

E. Coli Bacterial Foraging Algorithm Applied to Pressure Reducing Valves Control

Eduardo Ramírez-Llanos and Nicanor Quijano

Abstract—Continuous changes in the demand and valve operation in each district metered area (DMA) introduce undesired behaviors in a water distribution system (WDS). In order to dynamically correct the emergence of these undesired pressure transients, an adaptive controller using foraging theory as an optimization method is developed. Specifically, an E. Coli bacterial algorithm is customized to be used in an adaptive control scheme (indirect and direct) with real time requirements.

Index Terms—Pressure Reducing Valves, Adaptive Control, Water Distribution System, System Identification.

I. INTRODUCTION

LOSSES by leakage in a Water Distribution System (WDS) have been one of the most important problems that needs to be solved by the distribution companies. Some of them report almost 40% of water loss in distribution, which brings economical and social inconveniences [1]. The economical inconveniences arise because users and companies have to assume the cost of losses (by means of an increase of the value that users must pay), which implies that the company's utilities decrease. On the other hand, sustainability and ambiental protection of the resource generate a public concern. For instance, in [1] the authors say that *"the resource is often oversupplied relative to demand, generally underpriced relative to its intrinsic and economic values, and governed by institutions geared to augment supply rather than to manage demand."* This is clearly a social inconvenience.

In order to minimize leakages we need to study the transients that often appear in systems where valves are utilized. Manual or servo actuated valves, commonly used on WDS, have effects on transients due to the opening and closing times and the changes in the consumption patterns [2]. There is enough evidence that pressure control must be sampled in short intervals to study transients in each pipe, which imposes restrictions on the sampling, processing time, and algorithm complexity [3].

The influence of pressure reducing valves (PRV) in a WDS has been investigated in several publications. In [4], the authors conclude that the use of PRVs can reduce the losses by leakage to 50% of its original value. In [5], the authors formulate the PRV model using several approximations. These models contributed to develop a proportional controller in

This work has been supported by B&C Biosciences Ltda, Bogotá, Colombia. This work has also been supported by Plan de Apoyo a Profesores Asistentes, Universidad de los Andes. The authors are with Departamento de Ingeniería Eléctrica y Electrónica, Universidad de los Andes, Bogotá, Colombia (e-mail: ej.ramirez34; nquijano@uniandes.edu.co). The authors would like to thank S. Leirens for his inputs.

[3], which has problems with oscillations and settling time. An improvement in the control scheme in this particular application is introduced in [6], which uses a PID controller. A faster tracking response is obtained without oscillatory issues. The present document extends these works based on foraging theory for the design of an optimal controller that increases the system robustness.

In order to increase the robustness, some bioinspired approaches have been proposed to solve different problems. For instance, in [7], the authors reduce the E. Coli complexity to perform the optimization in dynamic environments. On the other hand, in [8] the authors combine genetic algorithms with foraging theory to improve the convergence in a non-linear system. In each case, the foraging algorithm proposed has shown advantages in implementation. In this paper we study the effects of using this approach on another problem. This work uses two models to validate the algorithm. The first one is based on an indirect adaptive controller, while the second one uses Non-linear Autoregressive with Exogenous Inputs (NARX) [9]. We design and implement the controller using adaptive control together with foraging theory as an optimization method. The foraging algorithm is based on the behavior of the E. Coli bacteria, which has 3 fundamental steps in the form of surviving, searching for, and obtaining nutrients (chemotaxis, reproduction, and elimination and dispersion) [10]. This implies that each bacteria is an agent searching the optimum place where the food is found, and noxious substances are avoided using some type of swarming among the whole group.

The document is divided as follows. First, we present different WDS models in order to get one for the PRV. In Section III we explain the E. Coli algorithm and its variations using direct and indirect approaches. In Section IV, we give some insights about implementing the E. Coli algorithm in a PRV application, and we discuss the advantages for using the proposed algorithm. Finally, some conclusions are drawn in Section V.

II. WATER DISTRIBUTION SYSTEM MODELS

There is a tendency in the water distribution companies to separate the distribution areas in closed small sectors called District Metered Areas (DMAs). Each DMA brings facilities for supervising and controlling the two main variables in the distribution, i.e., flow and pressure. Based on this, we explain the necessary models that describe the system behavior in each DMA. The pipes and their transients are described by one dimensional model using the so called *water hammer* equations. The demand profile (i.e., flow consumption) is

modeled in order to simulate the users dynamics. Finally, the PRV system (i.e, the actuator) is based on the behavioral model [5].

