Non-Linear Constrained Optimal Control Problem: A PSO-GA-Based Discrete Augmented Lagrangian Approach

M.A. Rahim[1], Haris M. Khalid[2], S.Z. Rizvi[2] and Amar Khoukhi[2]

[1]*Faculty of Business Administration, University of New Brunswick, Fredericton, E3B 5A3, Canada*
[2]*Department of Systems Engineering, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia*

Correspondence should be addressed to Haris M. Khalid, mhariskh@kfupm.edu.sa

Abstract— This work deals with the optimal control problem which has been proposed to solve using the discrete augmented lagrangian based non-linear programming approach. It is shown that this technique guarantee a satisfactory performance in the face of both optimality by minimizing the energy and maximizing the output. Later on, the optimization has been more effective by using PSO-GA-Based to achieve the optimal value of Lagrange Multipliers and required dynamic parameters and optimally controlling the dynamics. The designed scheme has been successfully tested through extensive simulation. The successful use of the proposed scheme encourages their extension to other physical systems. The proposed scheme is evaluated extensively on a two-tank process used in industry exemplified by a benchmarked laboratory scale coupled-tank system.

Keywords — optimal control, augmented lagrangian, non-linear programming, particle swarm optimization, genetic algorithm, control of two-tank system.

I. INTRODUCTION

Many processes in chemical and biotechnological industries are described by multiple sets of differential and algebraic equations. As such they are difficult to control and optimize in transient regimes if switching between the sets is to be taken into account. The switches can involve different regimes of operation (occurrence of flooding in distillation columns, explosive areas in mixtures of gasses, etc) or external actions (addition of second reactor when production increases, etc). In theoretical and numerical research, optimal control problems have been either treated as cone constrained optimization problems in functional spaces, or studied using some specialized tools. In the first approach, problems of optimal control are placed in a broader framework of optimization problems, and general techniques can be used to solve them, whereas the second approach allows us to take maximal advantage of the specific structure of the problems. Such a situation takes place also in applications of the methods involving Lagrangian for solving numerically optimal control problems.

The paper is organized as follows: in Section II the related work is presented and Optimal Control problem statement is considered in Section III. Section IV gives the system model and description, Section V discusses the implementation and simulation results for all the techniques implemented. Finally some conclusions are given Section VI.

II. RELATED WORKS

There are several approaches to solution of such as dynamic optimization problem. If the process to be optimized can be described accurately enough by piece-wise linear and logic formulation, powerful algorithms in the area of explicit model predictive control exists [1]. If fully nonlinear processes are concerned, original dynamic optimization problem has to be approximated by some simplified formulation. The usual approaches are complete discretisation of state and control variables – orthogonal collocation. Such formulation can be found in [2]. Other possibility is to leave the states intact and approximate only the control variables as piece-wise constant, or with some higher order approximations. This approach is known as control vector parameterization. Here different formulations can be found, depending on how gradients of the resulting nonlinear programming problem (NLP) are calculated. In [3], system of sensitivity equations is formed and the gradients are calculated from its solution. The advantage of this method is easy formulation of the problem and forward integration of both states and sensitivity equations. The drawback of this method lies in a large system of differential equations as each optimized parameter generates a set of differential equations with the same dimension as the number of states of the optimized process. Another possibility, which is pursued in this work, is to calculate the gradients of NLP via optimal control theory using the so-called co-state, or adjoint equations [4]. The advantage is that the number of differential equations is not proportional to number of optimized parameters, but to number of constraints. On the other side, adjoint equations have to be solved in opposite direction of time which makes the implementation more difficult. When dealing with processes comprised of a large number of state equations and only a small number of state-dependent constraints, this approach has favorable properties compared to calculation of sensitivities.

The classical Lagrange-Newton method [5], one of the most efficient numerical methods of solving optimization problems, was developed for problems with equality-type constraints. In this method, the Newton procedure is applied to the first-order optimality system, which has the form of a system of equations. In the case of inequality-type constraints, the first-order optimality system cannot be expressed as an equation. However, it can be expressed as an inclusion, or the so-called generalized equation [6]. It was shown by S.M. Robinson [6] that a Newtontype procedure applied to this general equation is locally quadratically convergent to the solution, provided that a property called strong regularity is satisfied. This approach has been successfully applied to a class of nonlinear cone-constrained optimization problems in infinite dimensional spaces [7][8][9] and optimal control problems subject to control and/or state constraints[10]. Hybridizing the classical optimization technique with evolutionary algorithms can be a good proposed option.

Genetic Algorithms (GAs) are a special type of *evolutionary algorithms,* algorithms that simulate biological processes to solve search and optimization problems. Given a specific problem, potential solutions are typically encoded as bit strings, constituting a *population.* The bit strings are allowed to *reproduce* on the basis of their fitness, thus forming a new population. Iterating this process, the population evolves according to a ‘natural selection and survival of the fittest’ process similar to the one described by Charles Darwin in the Origin of Species. If the GA is implemented successfully, the final population should consist of maximally-fit it individuals, approximating the sought optimal solution. GAs have been implemented for a wide variety of problems, both real-world (e.g. Fault diagnosis, fault tolerant) and abstract (e.g. solving NP-complete problems [11]). The bulk of the GA literature is concerned with practical applications. For a very complete bibliography, see [12], which contains a comprehensive survey. PSO has attracted much attention among researchers and has been used to solve complex optimization problems with wide applications in different fields [13].

Khoukhi et al. has also done considerable work in optimal dynamic modeling and optimal time-energy off-line programming [14-16].

The main aim of this work is to show a detailed derivation of the dynamic Augmented Lagrangian Based optimization based on the optimal control theory. The results obtained will then be presented on an example of benchmarked laboratory scaled two tank level control that exhibits non-linear dynamics. It is the most used prototype applied in the wastewater treatment plant, the petro-chemical plant, and the oil/gas systems. The main contribution of the paper is the implementation of PSO-GA-Based optimal control of augmented lagrangian functions.

III. Problem Statement

Complex optimal control problems require a serious attention for surviving smartly in the process industries. Many units in process industries are described by multiple sets of differential and algebraic equations. Many complex control tasks, for example those associated with controlling an autonomous vehicle to carry out a sequence of maneuvers or with controlling a collection of interacting process units in a process plant, involve two levels of decision making. At the maximum level, it is necessary to achieve a certain maximum output whereas at the same time, at the minimum, it is required to minimize the certain possibility of fault. Hybrid control addresses the problem of integrating the minimization and maximization for decision making in this context. The early optimal control of mission critical systems becomes highly crucial for preventing failure of equipment, loss of productivity and profits, management of assets, reduction of shutdowns.

