Mitigating the Effects of External Perturbations on a Gene Regulatory Network using Feedback Controllers

Abhinandan Khan*, Goutam Saha†, and Rajat Kumar Pal*
*Department of Computer Science and Engineering, University of Calcutta
Acharya Prafulla Chandra Roy Siksha Prangan, JD-2, Sector-III, Saltlake, Kolkata – 700106, India
Email: {khan.abhinandan, pal.rajatk}@gmail.com
†Department of Information Technology, North-Eastern Hill University, Shillong
Umshing Mawkynroh, Shillong, Meghalaya – 793022, India
Email: dr.goutamsaha@gmail.com

***Abstract*—Gene regulatory networks are generally robust in nature. However, unwanted perturbations arising out of extreme environmental conditions or external pathogen attacks may lead them to malfunction. Potentially, this can have an adverse effect on the biochemical functions of a living system. In this work, we have proposed a computational model based on negative feedback control to eliminate the effects of such unwanted perturbations. We have implemented the recurrent neural network formalism for modelling the underlying network dynamics from a given time-series gene expression dataset. The artificial bee colony optimisation technique has been employed for model parameter estimation. The controller used in this work is of the proportional-integral-derivative type. To the best of our knowledge, this is one of the first research works in this domain to consider a completely non-linear scenario. A 10-gene `DREAM4` benchmark network has been considered in this work. The results obtained herein show that the proposed formalism can mitigate the unwanted effects of external disturbances effectively.**

***Index Terms*—artificial bee colony, `DREAM4`, feedback control, gene regulatory networks, recurrent neural network.**

I. Introduction

Genes are the fundamental units of all living systems, from tiny single-cell entities to complicated multi-cellular organisms. The critical biochemical activities, viz. organ development, resistance to diseases, etc., take place inside living cells as a result of the intricate regulatory relationships amongst genes. The end products of gene expression are proteins, and they, in turn, act as the medium of communication amongst the genes. A *gene regulatory network* or a GRN is usually used to denote the regulatory inter-relationships between genes graphically. The nodes in a GRN represent the genes, while the edges signify the regulations [1]. Genetic regulations can be of two types, viz. *activation* and *repression*. If the expression of a gene is initiated or the rate of expression increases, under the influence of a gene/s, the type of regulation is known as *activation*. On the other hand, when the expression of a gene stops or the rate of expression reduces, the type of control is known as *inhibition* or *repression*.

Although GRNs are robust in nature, they can be susceptible to extreme environmental conditions like pollution, or pathogen attacks. Complex networks, other than GRNs, such as the world wide web, power grids used for electrical supply, social networking websites and applications, etc., are also vulnerable to external disruptions that can have an unfavourable affect on the entire network action. Some prominent examples of such unwanted effects are: (*i*) the circulation of malware through the internet with malicious intent, (*ii*) power outage in residential as well as industrial area due to the failure of the supply mechanism, (*iii*) the rapid propagation of baseless rumours and fabricated news via social networks, etc. in addition to the malfunction of critical biochemical functions in living things. Thus, researchers are becoming increasingly inspired to develop suitable strategies for eliminating the unwanted and harmful effects of external disturbances on the operation of complex networks.

Traditionally, researchers in the domain of network science [2], [3] have attempted to reduce the effects of external disturbances by concentrating on numerous network related notions, like hubs, components, degree of centralities, etc. Only recently, however, some researchers have turned their attention towards the concepts of feedback control [4], [5], [6], [7], and have proposed to eliminate the unwanted effects of perturbations in large-scale networks using the same. These research works consider external perturbations to be similar in nature to external disturbances in traditional control systems. Such systems need suitable control actions to guarantee that the output remains within the desired or feasible operating range. Hence, the problem of perturbation mitigation can be visualised as the problem of rejecting an external disturbance from the point of view of feedback control theory.

Thus, we have proposed a computational framework based on the concept of negative feedback control in this paper to eliminate the potentially harmful effects of external perturbations on a GRN. For this, we have considered a 10-gene *in silico* benchmark network taken from the **`DREAM4`** challenges [8], [9]. We have used the *recurrent neural network* (RNN) [10] formalism to model the underlying dynamics from the given temporal gene expression profiles. The *artificial bee*

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

colony (ABC) [11] optimisation technique has been implemented for the RNN model parameter training. We have used a *proportional-integral-derivative* or PID controller [12] to eliminate the potentially harmful effects of external perturbations on the considered network.

The rest of the paper has been organised as follows. Section II contains a brief overview of GRNs and their modelling, and the basics of RNN and ABC. We have presented the proposed methodology based on a PID controller in detail in Section III. The experimental results have been provided in Section IV along with relevant discussions. Section V concludes this work posing some future research opportunities.

II. Preliminaries

The basics of GRN analysis, the RNN formalism, the ABC algorithm, and the basic concepts of negative feedback control have been explained in Sections II-A through II-D

A. Modelling of Gene Regulatory Networks

The investigation of GRNs has developed into a critical topic of research in *Molecular Biology* because it can potentially reveal the reasons behind a disease and thus its lead to a prospective treatment. The reconstruction of GRNs from time-series gene expression datasets is a challenging task, nevertheless. With the rapid technological progress in the domain of genetic research, an extraordinary volume of gene expression related information has been produced. Notwithstanding some minor restrictions, the straightforward nature of present gene expression profiling strategies enables researchers to obtain large-scale concurrent measurements of gene expression values [13], [14], [15]. This has, in turn, made it feasible for scientists to investigate the dynamic behaviour [16] and communications amongst genes that are vital for the elucidation of key cellular activities, illustration of genetic activities, disease diagnosis, assessment of the effects of new drugs [17], [18], etc.