A. Demand Model

The demand in an WDS can be modeled by probabilistic events, which depend on the type of user (e.g., industrial or residential), pressure measured in each node, and sometimes on the weather. This work uses user depended models, considering the fact that the pressure in each node is adequate, and ignoring weather effects on changes in the consumption patterns [11]. The two types of demand models described in the literature are the residential and the industrial models. The first one has a great quantity of users with a low average of consumption. The second one is determined by fewer users with a high consumption, where the sum of the individual demands constitutes the total consumption of the system [6]. Independently of the type of model, what is used to define the consumption over one period of time, q_m , is the following: i) we define the average consumption, f_{avg} , based on empirical data; ii) next, over a sequence of rectangular pulses C_j , we define the duration and the intensity, i.e., T_i (seconds), and I_i (liters per seconds, l/s), respectively. The duration T_i is represented by an exponential distribution that is function of the arrival times τ_i , and I_i is represented by a Weibull distribution. Therefore, the consumption can be written as

$$q_m = \sum_{i=1}^{C_j} T(\tau_i)I_i(\tau_i) + f_{avg}(\tau_i) \quad (1)$$

The stochastic model is necessary for the correct simulation of the demand throughout the system and it can be seen as a system's perturbation (e.g., Section IV). This model suggests that f_{avg} will not change suddenly over time, but T_i and I_i change depending on the type of user. In other words, the industrial user will increase its consumption over a large time, and the residential user will consume less and increase the frequency. Both behaviors can be modeled by Equation (1), modifying their intensity and duration in each case.

B. Transients Model

The WDS is composed by several DMAs, where each one is assumed to be small, close, and composed by several nodes or service points joined by pipes among them. Sometimes, control valves are introduced to achieve some goals on the system. These valves can generate transients depending on the state [2]. To describe this behavior it is necessary to introduce the *water hammer* equations, which represent the dynamics of the transients of the system. This model is formulated in [12], and it is summarized in the numerical solution of the equations

$$g \int_{h_A}^{h_P} dh \pm \frac{a}{A} \int_{q_A}^{q_P} dq + \frac{f}{2DA^2} \int_{x_A}^{x_P} q|q|dx = 0 \quad (2)$$

where the second term can be positive or negative representing two equations that we call C^+ and C^- . In (2), h is the piezometric head, q is the flow over the pipe, x is the length of the pipe, D is the pipe diameter, A is the pipe area, f

is the Darcy-Weisbach friction factor, a is the wave velocity (supposedly to be constant), and g is the gravity. In order to find a numerical solution, the pipeline must be divided in N sections, each one with a length Δx . To find C^+ a time step $\Delta t = \Delta x/a$ is computed along the points AP , as seen in Figure 1. C^- is computed along the points BP . Note that the points A, B, P are used as subindices to represent the pressure (h_A , h_B or h_P), or the flow (q_A , q_B or q_P), or the position (x_A , x_B or x_P) in each pipe. Note that we do not show the numerical results for Equation (2), but we use a numerical approximator to compute them (a second order trapezoid rule is used in all simulations).

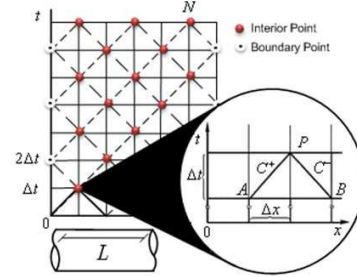


Fig. 1. Characteristic method for the water hammer equations, adapted from [12], [13]. L corresponds to the length of the pipe, and the points A, B, and P are used as subindices to represent the pressure (h_A , h_B or h_P), or the flow (q_A , q_B or q_P), or the position (x_A , x_B or x_P) in each pipe.

C. Pressure Reducing Valve Model

In an effort to reduce the pressure in each DMA it is necessary to use a control element placed between the water supply and the DMA. In this work we use a PRV since the benefits have been demonstrated [4]-[6]. According to [5] and [6], the best way to represent a PRV (shown in Fig. 2) is using the behavioral model described by Equation (3). This set of equations offers low complexity and high approach to the real operation of this type of valves [5]. The operation of the valve consists on strangling the flow q_m with a pressure h_{in} , to get a desire pressure h_{out} by the opening or closing of the diaphragm x_m through the input/output of liquid q_c on the cavity. The flow q_c can enter into the cavity because h_{in} will always be greater than the pressure on the cavity. The same phenomenon occurs when the flow q_c gets out from the cavity. The pressure h_{out} will be less than the one in the cavity. Analytically that is,

$$\begin{aligned} \dot{x}_m &= \frac{q_c}{A_{cs}(x_m)} \\ q_c &= \begin{cases} \alpha_{open}(h_{set} - h_{out}), & \dot{x}_m \geq 0 \\ \alpha_{close}(h_{set} - h_{out}), & \dot{x}_m < 0 \end{cases} \\ q_m &= C_v(x_m) \sqrt{h_{in} - h_{out}} \end{aligned} \quad (3)$$

