the tuning problem can be transformed into an optimization problem with the function to be optimized having *2n* parameters, where *n* is the number of PI loops in the closed-loop, multi variable system. The highly nonlinear and multi-modal nature of this function and the lack of derivatives, mainly owing to the presence of noise and other uncertain elements in the system, motivates the use of genetic algorithm in this optimization problem. A major advantage of the proposed genetic algorithm-based tuning method is that the optimality criteria can be explicitly defined by the designer in the time domain, in terms of the desired transient responses of all closed-loop system outputs under different, user-defined set point patterns, including loop-coupling specifications. This makes the method applicable to many complex multi-variable control problems. Furthermore, although only PI controllers are considered, the generality and open architecture of the proposed method makes it suitable for the automatic tuning of different parametric controllers, both linear and nonlinear.

A. Blocks definition

In a *n x n* multi-variable process the conventional PI controller configuration would be one in which *n* PI controllers would be used in the *n* loops associated with the process, as shown in Fig. 4.

Having the *Transfer Matrix Model* [9], starting from the proposed approach in [8], a parametrized optimization problem was set on k_p and k_i parameters of PI controller of each subsystem. The controller parameters have been arranged in a diagonal matrix, according to a decentralized structure:

$$\left[D(s)\right] = \begin{bmatrix} k_{p1}+\frac{k_{i1}}{s} & 0 & 0 & 0 \\ 0 & k_{p2}+\frac{k_{i2}}{s} & 0 & 0 \\ 0 & 0 & k_{p3}+\frac{k_{i3}}{s} & 0 \\ 0 & 0 & 0 & k_{p4}+\frac{k_{i4}}{s} \end{bmatrix}$$

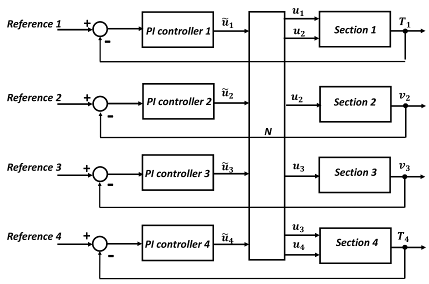


Fig. 4: Control scheme of the 4 overlapped subsystems

Aware of the presence of overlapping decomposition, the transfer matrix describing the system was also made diagonal neglecting mutual interactions between each subsystem, at first approximation, in order to lighten the very high computational load required:

$$\left[H(s)\right] = \begin{bmatrix} H_{11} & 0 & 0 & 0 \\ 0 & H_{22} & 0 & 0 \\ 0 & 0 & H_{33} & 0 \\ 0 & 0 & 0 & H_{44} \end{bmatrix}$$

Starting from these two matrices, the closed-loop matrix was calculated, having the following form:

$$\left[F(s)\right] = \frac{\left[D(s)\right]\left[H(s)\right]}{\left[I\right] + \left[D(s)\right]\left[H(s)\right]}$$

$$\left[F(s)\right] = \begin{bmatrix} F_{11} & 0 & 0 & 0 \\ 0 & F_{22} & 0 & 0 \\ 0 & 0 & F_{33} & 0 \\ 0 & 0 & 0 & F_{44} \end{bmatrix}$$

So the system can be thought of as a single closed loop transfer matrix between design input $\{r(s)\}$ and output $\{y(s)\}$

$$\{y(s)\} = \left[F(s)\right]\{r(s)\} \tag{20}$$

where:

$$\{y(s)\} = \begin{Bmatrix} T_1 \\ v_2 \\ v_3 \\ T_4 \end{Bmatrix} \qquad \{r(s)\} = \begin{Bmatrix} T_{d1} \\ v_{d2} \\ v_{d3} \\ T_{d4} \end{Bmatrix}$$

Each term F_{11}, F_{22}, F_{33}, F_{44} is parameterized on the k_p and k_i parameters of each controller.