The extent and diversity of this available information offer the fundamental impetus to researchers worldwide to investigate and thereby build computational tools for the biologically credible analysis of the available data. The essential outcome of this investigation has led to the development of effective means for network identification, finding the gene(s) responsible for a specific ailment, etc. The well-known methodologies are Boolean Networks [19], Bayesian Networks [20], Information Theory based approaches [21], Regression based approaches [22], RNN [10], S-systems [23], etc. Chai *et al.* [24] and Kaini *et al.* [25] have provided a thorough review of the existing computational approaches in the contemporary literature for the reverse engineering of GRNs from temporal expression profiles.

The problem of reverse engineering GRNs from time-series expression data is an ill-posed problem, and thus is susceptible to *overfitting*. However, real GRNs are sparse [26]. Yet, the majority of the techniques available in this domain have been unable to predict network architectures with complete accuracy, even for small-scale GRNs. Few techniques have been able to identify all the correct relationships till now, but even they also infer a significant amount of false regulations. Augmenting the proposed models with supplementary biological information seems beneficial in enhancing the accuracy of the identified network architectures [27], but the accurate identification of large-scale GRNs are yet to be achieved. This is the primary reason that we have considered a known benchmark network in this work, such that we can implement the synthetic controller on an accurate network without false predictions. The proposed methodology assumes (at least for this work) that the genetic regulations are known to us. Only the nature of the relationships have been obtained by training the RNN model parameters from the temporal expression datasets.

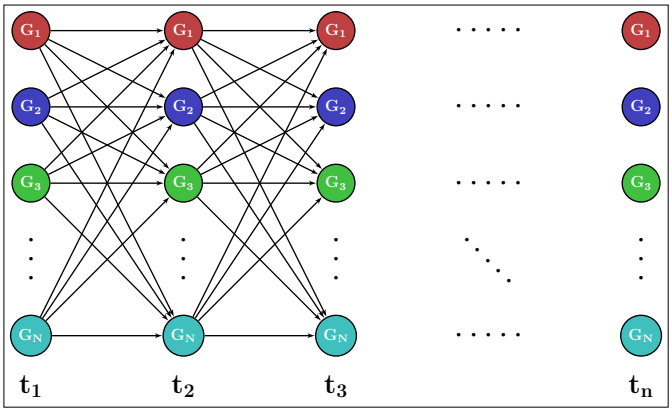


Fig. 1: The representation of a GRN using an RNN model. The network shown is unfolded from $t = t_1$ to $t = t_3$. All possible connections amongst the genes have been shown, whereas, real-world networks are sparse.

B. Recurrent Neural Networks

The regulation of the expression of a specific gene by some other gene/s can be represented using the RNN [10] framework shown in Fig. 1. The nodes denote genes while the edges signify the regulations among them. Each layer of the neural network (Fig. 1) represents the expression level of the genes at a specified time t_i. The amount/level of expression of any gene i, at any point of time, $t_{i+1} = t_i + dt$, depends on the expression levels of all the genes, x_j at the preceding timepoint t_i and the corresponding edge weights, $w_{i,j}$. Thus, the overall regulation on gene i can be encapsulated by the following:

$$g_i = \sum_{j=1}^{N} \left[w_{i,j} x_j + \beta_i \right], \tag{1}$$

where β_i denotes an external input that can be thought of as a reaction delay parameter. This relation can be transformed to the interval $[0, 1]$ with the help of a sigmoid function as shown in [10]. A larger value of β_i is symbolic of a reduced influence of $w_{i,j}$ on g_i. The rate of gene expression can then be modulated by a multiplicative constant, χ_1, which denotes the peak expression of a gene. The rate of expression of gene

i can be defined as the difference between the sum total effect of the regulators, δ_i and its self-degradation, γ_i, as follows:

$$\frac{dy_i}{dt} = \delta_i - \gamma_i, \tag{2}$$

where the degradation factor, γ_i can be defined on the basis of the kinetic structure of a first-order biochemical equation as:

$$\gamma_i = \chi_{2i} \cdot y_i \tag{3}$$

On the other hand, δ_i can be defined as:

$$\delta_i = \chi_{1i} \cdot f(g_i) \tag{4}$$

The constant χ_2 is the rate constant for the self-degradation part of the gene i. Thus, using (2), (3), and (4), we arrive to the following:

$$\frac{dy_i}{dt} = \chi_{1i} \cdot f\left(\sum_{j=1}^{N} w_{i,j} x_j + \beta_i\right) - \chi_{2i} \cdot \gamma_i, \tag{5}$$

where $f(\cdot)$ is a sigmoid transfer function, defined as:

$$f(a) = \frac{1}{1 + e^{-a}}$$

and x_j is the concentration of any element, j, in a given system (for $j = i$, $x_j = y_i$). Equation (5) describes the dynamics of gene expression and represents a node function [10], where each such can be connected to all other nodes to form a neural network as shown in Fig. 1.