In (3), C_v is the valve capacity which is a function of x_m , A_{cs} is the transversal area of the control space given as a function of the opening, α_{open} and α_{close} are two constants that are related to the servo-valve speed, and h_{set} is the reference pressure (i.e., the set point). The variables related to the static characteristics (i.e., $A_{cs}(\cdot)$, $C_v(\cdot)$, α_{open} , and α_{close}) can be determined by a set of experiments [5]. The flow q_m is the same one shown in Equation (1), i.e., the demand model.

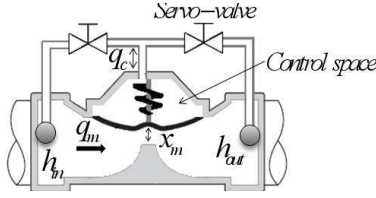


Fig. 2. Pressure reducing valve, adapted from [6]. h_{in} corresponds to the input pressure, q_m is the flow over the valve, x_m is the diaphragm space, q_c is the flow through the cavity, and h_{out} is the output pressure.

III. E. COLI ADAPTIVE CONTROLLER

The controller that we use is based on adaptive control and foraging theory as an optimization method. The foraging algorithm is based on the behavior of the E. Coli bacteria, which has three fundamental steps: chemotaxis, reproduction, and elimination and dispersion [10]. Chemotaxis is used in order to move and search for food; reproduction is used to make an identical copy; and elimination and dispersion is used to avoid noxious elements that can kill or spread a group of bacteria in the search space. The complete algorithm includes these three steps representing an important amount of operations that cannot allow real time implementation. Instead, we propose a simplification using just the chemotaxis step (illustrated in Algorithm 1). In this case, each bacteria i , $i = 1, \dots, S$, moves on the space $\theta^i(j) \in \mathbb{R}^p$, where S is the total amount of bacterias, j represents each iteration of the algorithm until a maximum time T is reached, and p is the dimension of the space. The bacteria is always seeking to increase favorably its environment. Thus, it can sense the changes in the nutrient's concentration. If the change increases, it continues moving N_s steps with size $C(i)$ in the $\Delta(i)$ direction. If not, it generates a random direction to search more concentration of nutrients. Then, we use a type of memory $J(i, j)$ to evaluate the performance of each bacteria in the last N steps. $J(i, j)$ depends on the error e , which represents the difference between the estimated pressure $\hat{h}(\theta^i(j))$ and the measured pressure $h_{out}(j)$ at time j . Two control strategies based on conventional adaptive control are used. These are the indirect and the direct adaptive controllers adapted from [10]. The first one is based on online identification method to estimate the plant dynamics. The second one observes the control signal and adapts the parameters to maintain the performance. In the indirect case, the E. Coli algorithm estimates the best parameters θ^* of a given model $\hat{\alpha}$ and $\hat{\beta}$. In the direct case, the algorithm estimates the control signal u^* using the plant structure ϕ_u . In both control schemes, each bacteria moves in order to change the parameters of plant structure. The bacteria who has the best behavior in a part of his life (i.e., minimize the cost function J based on N last steps) is chosen by the proposed algorithm to be applied in the control signal u^* .

A. Indirect Adaptive Controller

The type of plant considered here is $y(k+d) = f(x(k), u(k))$, where $u(k)$ is the control signal, $y(k)$ is a smooth function representing the output of the plant,