B. Objective function formulation

In a standard GA the objective function can be thought of as the environment in which the evolution takes place. Every individual in a GA is left to evolve and adapt to the environment (i.e. to optimize the objective function) through standard genetic operators such as reproduction, crossover and mutation. In the case of multi-loop PI tuning problem, the objective function can be based on some specified performance objectives or constraints. From a control point of view, the performance objectives are often associated with the shape of the closed loop system transient response to some input signal such as a step input. Standard response characteristics such as *peak overshoot*, *settling time*, *rise time*, *steady-state offset*, *velocity lag*, etc. may be used to specify a required controller performance. With reference to the method proposed in [8], the following objective function has been set, particularized for the output of each subsystem and for several set-points:

$$J(k_p,k_i) = \int_0^{t_1} [max\{(y_1(t)-u_b),0\}+max\{(0-y_1(t)),0\}]dt + \int_{t_1}^{t_2} [max\{(y_2(t)-u_b),0\}+max\{(l_b-y_2(t)),0\}]dt \tag{21}$$

where, for example, in the case of tension T_1 with setpoint of 10N:

- u_b is the upper boundary set, in this case, equal to 10N.
- l_b is the lower boundary, particularly important in the steady-state, set to a value of 1% less than u_b.
- t_1 is the time interval [0s : 2s].
- t_2 is the time interval [2.01s : 12s].
- $y_1(t)$ is the step response in closed-loop, with step amplitude of 10 N in the time interval t_1.
- $y_2(t)$ is the step response in closed-loop, with step amplitude of 10 N in the time interval t_2

By analyzing the formulation of the objective function it is possible to note that the duration of the simulation has been set to a value of 12 seconds.

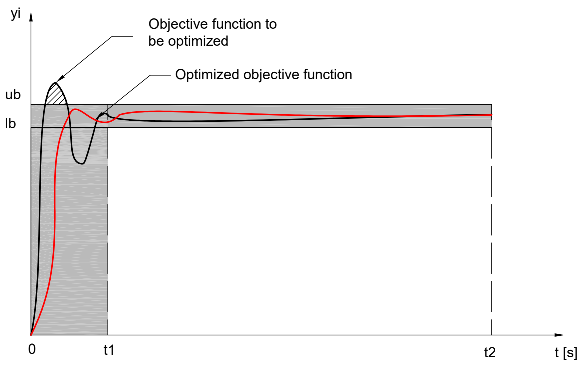


Fig. 5: Typical set-point tracking

As shown in Fig. 5, the basic idea is to minimize the areas formed by the parts of the response curves that do not belong to the shaded regions defined by the upper boundary u_b and the lower boundary l_b, and by the time interval t_1 and t_2. In particular, the first integral was defined to handle the response behaviour in the transient state: it has been imposed that the amplitude of the step response does not exceed the upper boundary u_b in the transient-state; this has allowed control of the overshoot and, by choosing the value of t_1, it is possible to define the time interval in which the transient runs out. The second integral handles what happens in the steady state: it has been imposed that the step response may oscillate, at most, in a band defined by u_b and l_b that is the response can reach a value of at most 1% lower than the set point value.

C. GA configuration

With reference to [8], the binary alphabet and the Gray coding was used for the encoding of the controller parameters because it has been shown to usually provide a more accurate solution than other conventional codings, such as direct binary coding. A simple genetic algorithm was used with the following characteristics: population size was chosen to be *80*, creation function wa chosen to be *Uniform*, scaling function was chosen to be *Rank*, selection function was chosen to be *Stochastic Uniform*, crossover fraction equal to *0.45* and *Single Point* crossover function, *Adaptive Feasible* mutation function, generation equal to *50*, function tolerance equal to 10^{-3} and bounds of the solutions set to *0-2*.

D. Optimization results

After defining the function objective and having appropriately configured the genetic algorithm, several optimizations have been conducted from different web tension and speed conditions. It is interesting to note that the genetic algorithm converges already after 9 generations with a residual equal to 0. For each set of input design, it has possible to obtain the full set of controller parameters as shown in Tables I, II, III, IV, with a total of 16 input set-point combinations.

TABLE I: Parameters of the controllers identified by GA

Tension **Speed**	10 N 0.5 m/s	10 N 0.6 m/s	10 N 0.7 m/s	10 N 0.8 m/s
k_{p1} k_{i1}	0.823 0.628	1.967 1.333	1.483 0.901	1.606 0.808
k_{p2} k_{i2}	1.760 1.098	1.927 0.985	1.807 0.785	1.869 0.674
k_{p3} k_{i3}	1.963 1.206	1.948 1.031	1.742 0.771	1.744 0.637
k_{p4} k_{i4}	0.397 0.854	0.014 1.426	0.630 1.643	1.813 1.183

TABLE II: Parameters of the controllers identified by GA

Tension **Speed**	11 N 0.5 m/s	11 N 0.6 m/s	11 N 0.7 m/s	11 N 0.8 m/s
k_{p1} k_{i1}	1.049 0.700	1.689 1.010	1.655 0.915	1.794 1.011
k_{p2} k_{i2}	1.760 1.098	1.927 0.985	1.807 0.785	1.869 0.674
k_{p3} k_{i3}	1.963 1.206	1.948 1.031	1.742 0.771	1.744 0.637
k_{p4} k_{i4}	0.269 0.808	0.532 1.449	1.384 1.427	1.095 1.220