The weight matrix, $W = [w_{i,j}]$ defines all the regulations or connections between the nodes of the network. A non-zero value of $w_{i,j}$ signifies the existence of regulation of gene i by gene j. The magnitude of the weight $w_{i,j}$ signifies the intensity of regulation. The neural network can be fully defined with the help of the differential equations corresponding to a specific gene (node), and the number of equations is defined by the number of nodes present, i.e. the number of genes in the network (N). The level of expression of gene i at time-point t can easily be computed from the set of differential equations. Equation (5) is a special instance of a class of RNNs, which are generally described as follows:

$$\tau_i \cdot \frac{dx_i}{dt} = f\left(\sum_{j=1}^{N} w_{i,j} x_j + \beta_i\right) - x_i \tag{6}$$

If $w_{i,j}$ is symmetric, the network represented by the model reaches stability in finite time [10]. Now, in real-world scenarios, data can only be obtained at discrete time-points only. As a result, assuming:

$$\frac{dx_i}{dt} \approx \frac{\Delta x_i}{\Delta t} = \frac{x_i(t + \Delta t) - x_i(t)}{\Delta t},$$

(6) can be rewritten in its discrete format as follows:

$$\tau_i \cdot \frac{x_i(t + \Delta t) - x_i(t)}{\Delta t} = f\left(\sum_{j=1}^{N} w_{i,j} x_j(t) + \beta_i\right) - x_i(t)$$

Simplifying the above, we get:

$$x_i(t + \Delta t) = \frac{\Delta t}{\tau_i} \cdot f\left(\sum_{j=1}^{N} w_{i,j} x_j(t) + \beta_i\right) + \left(1 - \frac{\Delta t}{\tau_i}\right) \cdot x_i(t) \tag{7}$$

C. Artificial Bee Colony Optimisation

The *artificial bee colony* (ABC) optimisation, proposed by Karaboga [11], mimics the cooperative foraging action of a group of bees. There are three different types of bees in an artificial bee colony, viz. *employed*, *onlooker*, and *scout* bees. At first, each *employed* bee chooses a food source randomly from the search space. The quality of each food source is estimated with the help of the given objective to assess their viability. The best solution, i.e. the food source with the maximum amount of nectar is stored in memory as well as shared within the swarm. The *employed* bees find a new food source (solution) in the neighbourhood of the chosen solution according to the following:

$$v^i = x^i + \phi^i \cdot (x^i - x^j), \tag{8}$$

where v^i is the newly found food source within the neighbourhood of the present one, x^i, being visited by the i-th virtual bee, such that $i \neq j$, and ϕ^i is a random number in the range $[-1, 1]$.

Next, the quality (fitness) of all the food sources (solutions) generated using (8) are calculated and subsequently compared with the earlier ones. If a virtual bee is able to find a better food source, it replaces the position of the old one with the current solution. The *onlooker* bees now enter the foraging process and calculate the probability of the food sources being selected by them. This information is shared with the other *onlooker* bees. We have calculated the probabilities in this work according to the approach proposed by Babayigit *et al*. [28]:

$$p^i = exp\left(\frac{-1}{\rho \cdot f^i}\right), \tag{9}$$

where p^i and f^i are the probability and the normalized fitness of the i-th food source, respectively; ρ is a control (input) parameter set to 1 here. In the subsequent step, the *onlooker* bees search for new food sources in the neighbourhood of the current best source according to the following:

$$v^i = x^{best} + p^i \cdot \left(x^{best} - x^j\right), \tag{10}$$

where the current global best solution is represented by x^{best}.

Here, j can be the same as $best$, unlike (8). Every time a new food source is found, the quality (fitness) of the food source (solution) is re-calculated, and the new source is selected if it is better than the old one. If any solution cannot be improved over a certain number of iterations, the food source is deemed to have been exhausted and the accompanying virtual bee becomes a *scout*, which then again generates a new food source randomly. The swarm size and the threshold number of iterations before a source can be assumed to be exhausted are two control parameters of ABC.

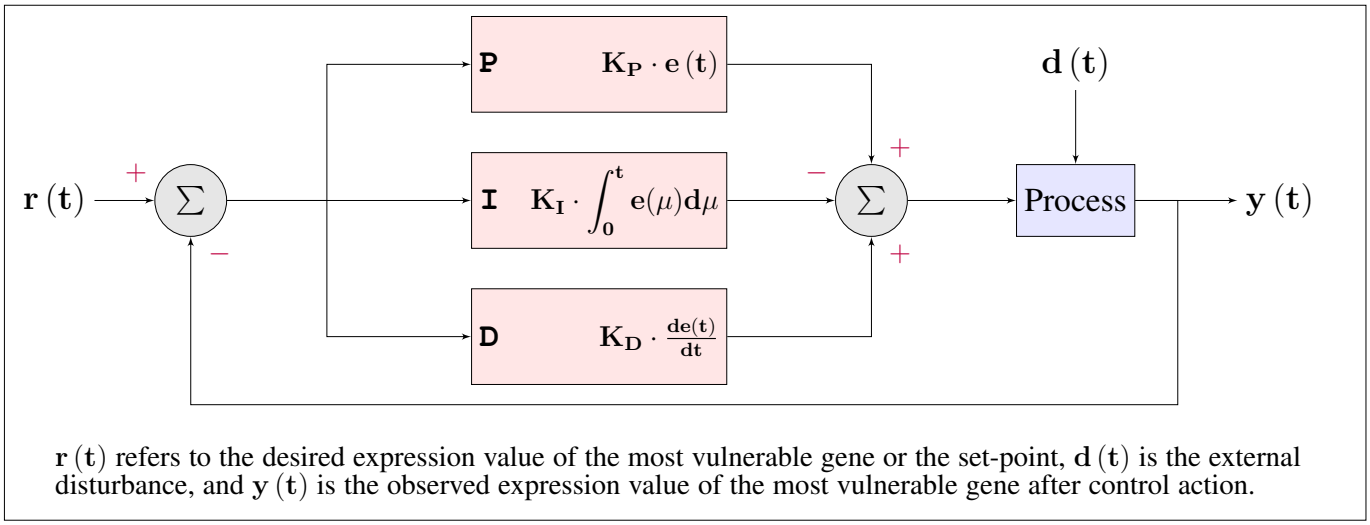


Fig. 2: A block diagram of the proposed PID controller.