Algorithm 1 Chemotaxis step

```

Random initialization of  $\theta^i \in \mathbb{R}^p$  over search space
for  $j = 1$  to  $T$  do
  Chemotactic step
  for  $i = 1$  to  $S$  do
     $e(\theta^i(j)) = \hat{h}(\theta^i(j)) - h_{out}(j)$ 
     $J_{cc}(\theta^i(j)) = (e(\theta^i(j)))^2$ 
     $J(i, j) = J(i, j) + J_{cc}(\theta^i(j))$ 
     $J_{last} = J(i, j)$ 
    Tumble: random vector  $\Delta(i) \in \mathbb{R}^p$ 
     $\Delta_m(i), m = 1, 2, \dots, p$ 
     $\theta^i(j+1) = \theta^i(j) + C(i) \frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \{\text{run}\}$ 
     $m = 0$  {swim}
    while  $m < N_s$  do
       $m = m + 1$ 
      if  $J(i, j+1) = J_{last}$  then
         $J_{last} = J(i, j+1)$ 
         $\theta^i(j+1) = \theta^i(j+1) + C(i) \frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}$ 
      else
         $m = N_s$ 
      end if
    end while
    next bacteria  $i + 1$ 
  end for
   $J$  minimized according to the last  $N$  steps
  if  $j > N$  then
     $J(i, j) = \sum_{k=j-N}^j (e(\theta^i(k)))^2, \forall i = 1, \dots, S$ 
     $\theta^*(j) = \arg \min \{J(i, j) : i = 1, \dots, S\}$ 
  end if
end for

```

$d \geq 1$ represents the delay among input and output. The state vector $x(k)$ is represented by $x(k) = [y(k), \dots, y(k-n), u(k-1), \dots, u(k-m)]$, $n, m \geq 0$. Clearly $x(k)$ depends on the amount of memory of the system, where n and m represent the number of past outputs and inputs, respectively. In this case, we use the type of plant represented by $y(k+d) = \alpha(x(k)) + \beta(x(k))u(k)$, where $\alpha(x(k))$ and $\beta(x(k))$ are unknown smooth functions. $\beta(x(k))$ must be bounded away from zero (i.e., non minimum phase plants are treated here). To linearize the dynamics, we use the estimated output $\hat{y}(k+d)$, i.e., $\hat{y}(k+d) = \hat{\alpha}(x(k)) + \hat{\beta}(x(k))\hat{u}(k) = \theta_{\alpha, \beta}^T \phi_{\alpha, \beta}(x)$, where $\theta_{\alpha, \beta}$ are the parameters of the fixed structure $\phi_{\alpha, \beta}$. In order to approximate \hat{u} to the ideal controller $u^*(k)$, we use the approximator composed by the estimated values for $\alpha(x(k))$ and $\beta(x(k))$, i.e., $\hat{u}(x(k), r(k+d)) = \frac{-\hat{\alpha}(x(k)) + r(k+d)}{\hat{\beta}(x(k))}$, where r is the reference to be tracked.

In order to achieve the appropriate \hat{u} , we have to minimize the error between the measured output y and its approximator \hat{y} , which implies tuning the parameters, i.e.,

$$\theta_{\alpha, \beta}^* = \arg \min_{\theta_{\alpha, \beta} \in \Omega_{\alpha, \beta}} \left(\sup_{x \in S_x} \left| \theta_{\alpha, \beta}^T \phi_{\alpha, \beta}(x) - y \right| \right), \text{ where } \theta_{\alpha, \beta}$$

is defined in the compact parameter set $\Omega_{\alpha, \beta}$, and S_x is the compact set where the state trajectory of $x(k)$ may travel under closed-loop control. Note that the parameter $\theta_{\alpha, \beta}^i$ means that each bacteria $i = 1, \dots, S$, can adjust the approximator $\hat{y}(k+d)$. Thus, the bacterial population tries to move close to the plant behavior, which in the PRV case is h_{out} .

B. Direct Adaptive Controller

In the direct case, in order to obtain the optimal control signal $u(k)$ at each discrete time k , it is necessary to approximate the measured output of the system y using a plant prediction, i.e., $\hat{y}_j^i(k+1)$, $j = 1, \dots, n_j$, where n_j is the number of outputs of the plant, and $i = 1, \dots, S$, is the total number of bacteria. We consider the non-linear system as a

black box, where the output $y_j(\cdot)$ is obtained by introducing to it an input vector $x(k)$ with size $1 \times n_x$. Input and output are introduced into a multilayer perceptron (MLP), i.e., $\hat{y}_j(k+1) = g_0 \left(\sum_{i=1}^{n_h} w_{ij} x_i^{(1)} + a_j \right)$, and $x_i^{(1)} = g_1 \left(\sum_{j=1}^{n_x} w_{ij} x_j + a_i^{(1)} \right)$, where g_0 is the activation function of the output neuron, g_1 is the activation function of the hidden layer, $a^{(1)}$ is a vector representing the independent term (bias) in the hidden layer, a represents the bias in the output layer, n_h is the number of neurons in the hidden layer, $w^{(1)}$ is a matrix that contains the weights between the input and the hidden layer, w is a matrix that contains the weights between the hidden and the output layer, and n_x is the number of inputs [9]. Then, the MLP is trained with backpropagation to tune the model parameters. We specifically use Levenberg Marquardt training. This type of identification is called NARX [9] that is represented by $\hat{y}_j^i(k+1) = f_m(y_j^i(k), \dots, y_j^i(k-n_y), x^i(k-1), \dots, x^i(k-n_u))$, where n_u and n_y represent the past inputs and outputs, respectively.