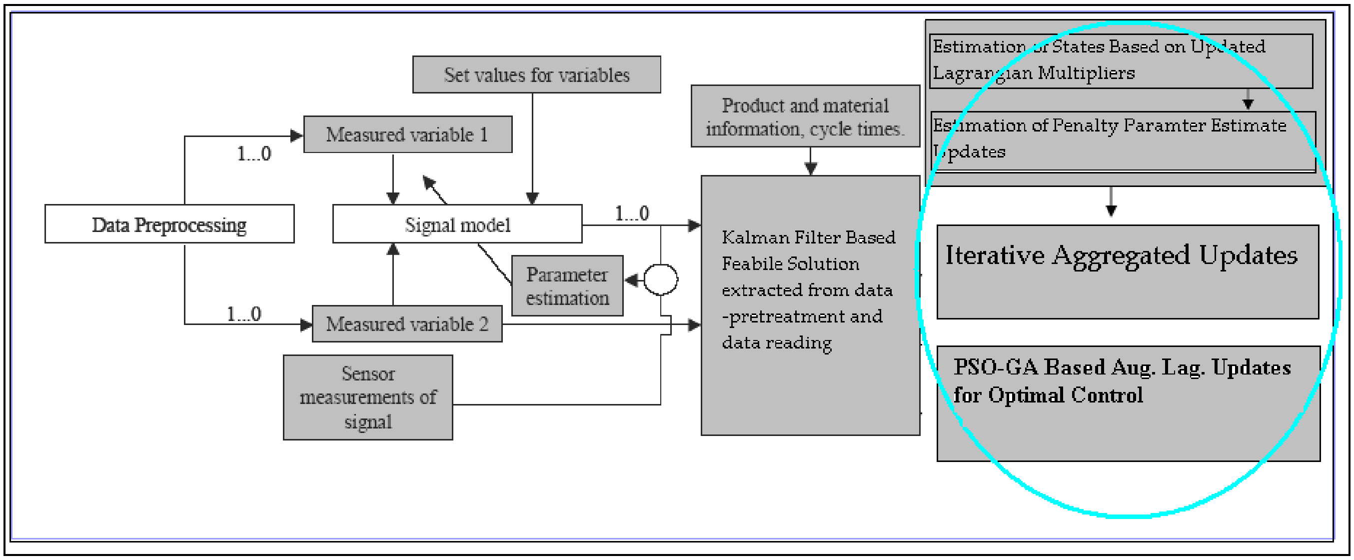


Figure 1. Implementation plan for the evaluation of the proposed scheme

The proposed scheme has been evaluated on the above- cited process control system. The scheme is carried out by jointly interpreting model outputs. The implementation plan for the proposed scheme is shown in the Fig 1. It should be noted that PSO-GA-Based NLP Constrained optimization is applied for the optimal control of the system.

IV. System Model and Description

A. System Description

The Benchmarked laboratory-scale process control system has been used to collect data. The data has been collected at a sampling time of 50 milli-second. The different data sets have been generated for PI Control-based water level control. Different fault scenarios have also been considered for the generation of the data sets.

B. Experimental Setup

Process Data has been generated through an experimental setup as shown in Figure 2. A two tank system has been used in order to collect the data with the introduction of actuator, and sensor faults through the system as can be seen in the labview circuit window. An amplified voltage of 18 volts has been used to handle the controller effectively for the changes/fluctuation produced in the system. So, the fault diagnosis was done over here in a closed loop identification where in the same time, the controller is suppressing the faults.

C. Process Data Collection and Description

The process data has been collected at 50 milli-seconds sampling time. The main objective of the benchmarked dual-tank system is to reach a reference height of 200 ml of the second tank. During this process, several faults have been introduced such as the leakage faults, sensor faults and actuator faults. Leakage faults have been introduced through the pipe clogs of the system, knobs between the first and the second tank etc. Sensor faults have been introduced by introducing a gain in the circuit as if there is a fault in the level sensor of the tank. Actuator faults have been introduced by introducing a gain in the setup for the actuator that comprises of the motor and pump. A PI controller has been employed in order to reach the desired reference height. Due to the inclusion of faults, the controller was finding it difficult to reach the desired level. For this reason, the power of the motor has been increased from 5 volts to 18 volts in order to provide it the maximum throttle to reach the desired level. In doing so, the actuator performed well in achieving its desired level but it also suppressed the faults of the system. So, it made the task of detecting the faults. After the collection of data, techniques such as settling time, steady state value, and coherence spectra can help us to give an insight of the fault.

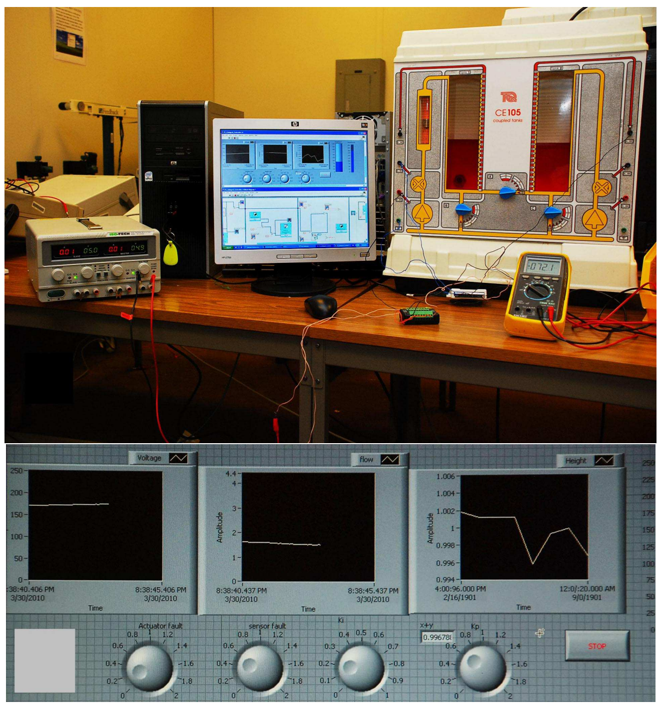


Figure 2 A – The two tank system interfaced with the Labview through a DAQ and the amplifier for the magnified voltage , B – The labview setup of the apparatus including the ciruit window and the block diagram of the experiment.

D. Model of the Coupled Tank System

The physical system under evaluation is formed of two tanks connected by a pipe. The leakage is simulated in the tank by opening the drain valve. A DC motor-driven pump supplies the fluid to the first tank and a PI controller is used to control the fluid level in the second tank by maintaining the level at a specified level, as shown in Fig 3.

A step input is applied to the dc motor- pump system to fill the first tank. The opening of the drainage valve introduces a leakage in the tank. Various types of leakage faults are introduced and the liquid height in the second tank, H_2, and the inflow rate, Q_i, are both measured. The National Instruments LABVIEW package is employed to collect these data.