D. Negative Feedback Control

Feedback is one of the most commonly used concepts in control theory, and is usually achieved with the help of a variety of approaches, viz. state space, full state feedback, etc. From the perspective of control theory, *feedback* is conventionally presumed to be *negative* [29]. The most frequently used all-purpose controller based on the concept of negative feedback loop is the *proportional-integral-derivative* or PID controller [12]. The terms of a PID controller can be explained based on time. The proportional part term is related to the current error. The integral part depends on the accumulation of past errors. Lastly, the derivative part provides an estimation of the expected error, based on the present error rate. A block diagram of the proposed PID controller has been shown in Fig. 2.

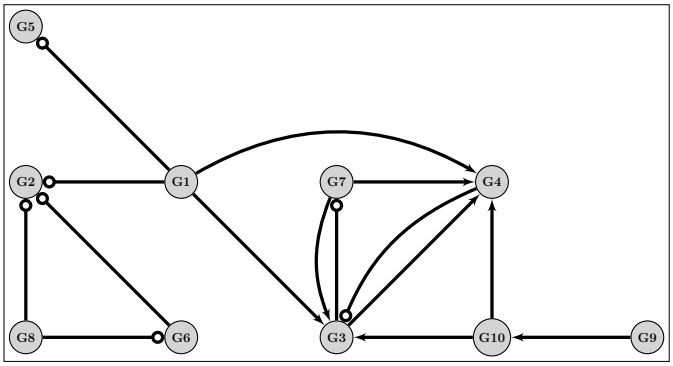


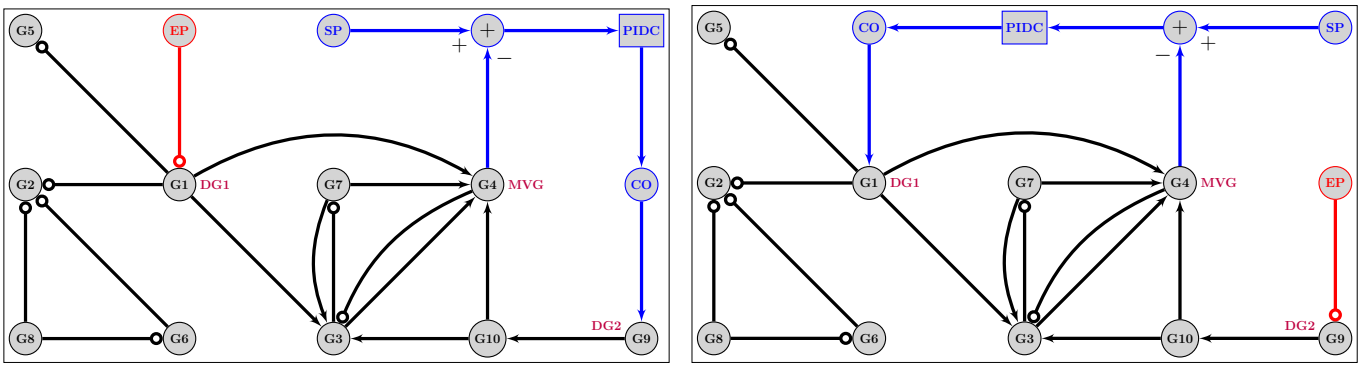
Fig. 3: A 10-gene **DREAM4** [8], [9] Challenge network extracted from **GNW** [30]. The arrowheads represent activation, and the open circles denote inhibition.

III. Proposed Methodology

Real-world GRNs existing in complex living systems are robust enough to reject small external disturbances or perturbations. However, extreme environmental conditions or pathogen infections can lead to malfunctioning of the genetic networks. This may lead to up-regulation or down-regulation of certain regulated gene/s in the network. This, in turn, may result in a respective increase or decrease in concentration of the end products of expression of those genes, i.e. the corresponding protein(s). This can hamper the functioning of the concerned proteins, which include life-critical activities like regulating the expression of other genes (that may belong to an altogether different network), forming protein complexes for acting as enzymes for metabolic reactions and other vital biochemical reactions, etc. This forms the primary motivation behind the present research endeavour.

Here, we have proposed a computational technique to mitigate the effects of external disturbances, based on the concept of feedback control theory. The present work can be divided into two main parts: (*i*) RNN model parameter training using the given time-series expression data to generate a suitable model, and (*ii*) designing a PID controller to eliminate the effects of an externally added disturbance to the trained model.

Let us consider the network given in Fig. 3. It is a **DREAM4** [8], [9] Challenge network comprising 10 genes and 15 regulations. In this network, two genes ($\mathbf{G1}$ and $\mathbf{G9}$) are not regulated by any other gene of the network, and act as regulators only. Hence, these genes have been termed as the *dominant genes* ($\mathbf{DG}$-s) of the network. External perturbations ($\mathbf{EP}$-s) can disrupt the normal functioning of the network, if only they can either increase or reduce the rate of expression of these $\mathbf{DG}$-s. On the other hand, gene $\mathbf{G4}$ is the most susceptible to the effects of any such perturbation, and hence such genes have been termed as the *most vulnerable gene* ($\mathbf{MVG}$). This is because there are several redundant pathways from the dominant genes to gene $\mathbf{G4}$ (e.g., $\mathbf{G1} \rightarrow \mathbf{G4}$, $\mathbf{G1} \rightarrow \mathbf{G3} \rightarrow \mathbf{G4}$, $\mathbf{G1} \rightarrow \mathbf{G3} \rightarrow \mathbf{G7} \rightarrow \mathbf{G4}$). As a result, the effect of any fluctuation in the level of expression of the dominant genes has a high probability of propagating to gene $\mathbf{G4}$ and



(a) The first perturbation mitigation approach.