In order to develop a direct controller, we update $u^i(k)$ via the E. Coli algorithm proposed using $u^i(k) = F_u(x(k), r(k+1), \theta_u^i(k)) = \theta_u^{iT} \phi_u(x(k), r(k+d))$, where $\theta_u^i(\cdot)$ represents the parameters to be tuned, $\phi_u(\cdot)$ is the approximator structure that is fixed. Then, we find the point θ_u^* in the convex compact set of search space $\Omega_u \subset \mathbb{R}^p$ that minimizes the error $e^i(k) = r(k+d) - \hat{y}^i$ by the i -th bacteria, i.e., $\theta_u^* = \arg \min_{\theta_u^i \in \Omega_u} \left(\sup_{x \in S_x, r \in S_r} \left| \theta_u^{iT} \phi_u(x, r) - u(x, r) \right| \right)$ where S_x is the compact set where the state trajectory $x(k)$ may travel under closed-loop control, and S_r is the reference input space that is bounded. In order to evaluate each bacterial behavior, it is necessary to introduce a cost function $J(\theta_u^i(k), N, k) = w_1 \sum_{j=k}^{k-N} (r(k+j) - \hat{y}^i(k, j))^2 + w_2 \sum_{j=k-1}^{k-N-1} (u^i(k, j))^2$. The two scaling factors w_1 and w_2 are used to weight the importance of achieving the tracking error near to zero, and minimizing the control energy, respectively.

IV. SIMULATION RESULTS AND DISCUSSION

A. Indirect Case

In order to compare our results, we use the same PID controller designed and tuned to control the PRV introduced in [6]. Our indirect controller has also been tuned using the simulation parameters as in [6]¹. To check the robustness of both controllers, we test the PID and our controller approach with a lower demand profile (i.e., a small value of q_m), and with an increase in the actuators velocities. The lower demand profile is used to test the behavior of the controller at night, where the consumption is reduced in several residential areas. The speed increment of the actuators represents the change of a servo-valve manufacturer. Every other parameter

¹The simulations are performed using the following parameters: $a = 1200m/s$, $f = 0.5mm$, a sampling time of $100ms$, $D = 150mm$, $g = 9.8m/s^2$, the set point $r = 30$ mH2O, $S = 10$, $N_s = 4$, $C = 0.01$, $\alpha_{open} = \alpha_{close} = 1 \times 10^{-6}$, a demand profile between 10 l/s and 15 l/s. The traversal area (A_c) and the valve capacity (C_v) are functions of x_m as it is shown in [6]. The PID values are: $K_p = 7.9 \times 10^{-5}$, $K_d = 2.4 \times 10^{-7}$ and $K_i = 6 \times 10^{-3}$.

remains equal. The PID response to a lower demand profile is shown in Fig. 3, and the E. Coli response is shown in Fig. 4. The response on lower demand is similar in both controllers, but the E. Coli algorithm has an increase in the response time. Hence, the only drawback for the lower demand case is in the settling time, which increases.

Figures 5 and 6 show the servo-actuators change (i.e.,

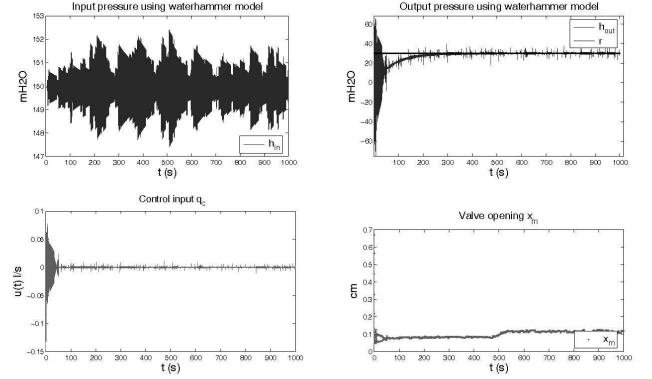


Fig. 3. PID response on a PRV using a demand profile between 2.6 and 3 l/s (lower profile demand).