A benchmark model of a cascade connection of a dc motor and a pump relating the input to the motor, *u*, and the flow, Q_i, is a first-order system:

$$\dot{Q}_i = -a_m Q_i + b_m \phi(u) \quad (1)$$

where a_m and b_m are the parameters of the motor-pump system and $\phi(u)$ is a dead-band and saturation type of nonlinearity. It is assumed that the leakage Q_ℓ occurs in tank 1 and is given by:

$$Q_\ell = C_{d\ell}\sqrt{2gH_1} \quad (2)$$

With the inclusion of the leakage, the liquid level system is modeled by:

$$A_1\frac{dH_1}{dt} = Q_i - C_{12}\varphi(H_1 - H_2) - C_\ell\varphi(H_1) \qquad (3)$$

$$A_2\frac{dH_2}{dt} = C_{12}\varphi(H_1 - H_2) - C_0\varphi(H_2) \quad (4)$$

where $\varphi(.) = sign(.)\sqrt{2g(.)}$, $Q_\ell = C_\ell\varphi(H_1)$ is the leakage flow rate, $Q_0 = C_0\varphi(H_2)$ is the output flow rate, H_1 is the height of the liquid in tank 1, H_2 is the height of the liquid in tank 2, A_1 and A_2 are the cross-sectional areas of the 2 tanks, g=980 $cm/\sec^2$ is the gravitational constant, C_{12} and C_o are the discharge coefficient of the inter-tank and output valves, respectively.

The model of the two-tank fluid control system, shown above in Fig. 3, is of a second order and is nonlinear with a smooth square-root type of nonlinearity. For design purposes, a linearized model of the fluid system is required and is given below in (5) and (6):

$$\frac{dh_1}{dt} = b_1 q_i - (a_1 + \alpha)h_1 + a_1 h_2 \quad (5)$$

$$\frac{dh_2}{dt} = a_2 h_1 - (a_2 - \beta)h_2 \qquad (6)$$

where h_1 and h_2 are the increments in the nominal (leakage-free) heights H_1^0 and H_2^0:

$$b_1 = \frac{1}{A_1}, \quad a_1 = \frac{C_{db}}{2\sqrt{2g(H_1^0 - H_2^0)}}, \quad \beta = \frac{C_0}{2\sqrt{2gH_2^0}}, a_2 = a_1 + \frac{C_{do}}{2\sqrt{2gH_2^0}} \quad \alpha = \frac{C_{d\ell}}{2\sqrt{2gH_1^0}}$$

and the parameter α indicates the amount of leakage.

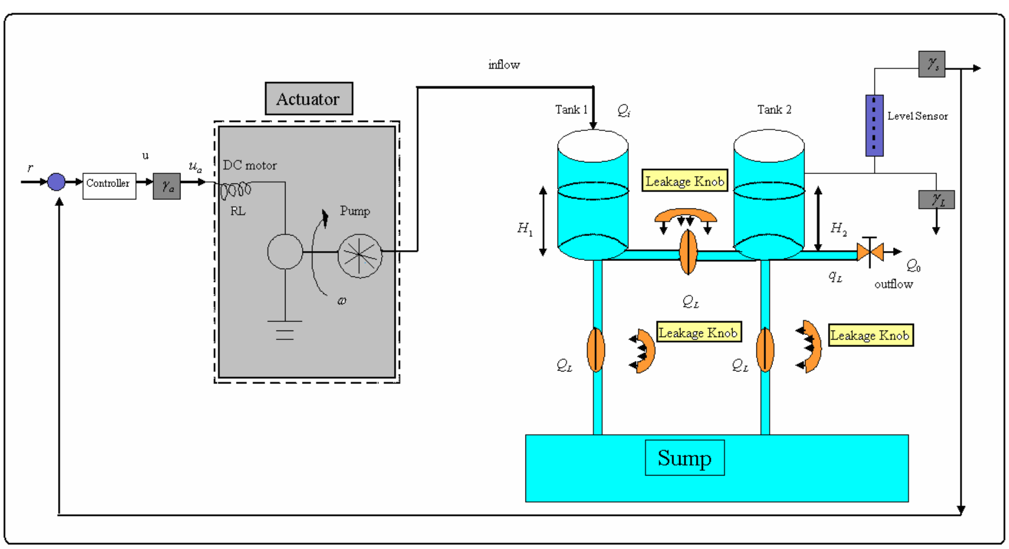


Fig. 3 Process control system: A Lab-scale two-tank system

A PI controller, with gains k_p and k_I, is used to maintain the level of the Tank 2 at the desired reference input r as:

$$\dot{x}_3 = e = r - h_2$$
$$u = k_p e + k_I x_3 \quad (7)$$

The linearized model of the entire system formed by the motor, pump, and the tanks is given by:

$$\dot{x} = Ax + Br \qquad y = Cx \quad (8)$$

where

$$x = \begin{bmatrix} h_1 \\ h_2 \\ x_3 \\ q_i \end{bmatrix}, \quad A = \begin{bmatrix} -a_1 - \alpha & a_1 & 0 & b_1 \\ a_2 & -a_2 - \beta & 0 & 0 \\ -1 & 0 & 0 & 0 \\ -b_m k_p & 0 & b_m k_I & -a_m \end{bmatrix},$$

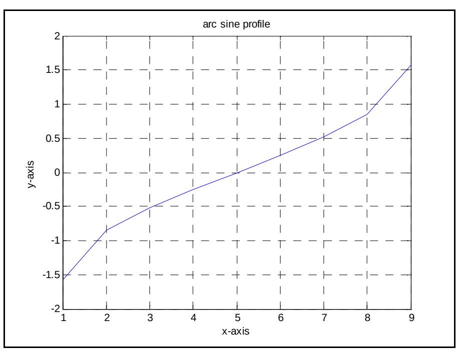
$$B = \begin{bmatrix} 0 & 0 & 1 & b_m k_p \end{bmatrix}^T, \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \quad (9)$$

where q_i, q_ℓ, q_0, h_1 and h_2 are the increments in Q_i, Q_ℓ, Q_o, H_1^0 and H_2^0, respectively, the parameters a_1 and a_2 are associated with linearization whereas the parameters α and β are respectively associated with the leakage and the output flow rate, i.e. $q_\ell = \alpha h_1$, $q_o = \beta h_2$.

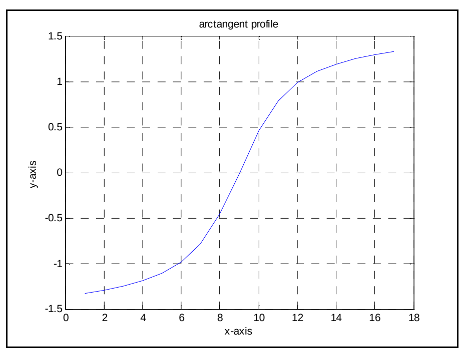
Proposition: During the implementation process, sign(.) can be approximated with arc tangent. A relationship for approximation can be as follows:

$$sign(x) = \arctan\left(\frac{x}{\sqrt{1 - x * x}}\right), \text{ where } x < 1$$

Likewise, after approximation, the profiles can be as follows: (See Fig. 4) where (a) representing the *sign(.)* profile and (b) representing the *arctangent* profile for a certain set of numbers.