(b) The second perturbation mitigation approach.

Fig. 4: (a) The first configuration of the proposed negative feedback controller used to eliminate the harmful effects of **EP** through **DG1** (i.e. gene **G1**). (b) The second configuration of the proposed negative feedback controller used to eliminate the harmful effects of **EP** through **DG2** (i.e. gene **G9**). **EP** stands for external perturbation; **DG** stands for dominant genes; **MVG** stands for the most vulnerable gene (i.e. gene **G4**), **PIDC** stands for the *PID controller*; **SP** stands for *set-point*; and **CO** stands for *control output*.

affecting its expression.

We have proposed to eliminate this problem by using a *negative feedback control* mechanism as shown in Fig. 4. It can be clearly seen from Fig. 4 that there are two possible configurations of the network. In the first configuration (shown in Fig. 4(a)), gene **G1** is the **DG** that is perturbed, while in the second configuration (shown in Fig. 4(b)), gene **G9** is the **DG** which is perturbed. Firstly, the edge weights have been estimated using RNN and ABC from the given temporal expression profiles. The RNN model used for training has been defined by (11)–(20):

$$x_1(t') = \lambda_1 \cdot f(\beta_1) + (1-\lambda_1) \cdot x_1(t) \tag{11}$$

$$x_2(t') = \lambda_2 \cdot f\left(\sum_{j=1,6,8} w_{2,j} x_j(t) + \beta_2\right) + (1-\lambda_2) \cdot x_2(t) \tag{12}$$

$$x_3(t') = \lambda_3 \cdot f\left(\sum_{j=1,4,7,10} w_{3,j} x_j(t) + \beta_3\right) + (1-\lambda_3) \cdot x_3(t) \tag{13}$$

$$x_4(t') = \lambda_4 \cdot f\left(\sum_{j=1,3,7,10} w_{4,j} x_j(t) + \beta_4\right) + (1-\lambda_4) \cdot x_4(t) \tag{14}$$

$$x_5(t') = \lambda_5 \cdot f(w_{5,1} x_1(t) + \beta_5) + (1-\lambda_5) \cdot x_5(t) \tag{15}$$

$$x_6(t') = \lambda_6 \cdot f(w_{6,8} x_8(t) + \beta_6) + (1-\lambda_6) \cdot x_6(t) \tag{16}$$

$$x_7(t') = \lambda_7 \cdot f(w_{7,3} x_3(t) + \beta_7) + (1-\lambda_7) \cdot x_7(t) \tag{17}$$

$$x_8(t') = \lambda_8 \cdot f(\beta_8) + (1-\lambda_8) \cdot x_8(t) \tag{18}$$

$$x_9(t') = \lambda_9 \cdot f(\beta_9) + (1-\lambda_9) \cdot x_9(t) \tag{19}$$

$$x_{10}(t') = \lambda_{10} \cdot f(w_{10,9} x_9(t) + \beta_{10}) + (1-\lambda_{10}) \cdot x_{10}(t) \tag{20}$$

Here,

$$t' = t + \Delta t \text{ and } \lambda_i = \frac{\Delta t}{\tau_i}$$

The set of all the parameters, i.e. all the τ-s, β-s, and the $w_{i,j}$-s has been estimated with the help of ABC. Here, the objective function that has been used for the ABC based parameter estimation has been defined as follows:

$$mse_i = \frac{1}{T} \cdot \sum_{t=1}^{T} \left[x_i(t) - \tilde{x}_i(t)\right]^2, \tag{21}$$

where T is the number of time-points available in the dataset, x_i is the original level of expression of gene i, and $\tilde{x}_i$ is the predicted level of expression of gene i. The process of ABC based model parameter estimation has been explained in detail using Algorithms 1 and 2.

The network model described by (11)–(20) constitutes the process of the proposed PID controller. The original level of expression of the most vulnerable gene has been assumed as the set-point for the controller. An external perturbation has been applied to one of the dominant genes and the change in the level of expression of the most vulnerable gene has been observed. This change is treated as the error for the controller to produce a control output. Subsequently, the generated control action has been used to alter the rate of expression of the other dominant gene. Again, the change in the level of expression of the most vulnerable gene has been observed and a new control output has been generated. This process has been repeated until the effect of the external disturbance is eliminated to the maximum possible extent. A schematic of the proposed controlling mechanism has been shown in Fig. 4.

IV. Experimental Results and Discussion

In this work, we have implemented our proposed technique on the 10-gene network shown in Fig. 3. The network is a

Algorithm 1 ABC Optimisation

Input: The number of genes in the GRN (N); the maximum number of iterations of ABC $(maxit)$; and the population of the hive in BAPSO (ph).
Output: The weight matrix representing the structure of the inferred network (W).