E. Experimental and analytical results

Starting from simulation results by means of the *Transfer Matrix Model* by imposing, as controller parameters, the values obtained with the optimization based on the genetic algorithm (Tables I, II, III, IV), this section presents a

TABLE III: Parameters of the controllers identified by GA

Tension **Speed**	12 N 0.5 m/s	12 N 0.6 m/s	12 N 0.7 m/s	12 N 0.8 m/s
k_{p1} k_{i1}	1.504 1.137	1.940 0.627	1.596 0.753	1.894 1.067
k_{p2} k_{i2}	1.760 1.098	1.927 0.985	1.807 0.785	1.869 0.674
k_{p3} k_{i3}	1.963 1.206	1.948 1.031	1.742 0.771	1.744 0.637
k_{p4} k_{i4}	0.276 1.722	1.103 1.229	1.125 1.734	1.109 1.235

TABLE IV: Parameters of the controllers identified by GA

Tension **Speed**	13 N 0.5 m/s	13 N 0.6 m/s	13 N 0.7 m/s	13 N 0.8 m/s
k_{p1} k_{i1}	1.151 0.874	1.381 .717	1.511 0.768	1.903 0.869
k_{p2} k_{i2}	1.760 1.098	1.927 0.985	1.807 0.785	1.869 0.674
k_{p3} k_{i3}	1.963 1.206	1.948 1.031	1.742 0.771	1.744 0.637
k_{p4} k_{i4}	0.708 0.987	1.114 1.241	0.534 1.077	1.502 1.227

comparison, for each set-point vector input, between the result of the single step experimental test and the set-point value. The set-point input vectors are as follows:

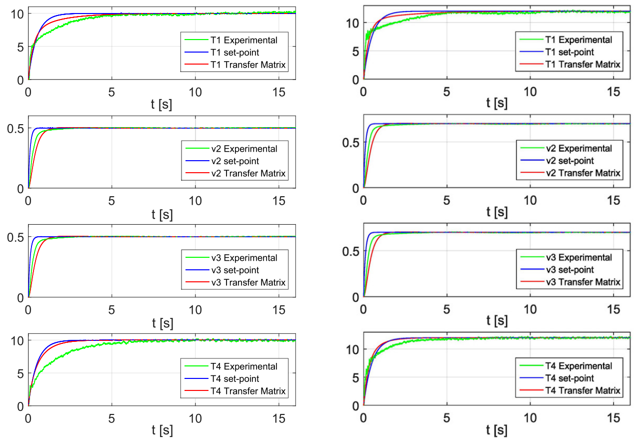
- $[10\,\mathrm{N}, 0.5\,\mathrm{m/s}, 0.5\,\mathrm{m/s}, 10\,\mathrm{N}]$
- $[11\,\mathrm{N}, 0.6\,\mathrm{m/s}, 0.6\,\mathrm{m/s}, 11\,\mathrm{N}]$
- $[12\,\mathrm{N}, 0.7\,\mathrm{m/s}, 0.7\,\mathrm{m/s}, 12\,\mathrm{N}]$
- $[13\,\mathrm{N}, 0.8\,\mathrm{m/s}, 0.8\,\mathrm{m/s}, 13\,\mathrm{N}]$

Analyzing Fig. 6 it is possible to note that the optimization carried out by generic algorithm has produced very positive results. Do not forget that, optimization, has been conducted under the basic hypotheses that have simplified the problem, by departing from its actual behaviour. It is good to remember, first of all, that decentralized control has been used, which in itself is a strong simplification. The subsequent adoption of the overlapping decomposition has allowed to consider a part of mutual interaction, making the problem a bit more truthful. The transfer matrix, that mathematically describes the system, has also been simplified and set diagonally. Despite all these simplified hypotheses, the trend of the steady state response is unquestionable. There is a perfect coincidence between the result of the simulations, the experimental result and the desired set-point value.

The non-perfect match in the transient-state results from how optimization has been set. The first integral within the objective function in (21), in fact, as it has been defined, allows the response of the system to evolve without following a precise shape but respecting the imposed boundaries.