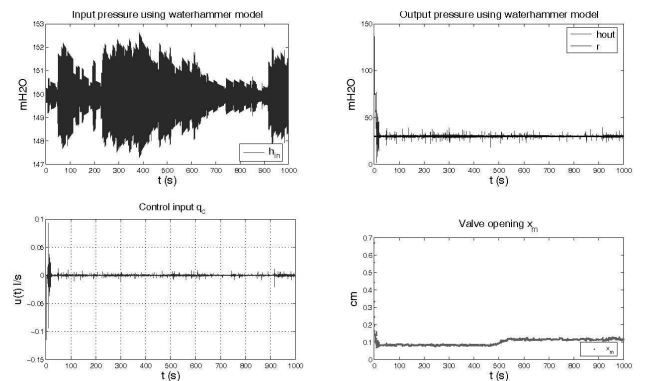


Fig. 4. E. Coli indirect controller response on a PRV using a demand profile between 2.6 and 3 l/s (lower profile demand).

$\alpha_{open} = \alpha_{close} = 3 \times 10^{-6}$). Note that we also introduce a change in the consumption from 10 to 15 l/s at $t = 500s$, which represents an industrial demand. With the change to non nominal conditions in the servo-actuators, the PID must be tuned again because it does not settle down. However, our proposed E. Coli adaptive controller performs much better to these changes, which represents the robustness advantage on using our controller. Table I summarizes the statistical results when we compare the behavior in the output pressure h_{out} of these two controllers. Nevertheless, the behavioral model used to simulate the PRVs is not useful in implementation because the control signal u is given as flow in liters per second (l/s). Normally we use a pair of servo-actuators like solenoid valves, which implies that the control signal u must be the time of opening. Therefore, we have to find a relationship between the flow q_c and the time of opening. Since this clearly represents a problem, we introduce next

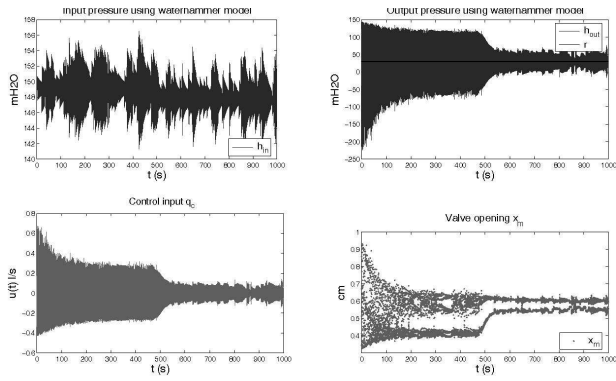


Fig. 5. PID response on a PRV using a demand profile between 10 and 15 l/s with speed increment $\alpha_{open} = \alpha_{close} = 3 \times 10^{-6}$.

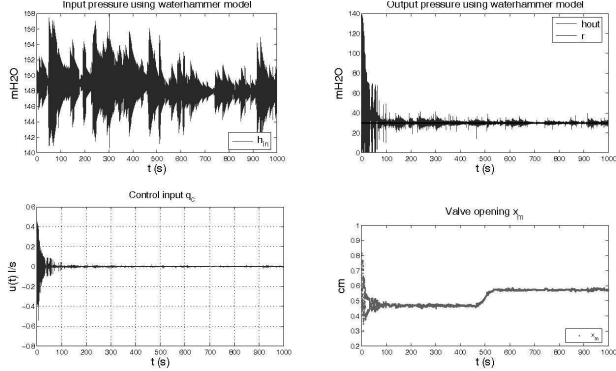


Fig. 6. E. Coli indirect controller response on a PRV using a demand profile between 10 and 15 l/s with speed increment $\alpha_{open} = \alpha_{close} = 3 \times 10^{-6}$.

another approach that uses an online identification of the plant.

TABLE I
STATISTICAL DESCRIPTION OF PID AND E. COLI SIMULATIONS.