for gene $g = 1$ to N **do**
  Initialise position vector, $\mathcal{P}^g = [p_i^g]_{1\times ph}$ randomly.
  Each element, p_i^g is defined as: $p_i^g = [w_{i1}^g, w_{i2}^g, \ldots, w_{iN}^g, \beta_i^g, \tau_i^g]$, where N is the number of genes.
  Calculate the fitness, er_i^g, of each of the employed bees using Algorithm 2 and store them in the fitness vector, $\mathcal{E}^g \leftarrow [er_i^g]_{1\times ph}$.
  Store the minimum (best) fitness, $fit_{best} \leftarrow minimum\,(\mathcal{E}^g)$ and its index in min.
  Calculate the *global best solution*, $gb^g \leftarrow p_{min}^g$.
  for $iter = 2$ to $maxit$ **do**
    Calculate position of new food sources (v_i^g) for employed bees using 8.
    Calculate the fitness, ter_i^g, of the updated food sources using 2 and store them in $\mathcal{TE}^g \leftarrow [ter_i^g]_{1\times ph}$.
    For each bee, update $p_i^g \leftarrow v_i^g$ and $er_i^g \leftarrow ter_i^g$, if $ter_i^g < er_i^g$.
    Update fit_{best}, min, and gb^g.
    Calculate probability of selection using (9).
    Calculate position of new food sources using (10).
    Calculate the fitness, er_i^g, of the onlooker bees using 2 and store them in $\mathcal{E}^g \leftarrow [er_i^g]_{1\times ph}$.
    Update fit_{best}, min, and gb^g.
  end for
  Store gb^g at the end of $maxit$ iterations.
end for
Combine the stored gb^g (for $1 \le g \le N$) to get an $N \times (N+2)$ matrix.
Extract the first N elements from each row to get an $N \times N$ matrix, $[w_{ij}]_{N\times N}$.
Return $W \leftarrow [w_{ij}]_{N\times N}$.

Algorithm 2 Fitness calculation of particles, i.e. obtaining the predicted time-series using RNN.

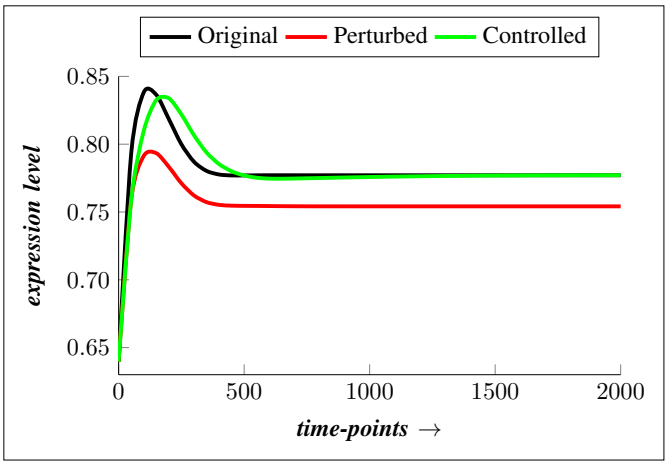
Input: The time-series gene expression dataset (X); the gene being considered (g); and the particle positions $(\mathcal{P}^g)$.
Output: Fitness of the swarm $(\mathcal{E}^g)$.

Extract the number of genes, N, from X.
Extract the number of time-points, tp, from X.
Extract the population size, ps, from $\mathcal{P}^g$.
for $i = 1$ to ps **do**
  Extract $\left[w_{ij}^g\right]_{1\times N}, \beta_i^g$, and τ_i^g from p_i^g.
  for $t = 2$ to tp **do**
    Calculate the predicted expression level, $\tilde{x}_i^g(t)$ of gene g from the original expression level at the previous time-point, i.e. $x_i^g(t-1)$, using (11)–(20).
  end for
  Calculate the fitness of particle p_i^g and store it in er_i^g using (21).
end for
Return $\mathcal{E}^g \leftarrow [er_i^g]_{1\times ps}$.

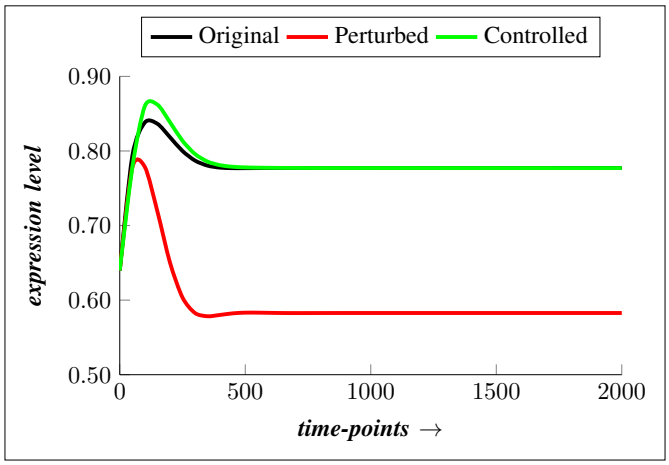
benchmark challenge for **DREAM4** [8], [9] and can be found in the **GNW** [30] database. We have produced the corresponding time-series expression datasets with the help of **GNW** using **DREAM4** settings. The dataset consists of 41 time-points. The swarm size of ABC and the maximum number of iterations has been set to 100 and 1000, respectively. The experiments have been performed on MATLAB 2019b, using a desktop computer with an Intel® Core™ i7 8700 processor and 32GB RAM. The training and validation errors have been presented in Table I.

It is clear from Fig. 3 that the given GRN has two **DG**-s, i.e. gene **G1** and gene **G9**. We have first disturbed gene **G1** using **EP**. This perturbation has affected gene **G4** (i.e. the **MVG**) the maximum, through the pathways, **G1** → **G4**, **G1** → **G3** → **G4**, **G1** → **G3** → **G7** → **G4**. Control action has been imparted to the other unperturbed **DG** (i.e. gene **G9**) to mitigate the harmful effects of **EP**, i.e. to bring back the expression of gene **G4** to its normal level.