F. Results without optimization process

This section highlights how it is necessary to identify the controller parameters when changing the set-point value both in terms of tension and speed. Starting from the parameters optimized for the vector $[10\,\mathrm{N}, 0.7\,\mathrm{m/s}, 0.7\,\mathrm{m/s}, 10\,\mathrm{N}]$, these have been used for several tests in which voltage and speed



(a) 10 N, 0.5 m/s, 0.5 m/s, 10 N (b) 12 N, 0.7 m/s, 0.7 m/s, 12 N

Fig. 6: Results

set-point values have been changed, first individually and then simultaneously.

In Fig. 7 a test with the following set-point values, $[13\,\mathrm{N}, 0.8\,\mathrm{m/s}, 0.8\,\mathrm{m/s}, 13\,\mathrm{N}]$, was performed.

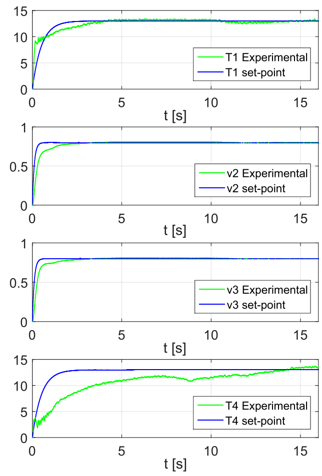
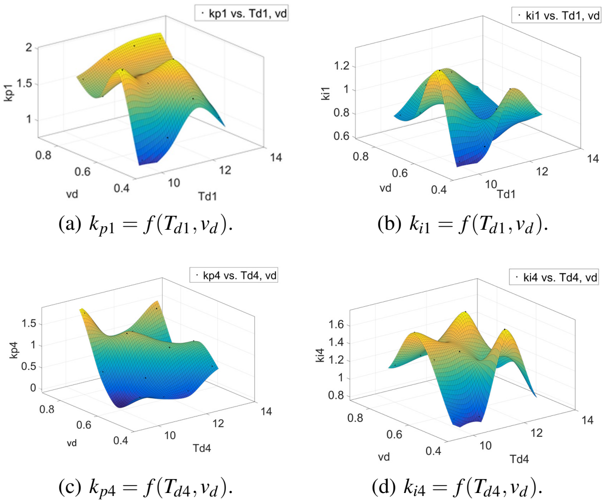


Fig. 7: Test with set-point input vector: $[13\,\mathrm{N}, 0.8\,\mathrm{m/s}, 0.8\,\mathrm{m/s}, 13\,\mathrm{N}]$

It is possible to note that both the set-point tension and speed value have been changed.

V. Interpolating Surface Approach

The genetic algorithm for searching the parameters of PI controllers has provided very satisfactory results; it resulted to be a functional approach but at the same time delicate in both setting and management. In order to make web platform control more versatile and immediate, a relationship between the gains of the controllers and both design speed v_d and design tension T_{di} has been researched. Based on these considerations, starting from the database of points identified by the genetic algorithm in Tables I, II, III, IV, third degree interpolating surfaces and curves were defined (Figs. 8, 9).



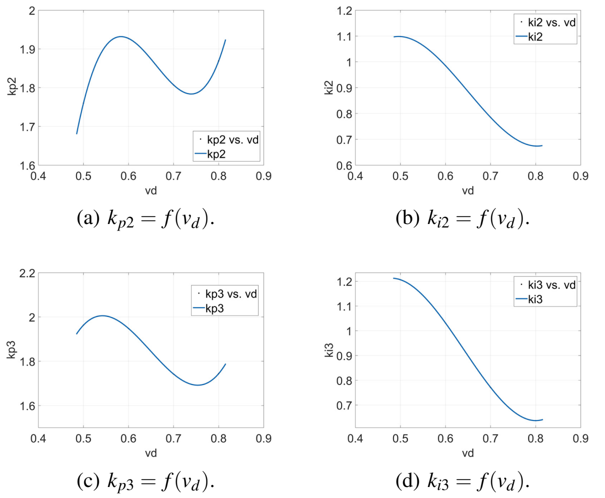
(a) $k_{p1} = f(T_{d1}, v_d)$.

(b) $k_{i1} = f(T_{d1}, v_d)$.

(c) $k_{p4} = f(T_{d4}, v_d)$.

(d) $k_{i4} = f(T_{d4}, v_d)$.

Fig. 8: Gains of controller 1 and 4.



(a) $k_{p2} = f(v_d)$.

(b) $k_{i2} = f(v_d)$.

(c) $k_{p3} = f(v_d)$.

(d) $k_{i3} = f(v_d)$.