Simulation	Mean (mH2O)	Standard Deviation(mH2O)	Settling Time(s)
PID (Fig. 3)	29.93	0.75	250
E. Coli (Fig. 4)	29.99	0.67	40
E. Coli (Fig. 6)	30	1.05	150

B. Direct Case

The simulations shown in this section are based on an identification model for a PRV. In order to get an approximation of the system's behavior we use an experiment, where the PRV system is excited by the control signal $u(t)$ (opening times for servo-valves) to find the model \hat{y} that captures the dynamics of the output pressure h_{out} . Like it was shown in the last section, we use NARX method to choose the adequate structure of \hat{y} and its parameters. In order to evaluate how good is our model, we perform three tests after the training. The first one (Fig. 7-a) shows the response of h_{out} and \hat{y} to the input signal $u(t)$ using the same data of the backpropagation training. The second one (Fig. 7-b) shows the response to new data (i.e., data that

are not used in the training test). The third one (Fig. 7-c) is similar to the last test, but we do not use h_{out} as an input for our model. The first test does not have good information about model generalization. For this reason, we introduce the second test, where we evaluate the behavior of the model to newer data. This last simulation suggests that the model has good generalization property because we obtain an error less than 1% between \hat{y} and h_{out} . The third test is used to see how the model can predict the behavior of the plant over time without full information (i.e., we use only the initial condition and $u(t)$ from the experiment). This test shows that \hat{y} and h_{out} has a maximum error of 20%, but the changes in both outputs have the same proportion. The E. Coli direct controller has to find the best bacteria that minimizes the cost function $J(\theta_u^i(k), N, k)$, i.e., to find the best control signal \hat{u} . The NARX model works as one step predictor for the pressure output $\hat{y}(k+1)$. Figure 8-a shows the response to several operation points. Figure 8-b shows the control signal u . We can see some oscillations in the output due to the error introduced in the reference model \hat{y} . The real and the reference model differ in 10% in some parameters, which emulates non exactly knowledge of the plant. In contrast to the indirect case, here the control signal u is shown in seconds, which represents the time of aperture of each servo-actuator. The NARX model includes in its dynamics the influence of q_c in the control space, which is helpful in the implementation of the controller because the control signal u is referred to time and not to flow.

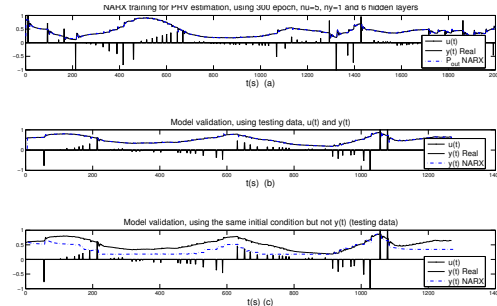


Fig. 7. Plant identification: NARX technique with $n_u = 5$, $n_y = 1$, six hidden layers, and 300 epochs. a) Training, b) Testing, c) Generalization.

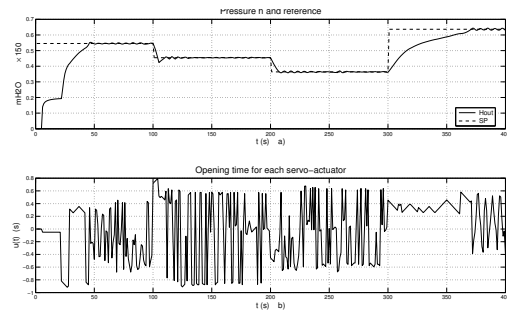


Fig. 8. E. Coli simulation using multiple operation points with $S = 10$, $N = 20$, $N_c = 4$, and $C = 0.02$. a) PRV response, b) Control signal $u(t)$.

C. Discussion

The WDS simulation in this document is based on some assumptions that simplifies the model complexity. First, we assume that the flow is independent of the pressure based in [11], although this assumption is not always valid. Next, we do not describe when cavitation can appear, which may be present when x_m is low, and the water velocity is increased. This phenomenon generates bubbles in the water and it changes the PRV dynamics. Therefore, we try to present a general model that includes the behavior of the pipes, valves, and user dynamics in the system. These models have been accepted in the literature [6], [12]. We use these models in one part of this article to show that our control scheme works fine in a theoretical scenario. However, we introduce a system identification experiment in a PRV system in order to implement our control algorithm.

Tuning a controller using the theoretical PRV model implies almost three experiments to obtain the model parameters like A_{cs} , C_v , α_{open} , and α_{close} [5]. It is necessary to use almost four sensors to measure the flow (i.e., q_m) and the pressure (i.e., h_{in} , h_{out} , and the pressure on the control space). These issues make this alternative allowable only for academics goals. Another alternative based in experimentation is presented in [6], where a PID is tuned using the Åström relay method [14]. They conclude that this technique is not perfect and they are obligated to use a fine tuning method to improve the controller performance, which implies more time in experimentation without any knowledge on the robustness. For these reasons, we are motivated to extract the dynamics of the system via system identification. The NARX experiment implemented uses one sensor to measure h_{out} , and in our case, two solenoids valves as control elements.