In the next case, the reverse has been done, i.e. gene **G9** has been disturbed by **EP** resulting in a disturbance in the



(a) Results for the first approach (Fig. 4(a)).



(b) Results for the second approach (Fig. 4(b)).

Fig. 5: (a) The expression profiles of the *most vulnerable gene*, **MVG** (i.e gene **G4**) for the setup shown in Fig. 4(a). (b) The expression profiles of gene **G4** for the setup shown in Fig. 4(b). The black curve indicates the original unperturbed expression profile of gene **G4**. The red curve represents the expression profile under the influence of external perturbation **EP**. The green curve shows the expression level when the proposed negative feedback control action has been provided to rectify the effects of external perturbation.

TABLE I: The training and validation errors.

Gene	Training Error	Validation Error
G1	0.0384	0.0402
G2	0.0043	0.0083
G3	0.0009	0.0124
G4	0.0053	0.0070
G5	0.0084	0.0093
G6	0.0013	0.0005
G7	0.0005	0.0008
G8	0.0032	0.0042
G9	0.0273	0.0137
G10	0.0036	0.0051

expression level of gene **G4** (the **MVG**) to the maximum extent, via the following paths: **G9** → **G10** → **G3** → **G4**, **G9** → **G10** → **G3** → **G7** → **G4**, **G9** → **G10** → **G4**. In this case, control action has been imposed on gene **G1** to bring back gene **G4** to normalcy. The two cases have been respectively shown in figures 5(a) and 5(b). Here, the PID controller constants, $\mathbf{K_P}$, $\mathbf{K_I}$, and $\mathbf{K_D}$ have been estimated using PSO. Table II presents the estimated values of the controller parameters for both the experimental setups.

The original expression level of the **MVG** (i.e. gene **G4**) has been considered as the set-point or observation point for the proposed PID controller. The error drives the controller, **PIDC**, to generate a control action, **CO**, which is subsequently given to the other unperturbed **DG**. The outcome of the control action given by the controller has been shown in Fig. 5, which represents the expression profiles of the **MVG** (i.e. gene **G4**) under the unperturbed, perturbed, and remediated cases. The expression profiles of gene **G4** for

TABLE II: Estimated values of the coefficients of the proportional, integral, and derivative terms of the proposed PID controller for the network shown in Fig. 3.

Configuration	K_P	K_I	K_D
The First Approach (Fig. 4(a))	-0.389	0.023	-0.240
The Second Approach (Fig. 4(b))	0.338	0.032	1.000

the experimental setups shown in figures 4(a) and 4(b) have been presented in figures 5(a) and 5(b), respectively. It can be observed from Fig. 5 that the proposed PID controller has been able to effectively counteract the reduction in the expression levels of the **MVG** (i.e. gene **G4**) due to the external disturbance (**EP**) acting on either of the two **DG**s, i.e. gene **G1** or gene **G9**.

V. Conclusion

GRNs are quite robust in real-life. However, some extreme environmental conditions or pathogen infections can cause unwanted external disturbances. Such disturbances, if present over a prolonged period of time, can hamper the proper functioning of a GRN, which may cause breakdown of vital biochemical processes potentially leading to disease. In this work, we have proposed a novel methodology based on the concepts of negative feedback control to eliminate the unwanted effects of the external perturbations. For our investigation, we have considered a 10-gene **DREAM4** [8], [9] benchmark network available in the **GNW** [30] database.

We have used the RNN formalism to model the network from the given gene expression data. The model parameter estimation has been done using ABC. Based on the derived

model structure, we have proposed a self-adaptive PID controller to eliminate the unwanted effects of perturbations. The obtained results suggest that the PID controller is able to restore the expression level of the affected gene almost to the original level.

The proposed technique is promising and can lead to personalised drug design endeavours in the future. For this, the biological representation of the synthetic control output needs to be investigated, which may be in the form of externally supplied protein molecule(s). The design of a real-time biological observer also requires further investigation. The scenario considered here is a simple *single-input-single-output* or SISO type. However, in large-scale networks, there may be an opportunity to impart additional control outputs to multiple **DG**-s, thus leading to a *multiple-input-multiple-output* or MIMO type system. In other words, more than one **DG** may be perturbed in a network containing multiple **MVG**-s, which may require more than one of the remaining **DG**-s for controlling purpose. Also, the proposed model is limited by the fact that it requires prior biological knowledge regarding the network structure. All these provide further scope of research in the future.

ACKNOWLEDGEMENT

The authors would like to acknowledge the contribution of the Senior Research Fellowship (NET), awarded by the Council of Scientific & Industrial Research (CSIR), India to the corresponding author (Award No.: 09/028(0974)/2015-EMR-I).

REFERENCES

[1] S. Kimura, K. Ide, A. Kashihara, M. Kano, M. Hatakeyama, R. Masui, N. Nakagawa, S. Yokoyama, S. Kuramitsu, and A. Konagaya, "Inference of S-System Models of Genetic Networks using a Cooperative Coevolutionary Algorithm," *Bioinformatics*, vol. 21, no. 7, pp. 1154–1163, 2004.

[2] M. Newman, *Networks*. Oxford University Press, 2018.

[3] M. Ventresca and D. Aleman, "Evaluation of Strategies to Mitigate Contagion Spread using Social Network Characteristics," *Social Networks*, vol. 35, no. 1, pp. 75–88, 2013.

[4] Y.-Y. Liu, J.-J. Slotine, and A.-L. Barabási, "Controllability of Complex Networks," *Nature*, vol. 473, no. 7346, p. 167, 2011.