(a)



(b)

Fig. 4 (a) sign(.) profile and (b) Arctangent profile for a certain set of numbers

V. Implementation and Simulation Results

The tasks of our optimal control scheme, are executed with an increasing precision accompanied with a more detailed optimality picture. Firstly, the data collected from the plant has been initialized and the parameters are being optimized which comprises of the pre-processing and normalization of the data and applying the Kalman filter to obtain the feasible solution. Then, the estimation of states based on updated Lagrangian multipliers, and the penalty parameter estimate updates are being done followed by the iterative aggregated updates. To have a cross-optimal solution, the functions have been optimized using PSO-GA-Based approach. The flowchart of the implementation can be seen in the Figure 6.

A. Kalman Filter-Based Feasible solution for Optimal Control

Using the leakage-free model together with the covariance of the measurement noise, R, and the plant noise covariance, Q, the Kalman filter model is finally derived. As it is difficult to obtain an estimate of the plant covariance, Q, a number of experiments were performed under different plant scenarios to tune the Kalman gain, K_0 obtained by: $\hat{x}(k+1) = A_0\hat{x}(k) + B_0u(k-d) + K_0\left(y(k) - C_0\hat{x}(k)\right)$ and $e(k) = y(k) - C_0\hat{x}(k)$. The initial feasible solution for the Kalman filter filter can be shown in Figure 5 where height profile is shown in the upper window and the kalman filter residual analysis is shown in the lower window.

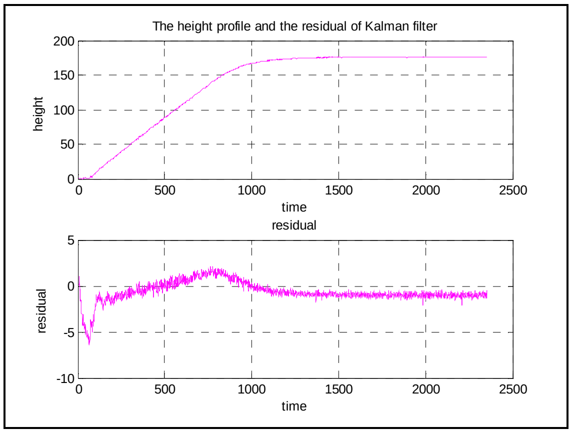


Figure 5. Kalman filter residual analysis considered as feasible solution

B. Cost Function of the System

$$L_p = -\frac{1}{2}H_1^{\,2}(t) - \frac{1}{2}H_2^{\,2}(t) + \frac{1}{2}u^2 \tag{10}$$

Constraints:

$$H_1(t+1) = H_1(t) + [\frac{1}{A_1}(Q_i - C_{12}\arctan(H_1 - H_2)\sqrt{2g(H_1 - H_2)} - C_l\arctan(H_1)\sqrt{2gH_1})]*T_s \tag{11}$$

$$H_2(t+1) = H_2(t) + [\frac{1}{A_2}(C_{12}\arctan(H_1 - H_2)\sqrt{2g(H_1 - H_2)} - C_0\arctan(H_2)\sqrt{2gH_2})]*T_s \tag{12}$$

$$Q_i(t+1) = Q_i(t) + [b_m\arctan(u)\sqrt{2gu(t)} - a_mQ_i(t)]*T_s \tag{13}$$

$$Q_L - C_{dl}\sqrt{2gH_1} = 0 \tag{14}$$

$$e - r + H_2 = 0 \tag{15}$$

$$u - k_pe - k_Ix_3 = 0 \tag{16}$$

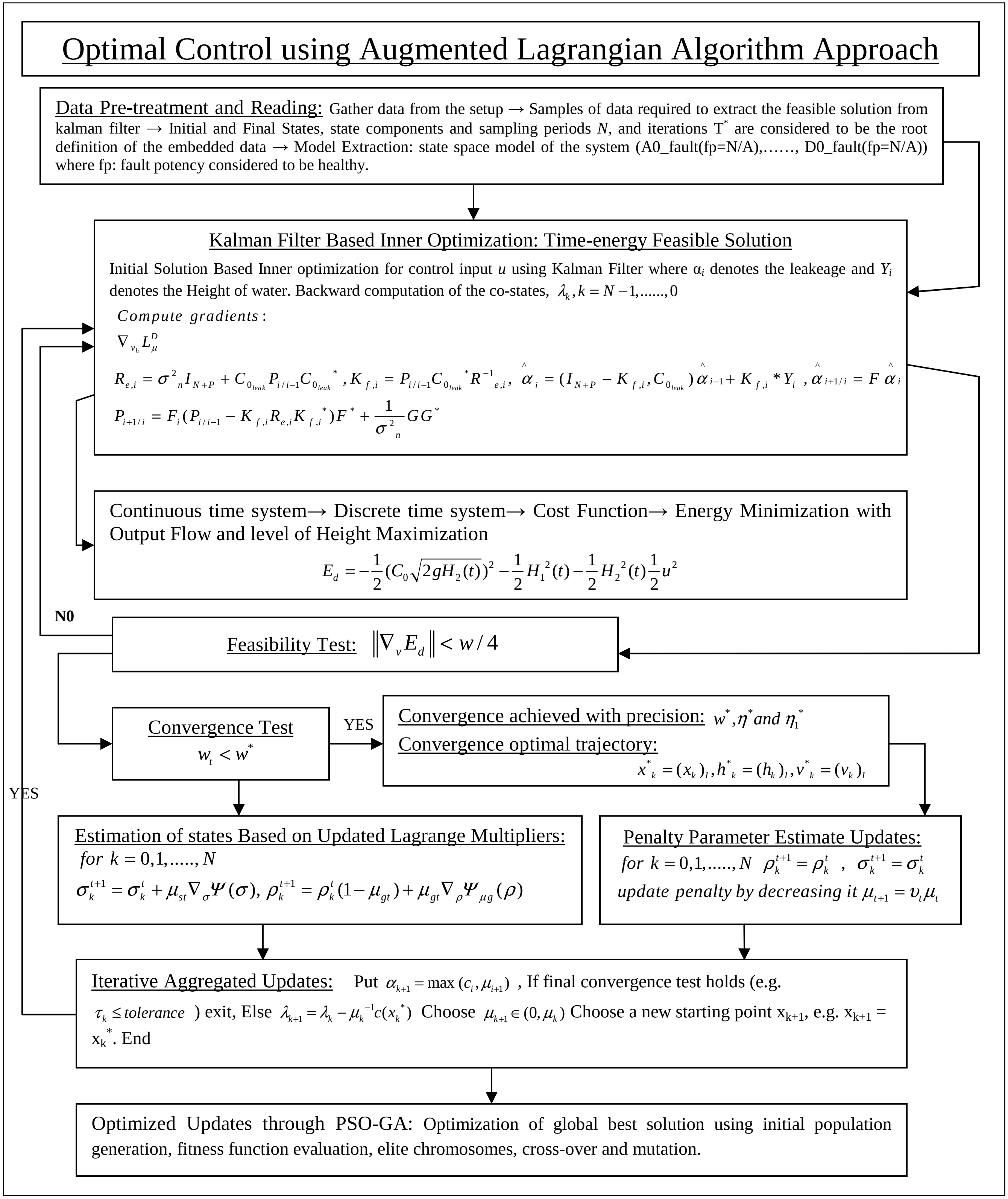


Figure 6. Flowchart for the implementation of the proposed scheme

Bounds:

1) $Q_i \geq 0,\ Q_i \leq 200$
2) $Q_0 \geq 0,\ Q_0 \leq 200$
3) $0 \leq H_1 \leq 200$
4) $50 \leq H_2 \leq 200$

4) $u \geq 5, \; u \leq 18$

5) $C_0 \geq 0.25, \; C_0 \leq 1.0$

6) $C_1 \geq 0.25, \; C_1 \leq 1.0$

7) $a_m \geq 0, \; a_m \leq 18$

8) $a_m \geq 0, \; b_m \leq 18$

Where

Q_0 *represents the output flow rate.*

k_p *represents the proportional gain of the controller.*

k_I *represents the integral gain of the controller.*

H_1 *represents the Height of the first tank.*

H_2 *represents the Height of the second tank.*

Q_i *represents the Initial flow of water*

a_m *represents the motor system*

b_m *represents the pump system*

C_{12} *represents the Discharge coefficient of Inter-tank*

C_0 *represents the Discharge coefficient of output valves*

C. Augmented Lagrangian Method for Optimal Control

C.I. Lagrangian Algorithmic Framework of Optimal Control

The augmented lagrangian multiplier method combines the lagrange multiplier and the penalty function methods. The augmented Lagrangian function is given as:

$$A(X,\alpha,r_k) = f(X) + \sum_{j=1}^{p} \alpha_j h_j(X) + r_k \sum_{j=1}^{p} h^2{}_j(X)$$

Where r_k is the penalty parameter. It can be noted that the function A reduces to the Lagrangian if r_k=0.

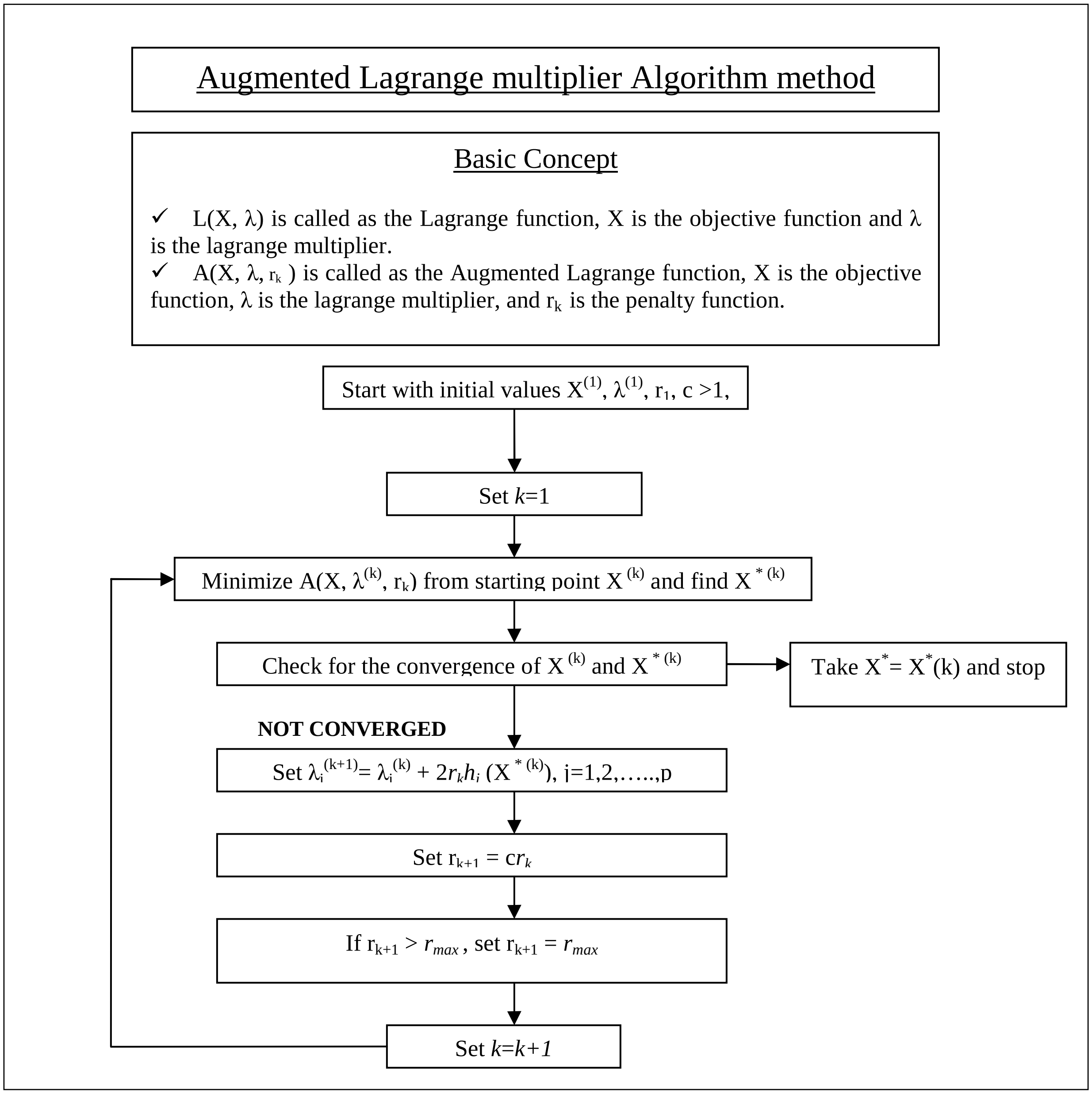


Figure 7. Flowchart for the augmented largrangian multiplier Algorithm method for optimal control

Given $\mu_0 > 0, x_0$ *and* λ_0 and final tolerance *tolerance,*

For k = 0,1,2,…….

Starting with x_k minimize $L_A(x, \lambda_k; \mu_k)$ terminating when

$\|\nabla L_A(x, \lambda_k; \mu_k)\| \le \tau_k$

Call the result x_k^*

If final convergence test holds (e.g. $\tau_k \le tolerance$) exit

Else

$\lambda_{k+1} = \lambda_k - \mu_k^{-1} c(x_k^*)$

Choose $\mu_{k+1} \in (0, \mu_k)$

Choose a new starting point x_{k+1}, e.g. $x_{k+1} = x_k^*$.