Fig. 9: Gains of controller 2 and 3.

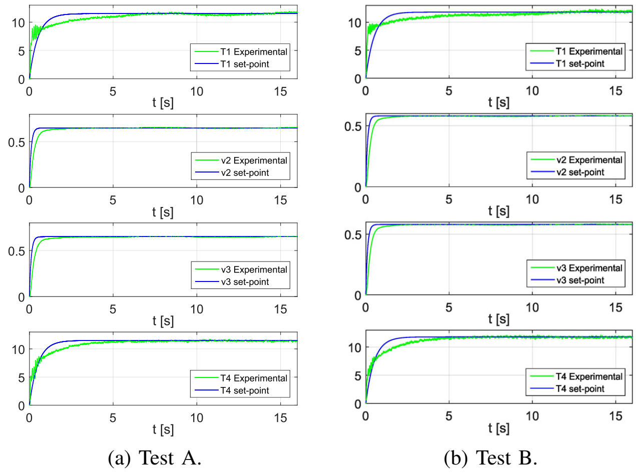
A. Results using interpolated values

Thanks to the interpolating surfaces and curves it is possible to obtain PI gains for whatever reference input, without using genetic algorithm. After selecting the set point values of tension and speed, it is only necessary to read from the plots the corresponding k_p and k_i values for each controller. Fig. 10 highlights the goodness of this strategy: the results are very good in both transient-state and steady-state. The set-point input vector is as follows:

- A=[11.5 N, 0.65 m/s, 0.65 m/s, 11.5 N]
- B=[11.8 N, 0.58 m/s, 0.58 m/s, 11.8 N]

VI. CONCLUSIONS

A novel method for the tuning of decentralized PI controllers for multivariable processes through genetic algorithm, based on overlapping decomposition has been presented. The powerful capabilities of genetic algorithms in locating the optimal or near-optimal solutions to a given optimization problem have been exploited by determining the parameters of the PI controllers to meet specified performance objectives. The designer has the freedom to explicitly specify the required performance objectives for a given problem in terms of time-domain bounds on the closed-loop responses. Each set of controller parameters has been evaluated, firstly by simulating the closed-loop system and then by using it directly on the platform for experimental tests. Simulations and experimental results have illustrated the effectiveness of the proposed method. A strategy for defining the parameters of the controllers has also been proposed by using interpolating curves and surfaces a database of points obtained by the genetic algorithm without always resorting to optimization processes. This made it possible to find a correlation between the gains of the controllers and the values of tension and speed set-point, making it much faster and much simpler, as demonstrated by experimental tests.



(a) Test A.

(b) Test B.

Fig. 10: Test using interpolated values of PI gains

REFERENCES

[1] N.I. Giannoccaro, T. Sakamoto, and I. Uchitomi. *"Performance of a web transport system with tension control subsystem using speed as control input."* In: 41st Annual Conference of the IEEE Industrial Electronics Society (IECON) (2015), pp. 1-6.

[2] N.I. Giannoccaro, A. Messina, and T. Sakamoto. *"Updating of a lumped model for an experimental web tension control system using a multivariable optimization method"*. In: Applied Mathematical Modelling, Elsevier Science 324 (3) (2010), pp. 671-683.

[3] T. Sakamoto. *"PI Control of web tension control system based on overlapping decomposition"*. In: IEEE Nordic Workshop on Power and Industrial Electronics (1998).

[4] D. E. Goldberg. *"Genetic algorithms in search, optimization and machine learning"*. In: (Addison-Wesley 1989).

[5] M. Mitchell. *"An introduction to genetich algorithms"*. In: (MIT Press, 1996).

[6] P. Wang and D. P. Kwok. *"Optimal design of PID process controllers based on genetic algotirhms"*. In: Control Eng. Prac. 2 (1994).

[7] C. Vlachos, D. Williams, and J. B. Gomm. *"Solution to the Shell Standard Control Problem using genetic algorithm"*. In: Proceedings of the UKACC internat. conf. on Control UK. 1998, pp. 1587-1592.

[8] C. Vlachos, D. Williams, and J. B. Gomm. *"Genetic approach to decentralized PI controller tuning for multivariable processes"*. In: IEEE Proc.-Control Theory Appl. 146 (1999).

[9] N. I. Giannoccaro, G. Manieri, P. Martina, T. Sakamoto. *"Implementation and validation of a lumped model for an experimental multi-span web transport system"*. In Australian and New Zealand Control Conference - ANZCC 2017, Gold Coast,17-20 December 2017.