[5] Y.-Y. Liu, J.-J. Slotine, and A.-L. Barabási, "Observability of Complex Systems," *Proceedings of the National Academy of Sciences*, vol. 110, no. 7, pp. 2460–2465, 2013.

[6] A. Vinayagam, T. E. Gibson, H.-J. Lee, B. Yilmazel, C. Roesel, Y. Hu, Y. Kwon, A. Sharma, Y.-Y. Liu, N. Perrimon *et al.*, "Controllability analysis of the directed human protein interaction network identifies disease genes and drug targets," *Proceedings of the National Academy of Sciences*, vol. 113, no. 18, pp. 4976–4981, 2016.

[7] Y.-Y. Liu and A.-L. Barabási, "Control Principles of Complex Systems," *Reviews of Modern Physics*, vol. 88, no. 3, p. 035006, 2016.

[8] G. Stolovitzky, D. Monroe, and A. Califano, "Dialogue on Reverse Engineering Assessment and Methods," *Annals of the New York Academy of Sciences*, vol. 1115, no. 1, pp. 1–22, 2007.

[9] A. Greenfield, A. Madar, H. Ostrer, and R. Bonneau, "DREAM4: Combining Genetic and Dynamic Information to Identify Biological Networks and Dynamical Models," *PloS One*, vol. 5, no. 10, p. e13397, 2010.

[10] J. Vohradsky, "Neural Model of the Genetic Network," *Journal of Biological Chemistry*, vol. 276, no. 39, pp. 36 168–36 173, 2001.

[11] D. Karaboga, "An Idea based on Honey Bee Swarm for Numerical Optimisation," Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, Tech. Rep., 2005.

[12] https://en.wikipedia.org/wiki/PID_controller.

[13] P. D'haeseleer, X. Wen, S. Fuhrman, and R. Somogyi, "Linear Modelling of mRNA Expression Levels during CNS Development and Injury," in *Pacific Symposium on Biocomputing*, 1999, pp. 41–52.

[14] I. M. Ong, J. D. Glasner, and D. Page, "Modelling Regulatory Pathways in *E. coli* from Time Series Expression Profiles," *Bioinformatics*, vol. 18, no. suppl_1, pp. S241–S248, 2002.

[15] Z. Bar-Joseph, "Analysing Time Series Gene Expression Data," *Bioinformatics*, vol. 20, no. 16, pp. 2493–2503, 2004.

[16] L. Ironi and D. X. Tran, "Nonlinear and Temporal Multi-Scale Dynamics of Gene Regulatory Networks: A Qualitative Simulator," *Mathematics and Computers in Simulation*, vol. 125, pp. 15–37, 2016.

[17] G. McLachlan, K.-A. Do, and C. Ambroise, *Analysing Microarray Gene Expression Data*. John Wiley & Sons, 2005, vol. 422.

[18] R. Xu and D. Wunsch, "Survey of Clustering Algorithms," *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645–678, 2005.

[19] H. Lähdesmäki, I. Shmulevich, and O. Yli-Harja, "On Learning Gene Regulatory Networks under the Boolean Network Model," *Machine Learning*, vol. 52, no. 1, pp. 147–167, 2003.

[20] N. Friedman, M. Linial, I. Nachman, and D. Pe'er, "Using Bayesian Networks to Analyse Expression Data," *Journal of Computational Biology*, vol. 7, no. 3-4, pp. 601–620, 2000.

[21] A. A. Margolin, I. Nemenman, K. Basso, C. Wiggins, G. Stolovitzky, R. Dalla Favera, and A. Califano, "ARACNE: An Algorithm for the Reconstruction of Gene Regulatory Networks in a Mammalian Cellular Context," *BMC Bioinformatics*, vol. 7, no. 1, p. S7, 2006.

[22] V. A. Huynh-Thu, A. Irrthum, L. Wehenkel, and P. Geurts, "Inferring Regulatory Networks from Expression Data Using Tree-Based Methods," *PLOS ONE*, vol. 5, no. 9, pp. 1–10, 09 2010.

[23] E. O. Voit, *Computational Analysis of Biochemical Systems: A Practical Guide for Biochemists and Molecular Biologists*. Cambridge University Press, 2000.

[24] L. E. Chai, S. K. Loh, S. T. Low, M. S. Mohamad, S. Deris, and Z. Zakaria, "A Review on the Computational Approaches for Gene Regulatory Network Construction," *Computers in Biology and Medicine*, vol. 48, pp. 55–65, 2014.

[25] N. A. Kiani, H. Zenil, J. Olczak, and J. Tegnér, "Evaluating Network Inference Methods in Terms of their Ability to Preserve the Topology and Complexity of Genetic Networks," *Seminars in Cell & Developmental Biology*, vol. 51, no. Supplement C, pp. 44–52, 2016.

[26] H. Bolouri and E. H. Davidson, "Modelling Transcriptional Regulatory Networks," *BioEssays*, vol. 24, no. 12, pp. 1118–1129, 2002.

[27] A. R. Chowdhury and M. Chetty, "Reconstruction of Large-Scale Gene Regulatory Network using S-System Model," *Evolutionary Computation in Gene Regulatory Network Research*, p. 185, 2016.

[28] B. Babayigit and R. Ozdemir, "A Modified Artificial Bee Colony Algorithm for Numerical Function Optimisation," in *2012 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2012, pp. 245–249.

[29] https://en.wikipedia.org/wiki/Feedback#Control_theory.

[30] T. Schaffter, D. Marbach, and D. Floreano, "GeneNetWeaver: *in silico* Benchmark Generation and Performance Profiling of Network Inference Methods," *Bioinformatics*, vol. 27, no. 16, pp. 2263–2270, 2011.