End

End

This algorithmic framework allows a gentler decrease in μ.

D. Augmented Lagrangian Implementation in Optimal Control

The augmented lagrangian implementation can be made by the implementation while considering L_p as the main function and α as Lagrange multipliers and r_k as augmented Lagrangian multiplier. In order to cater for the constraints in this minimization, we need to allocate them the Lagrange multipliers α and r_k as augmented Lagrangian multiplier, where $\alpha_i \geq 0$ $\forall_i$ and $r_k \geq 0$ respectively:

$$\begin{aligned}
L_p = &-\frac{1}{2}H_1(t)^2 - \frac{1}{2}H_2(t)^2 + \frac{1}{2}u(t)^2 - a_1(t)*[H_1(t+1) - H_1(t) - [\frac{1}{A_1}(Q_i - C_{12}\arctan(H_1 - H_2)\sqrt{2g(H_1 - H_2)} \\
&- C_l \arctan(H_1)\sqrt{2gH_1})]*T_s] - r_k*[*[H_1(t+1) - H_1(t) - [\frac{1}{A_1}(Q_i - C_{12}\arctan(H_1 - H_2)\sqrt{2g(H_1 - H_2)} \\
&- C_l \arctan(H_1)\sqrt{2gH_1})]*T_s]^2 - a_2(k)*[H_2(t+1) - H_2(t) - [\frac{1}{A_2}(C_{12}\arctan(H_1 - H_2)\sqrt{2g(H_1 - H_2)} \\
&- C_0 \arctan(H_2)\sqrt{2gH_2})]*T_s] - r_k*[H_2(t+1) - H_2(t) - [\frac{1}{A_2}(C_{12}\arctan(H_1 - H_2)\sqrt{2g(H_1 - H_2)} \\
&- C_0 \arctan(H_2)\sqrt{2gH_2})]*T_s]^2 - a_3(k)*[Q_i(t+1) - Q_i(t) - [b_m \arctan(u)\sqrt{2gu(t)} - a_m Q_i(t)]*T_s] \\
&- r_k*[Q_i(t+1) - Q_i(t) - [b_m \arctan(u)\sqrt{2gu(t)} - a_m Q_i(t)]*T_s]^2 - a_4(k)*[Q_L - C_{dl}\sqrt{2gH_1}] - r_k*[Q_L - C_{dl}\sqrt{2gH_1}]^2 \\
&- a_5(k)*[e - r + H_2] - r_k*[e - r + H_2]^2 - a_6(k)*[u - k_p e - k_I x_3] - r_k*[u - k_p e - k_I x_3]^2 - h_{n_1}*[Q_i(t)] - r_k*[Q_i(t)]^2 \\
&- h_{n_2}*[200 - Q_i(t)] - r_k*[200 - Q_i(t)]^2 - h_{n_3}*[Q_0(t)] - r_k*[Q_0(t)]^2 - h_{n_4}*[200 - Q_0(t)] - r_k*[200 - Q_0(t)]^2 \\
&- h_{n_5}*[H_1(t)] - r_k*[H_1(t)]^2 - h_{n_6}*[200 - H_1(t)] - r_k*[200 - H_1(t)]^2 - h_{n_7}*[H_2(t)] - r_k*[H_2(t)]^2 \\
&- h_{n_8}*[200 - H_2(t)] - r_k*[200 - H_2(t)]^2 - h_{n_9}*[u(t)] - r_k*[u(t)]^2 - h_{n_{10}}*[18 - u(t)] - r_k*[18 - u(t)]^2 \\
&- h_{n_{11}}*[a_m] - r_k*[a_m]^2 - h_{n_{12}}*[18 - a_m] - r_k*[18 - a_m]^2 - h_{n_{13}}*[b_m] - r_k*[b_m]^2 - h_{n_{14}}*[18 - b_m] - r_k*[18 - b_m]^2
\end{aligned} \tag{20}$$

Where $\alpha_1 \to \alpha_6$ are the lagrange multipliers and $h_{n_1} \to h_{n_{14}}$ are the penalty coefficients.

Now minimizing L_p with respect to H_1, H_2, Q_i, Q_0, a_m, b_m, $u(t)$, $H_1(t+1)$, $H_2(t+1)$, $Q_i(t+1)$ and setting the derivatives to zero.

Minimizing L_p with respect to H_1 gives:

$$\frac{\partial L_p}{\partial H_1} \Rightarrow -H_1(t) - \alpha_1(-1-[\frac{1}{A_1}(\frac{-C_{12}}{1+((H_1(t)-H_2(t))\sqrt{2g(H_1(t)-H_2(t)}^2}.(\sqrt{2g(H_1(t)-H_2(t))} + \frac{((H_1(t)-H_2(t)).g}{\sqrt{2g(H_1(t)-H_2(t))}})$$
$$-(\frac{-C_L}{1+(H_1(t)\sqrt{2g(H_1(t))}^2}.(\sqrt{2g(H_1(t))} + \frac{(H_1(t).g}{\sqrt{2g(H_1(t))}}) - 2r_k[(H_1(t+1)-H_1(t)-[\frac{1}{A_1}(Q_i - C_{12}\arctan(H_1-H_2)$$
$$\sqrt{2g(H_1-H_2)} - C_l\arctan(H_1)\sqrt{2gH_1})]*T_s] + \frac{\alpha_2}{A_2}(\frac{C_{12}}{1+((H_1(t)-(H_2(t))\sqrt{2g(H_1(t)-H_2(t))}^2}).(\sqrt{2g(H_1(t)-H_2(t))} - \quad (21)$$
$$\frac{((H_1(t)-H_2(t)).g}{\sqrt{2g(H_1(t)-H_2(t))}})) - 2r_k(H_2(t+1)-H_2(t)-[\frac{1}{A_2}(C_{12}\arctan(H_1-H_2)\sqrt{2g(H_1-H_2)} - C_0\arctan(H_2)\sqrt{2gH_2})]*T_s)]$$
$$+\alpha_4(\frac{C_{dl}}{\sqrt{2gH_1}}) - 2r_k[(Q_L - C_{dl}\sqrt{2gH_1})] - h_{n_5}(1) - 2r_k[(H_1)] - h_{n_6}(1) - 2r_k[(200-H_1)] = 0$$