We use two types of schemes to test the robustness of the controller. The first one is based on the change of the demand profile. Another one is based on the servo-valve parameters. The first test suggests that the PID has settling time problems in lower consumption. The second one shows that the PID has to be tuned again every time when the plant parameters change. Instead, our bioinspired approach does not have problems in lower consumption, and it does not have to be tuned again because it is adaptive. This implies that the commissioning time decreases, which leads to have a better service and savings for the company. Although the PID controller offers a quick settling time to nominal conditions, the Åström relay method does not guarantee the robustness of the system. In contrast, the adaptive controller biologically motivated by the E. Coli bacteria possesses the advantages of the good controllers, as well as a good response when the parameters change (i.e., robustness). Contrary to other controllers, this one offers the feasibility to be used with non-linear models in the approximators of the system, with the possibility of real time operation.

V. CONCLUSION

We have shown in this paper an improvement to the control scheme in a PRV. The demand variable and the transients in the system make this problem a challenging

one in the controller design for a PRV. We extend the results of previous works including two probable scenarios like low consumption and plant changes, in order to test the robustness of the controller. For that, we use a PRV model, and an identified one. The behavioral model is used to compare the performance of the proposed algorithm with previous results using a PID. Then, the experimental model, which is used to capture the system's dynamics via system identification (NARX), allows the controller implementation using a specific servo-actuator in the plant. A modified E. Coli algorithm is proposed to be applied using adaptive control schemes. The algorithm offers some advantages over the PID, like the incorporation of the non-linear model based on system identification (i.e., NARX). Complexity is not a barrier to implement it because the original algorithm is simplified, and it is shown that it converges in a PRV application. We obtain good responses to plant changes, which suggests that this controller can be used in commercial and industrial applications. In the future, we would like to implement our controller in Florencia (Caquetá, Colombia), where we want to improve the behavior of the PID controllers which are now used to control the PRVs.

REFERENCES

- [1] D. Brooks and G. Rouse. Another path not taken: A methodological exploration of water soft paths for Canada and elsewhere. Technical Report 1, Ottawa: Friends of the Earth Canada, 2007.
- [2] B. Brunone and L. Morelli. Automatic control valve-induced transients in operative pipe system. *Journal Of Hydraulic Engineering*, 125(5):534–542, May 1999.
- [3] S. Prescott and B. Ulanicki. Investigating interaction between pressure reducing valves and transients in water networks. *Proceedings 49th Int. Scientific Colloquium*, 1:49–54, 2004.
- [4] B. Ulanicki, J. Rance, and P. Bounds. Control of pressure management areas supplied by multiple prvs. Research Report Water Software Systems LE19BH, De Montfort University, UK, 1999.
- [5] S. Prescott and B. Ulanicki. Dynamic modeling of pressure reducing valves. *Journal of Hydraulic Engineering*, 129(10):804–812, 2003.
- [6] S. Prescott and B. Ulanicki. Improved control of pressure reducing valves in water distribution networks. *Journal of Hydraulic Engineering*, 134(1):56–65, 2008.
- [7] W. Tang, Q. Wu, and J. Saunders. Bacterial foraging algorithm for dynamic environments. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pages 1324–1330, July 16–2 2006.
- [8] T. Chen, P. Tsai, S. Chu, and J. Pan. A novel optimization approach: Bacterial-ga foraging. *International Conference on Innovative Computing, Information and Control*, pages 391–391, September 2007.
- [9] I.J. Leontaritis and S.A. Billings. Input-output parametric models for non-linear systems. part ii: stochastic non-linear systems. *International Journal of Control*, 41:329–344, May 1985.
- [10] Kevin M. Passino. *Biomimicry for Optimization, Control and Automation*. Springer-Verlag, London, UK, 2005.
- [11] Z. Yi Wu, R. He W., T. Walski, S. Yu Yang, D. Bowdler, and C. C. Baggett. Efficient pressure dependent demand model for large water distribution system analysis. In *8th Annual International Symposium on Water Distribution System Analysis*, pages 1–15, August 2006.
- [12] E. Benjamin Wylie and Victor L. Streeter. *Fluid transients in systems*. Prentice Hall, Englewood Cliffs, New Jersey, 1993.
- [13] Duncan McInnis and Bryan W. Karney. Transients in distribution networks: Field test and demand models. *Journal of Hydraulic Engineering*, 121(3):219–231, March 1995.
- [14] K. Åström and T. Hägglund. *PID controllers: Theory, design, and tuning*. Research Triangle Park, N.C. International Society for Measurement and Control, 2nd edition, 1994.