Minimizing L_p with respect to H_2 gives:

$$\frac{\partial L_p}{\partial H_2} \Rightarrow -H_2(t) - \frac{\alpha_1}{A_1}(\frac{-C_{12}}{1+((H_1(t)-H_2(t))\sqrt{2g(H_1(t)-H_2(t)}^2}.(-\sqrt{2g(H_1(t)-H_2(t))} - \frac{((H_1(t)-H_2(t)).g}{\sqrt{2g(H_1(t)-H_2(t))}}) -$$
$$2r_k[(H_1(t+1)-H_1(t)-[\frac{1}{A_1}(Q_i - C_{12}\arctan(H_1-H_2)\sqrt{2g(H_1-H_2)} - C_l\arctan(H_1)\sqrt{2gH_1})]*T_s] +$$
$$\frac{\alpha_2}{A_2}(-1+\frac{C_{12}}{1+((H_1(t)-(H_2(t))\sqrt{2g(H_1(t)-H_2(t))}^2}).(-\sqrt{2g(H_1(t)-H_2(t))} - \frac{((H_1(t)-H_2(t)).g}{\sqrt{2g(H_1(t)-H_2(t))}})) - \quad (22)$$
$$\frac{C_0}{1+[(H_2(t)\sqrt{2gH_2(t)}]^2}.(\sqrt{2gH_2(t)} + \frac{(H_2(t)).g}{\sqrt{2g(H_2(t))}}) - 2r_k(H_2(t+1)-H_2(t)-[\frac{1}{A_2}(C_{12}\arctan(H_1 -$$
$$H_2)\sqrt{2g(H_1-H_2)} - C_0\arctan(H_2)\sqrt{2gH_2})]*T_s)] - \alpha_6(1) - 2r_k[(e-r+H_2)] - h_{n_7}(1) - 2r_k[(H_2)] +$$
$$h_{n_8}(1) - 2r_k[(200-H_2)] = 0$$

Minimizing L_p with respect to Q_i gives:

$$\frac{\partial L_p}{\partial Q_i(t)} \Rightarrow -Q_i(t) + Q_0(t)) + Q_i(t) - \alpha_3(1+a_m) - 2r_k[(Q_i(t+1) - Q_i(t) - [b_m\arctan(u)\sqrt{2gu(t)} - a_m Q_i(t)]*T_s)]$$
$$-h_{n_1}(1) - 2r_k[(Q_i)] + h_{n_2}(1) - 2r_k[(200-Q_i)] = 0 \quad (23)$$

Minimizing L_p with respect to Q_0 gives:

$$\frac{\partial L_p}{\partial Q_0} \Rightarrow (Q_i(t) - Q_0(t)) - h_{n_3} - 2r_k[(Q_0)] + h_{n_4} - 2r_k[(200-Q_0)] = 0 \quad (24)$$

Minimizing L_p with respect to a_m gives:

$$\frac{\partial L_p}{\partial a_m} \Rightarrow a_m - \alpha_3(Q_i(t)) - 2r_k[(Q_i(t+1) - Q_i(t) - [b_m\arctan(u)\sqrt{2gu(t)} - a_m Q_i(t)]*T_s)] -$$
$$h_{n_{11}} - 2r_k[(a_m)] + h_{n_{12}} - 2r_k[(18-a_m)] = 0 \quad (25)$$

Minimizing L_p with respect to b_m gives:

$$\frac{\partial L_p}{\partial b_m} \Rightarrow b_m - \alpha_3(-\arctan u(t)\sqrt{2gu(t)}) - 2r_k(H_2(t+1) - H_2(t) - [\frac{1}{A_2}(C_{12}\arctan(H_1 - H_2)\sqrt{2g(H_1 - H_2)} - C_0\arctan(H_2)\sqrt{2gH_2})]*T_s)] - h_{n_{13}} - 2r_k[(b_m)] + h_{n_{14}} - 2r_k[(18 - b_m)] = 0 \quad (26)$$

Minimizing L_p with respect to *u(t)* gives:

$$\frac{\partial L_p}{\partial u(t)} \Rightarrow u(t) - \alpha_3(-\frac{b_m}{1+(u(t)\sqrt{2gu(t)})^2}.\sqrt{2gu(t)} + \frac{u(t).g}{\sqrt{2gu(t)}} - 2r_k[(Q_i(t+1) - Q_i(t) - [b_m\arctan(u)\sqrt{2gu(t)} - a_mQ_i(t)]*T_s)] - h_{n_9} - 2r_k[(u(t))] + h_{n_{10}} - 2r_k[(18 - u(t))] = 0 \quad (27)$$

D.I. Solving the derivatives:

Considering all the lagrange multipliers α , minimizing L_p with respect to the states gives (See-Equation 28-33):

$$Q_0(t) + \alpha_3 - \alpha_3(a_m) - h_{n_1} + h_{n_2} = 0 \quad (28)$$

$$a_m - \alpha_3(Q_i(t)) - h_{n_{11}} + h_{n_{12}} = 0 \quad (29)$$

$$b_m + \alpha_3(\arctan u(t)\sqrt{2gu(t)} - h_{n_{13}} + h_{n_{14}} = 0 \quad (30)$$

$$u + \alpha_3(\frac{b_m}{1+(u(t)(\sqrt{2gu(t)})^2})(\sqrt{2gu(t)} + \frac{u(t).g}{\sqrt{2gu(t)}}) - h_{n_9} + h_{n_{10}} = 0 \quad (31)$$

$$H_2(t) - \alpha_6 - h_{n_7} + h_{n_8} = 0 \quad (32)$$

$$H_1(t) - h_{n_5} - h_{n_6} = 0 \quad (33)$$

$$\frac{\partial L_p}{\partial H_1(t+1)} \Rightarrow -\alpha_1 = 0 \quad (34)$$

$$\frac{\partial L_p}{\partial H_2(t+1)} \Rightarrow -\alpha_2 = 0 \quad (35)$$

$$\frac{\partial L_p}{\partial Q_i(t+1)} \Rightarrow -\alpha_3 = 0 \quad (36)$$

Substituting, Equation # (21-36) gives a new formulation (Equation 31, which being dependent on α , we need to maximize:

$$Q_0(t) - h_{n_1} + h_{n_2} = 0 \quad (37)$$

$$a_m - h_{n_{11}} + h_{n_{12}} = 0 \quad (38)$$

$$b_m - h_{n_{13}} + h_{n_{14}} = 0 \quad (39)$$

$$Q_i(t) - Q_0(t) - h_{n_3} + h_{n_4} = 0 \quad (40)$$

$$u + \alpha_5 - h_{n_9} + h_{n_{10}} = 0 \quad (41)$$

$$H_1(t) - h_{n_5} - h_{n_6} + \alpha_4(\frac{gC_{dl}}{\sqrt{2gH_1}}) = 0 \quad (42)$$

$$H_2(t) - \alpha_6 - h_{n_7} + h_{n_{10}} = 0 \quad (43)$$

$$\alpha_1 = H_1(t+1) - H_1(t) - [\frac{1}{A_1}(Q_i - C_{12}\arctan(H_1 - H_2)\sqrt{2g(H_1 - H_2)} - C_l \arctan(H_1)\sqrt{2gH_1})] \quad (44)$$

$$\alpha_2 = H_2(t+1) - H_2(t) - [\frac{1}{A_2}(C_{12}\arctan(H_1 - H_2)\sqrt{2g(H_1 - H_2)} - C_0 \arctan(H_2)\sqrt{2gH_2}) \quad (45)$$

$$\alpha_3 = Q_i(t+1) - Q_i(t) - [b_m \arctan(u)\sqrt{2gu(t)} - a_m Q_i(t)] \quad (46)$$

$$C_{dl} + \alpha_4(\sqrt{2gH_1}) = 0 \quad (47)$$

Thus, the implication of lagrangian without implication of augmentation after the insertion of all the constraint derivates is as follows: (See Equation # 48-68).

$$L_p = -\frac{1}{2}(C_0\sqrt{2gH_2(t)})^2 - H_1(t)[\frac{1}{2}H_1(t) + 200] - H_2(t)[\frac{7}{2}H_2(t) + e - r - 200] - \frac{1}{2}[Q_i(t) - Q_0^{\,2}(t) + \frac{1}{2}Q_i^{\,2}(t)$$
$$-4Q_i(t)Q_0(t) + 200Q_i(t) + 2Q_0^{\,2}(t) + 18a_m + 18b_m - u(t)[\frac{1}{2}u(t) + k_p e + k_I x_3 - 18] + C_{dl}(\frac{Q_L}{\sqrt{2gH_1}} - \frac{1}{2}C_{dl}) -$$
$$\alpha_4(\frac{200gC_{dl}}{\sqrt{2gH_1(t)}}) + \alpha_5(18 - u(t)) + \alpha_6(2H_2(t) - 200) + \alpha_7(Q_i(t) - 200) + \alpha_8(-Q_i(t)) + \alpha_9(-200 + Q_0(t)) \quad (48)$$
$$+\alpha_{10}(-Q_0(t)) + \alpha_{11}(-H_1(t) + 200) + \alpha_{12}(H_1(t)) + \alpha_{13}(2H_2(t) + e - r - 200) + \alpha_{14}(-2H_2(t) - e + r)$$
$$+\alpha_{15}(-2u(t) + k_p e + k_I x_3 - 18 + u(t)) + \alpha_{16}(u(t) - k_p e - k_I x_3 - u(t)) + \alpha_{17}(a_m - 18) + \alpha_{18}(-a_m) + \alpha_{19}(b_m - 18)$$
$$+\alpha_{20}(-b_m)$$

Where,

$$\alpha_1 = H_1(t+1) - H_1(t) - [\frac{1}{A_1}(Q_i - C_{12}\arctan(H_1 - H_2)\sqrt{2g(H_1 - H_2)} - C_l \arctan(H_1)\sqrt{2gH_1})], \quad (49)$$

$$\alpha_2 = H_2(t+1) - H_2(t) - [\frac{1}{A_2}(C_{12}\arctan(H_1 - H_2)\sqrt{2g(H_1 - H_2)} - C_0 \arctan(H_2)\sqrt{2gH_2}), \quad (50)$$

$$\alpha_3 = Q_i(t+1) - Q_i(t) - [b_m \arctan(u)\sqrt{2gu(t)} - a_m Q_i(t)], \quad (51)$$

$$\alpha_4 = \frac{-C_{dl}}{\sqrt{2gH_1}}, \quad (52)$$

$$\alpha_5 = \alpha_{15} - \alpha_{16} - u(t), \quad (53)$$

$$\alpha_6 = H_2(t) - \alpha_{13} + \alpha_{14}, \quad (54)$$

$$h_{n_1} = Q_0(t) + h_{n_2}, \tag{55}$$

$$h_{n_2} = h_{n_1} - Q_0(t), \tag{56}$$

$$h_{n_3} = Q_i(t) - Q_0(t) + h_{n_4}, \tag{57}$$

$$h_{n_4} = Q_0(t) - Q_i(t) + h_{n_3}, \tag{58}$$

$$h_{n_5} = H_1(t) - h_{n_6} + \alpha_4(\frac{gC_{dl}}{\sqrt{2gH_1}}) \tag{59}$$

$$h_{n_6} = H_1(t) - h_{n_5} + \alpha_4(\frac{gC_{dl}}{\sqrt{2gH_1}}) \tag{60}$$

$$h_{n_7} = H_2(t) - \alpha_6 + h_{n_8} \tag{61}$$

$$h_{n_8} = -H_2(t) + \alpha_6 + h_{n_7} \tag{62}$$

$$h_{n_9} = u(t) + \alpha_5 + h_{n_{10}} \tag{63}$$

$$h_{n_{10}} = h_{n_9} - \alpha_5 - u(t) \tag{64}$$

$$h_{n_{11}} = a_m + h_{n_{12}} \tag{65}$$

$$h_{n_{12}} = h_{n_{11}} - a_m \tag{66}$$

$$h_{n_{13}} = b_m + h_{n_{14}} \tag{67}$$

$$h_{n_{14}} = h_{n_{13}} - b_m \tag{68}$$

Now, the final formulation, dependent on *α, penalty function* and augmented lagrange multliplier, r_k yields (See Equation # 69):

$$
\begin{aligned}
L_p &= -\frac{1}{2}(C_0\sqrt{2gH_2(t)})^2 - H_1(t)[\frac{1}{2}H_1(t)+200] - H_2(t)[\frac{7}{2}H_2(t)+e-r-200] - \frac{1}{2}[Q_i(t)-Q_0{}^2(t)+\frac{1}{2}Q_i{}^2(t) \\
&-4Q_i(t)Q_0(t)+200Q_i(t)+2Q_0{}^2(t)+18a_m+18b_m-u(t)[\frac{1}{2}u(t)+k_pe+k_Ix_3-18]+C_{dl}(\frac{Q_L}{\sqrt{2gH_1}}-\frac{1}{2}C_{dl})-\alpha H^T-h_nI^T-2r_kA^T
\end{aligned}
$$

$$where\ \alpha=\begin{bmatrix}\alpha & \alpha_2 & \alpha_3 & \alpha_4 & -\alpha_5 & -\alpha_6\end{bmatrix}$$

$$where\ I=\begin{bmatrix}-h_{n_1} & -h_{n_2} & -h_{n_3} & -h_{n_4} & -h_{n_5} & -h_{n_6} & -h_{n_7} & -h_{n_8} & -h_{n_9} & -h_{n_{10}} & -h_{n_{11}} & -h_{n_{12}} & -h_{n_{13}} & -h_{n_{14}}\end{bmatrix}$$

$$and\ \ H^T=\begin{bmatrix} H_1(t+1)-H_1(t)-[\frac{1}{A_1}(Q_i-C_{12}\arctan(H_1-H_2)\sqrt{2g(H_1-H_2)}-C_l\arctan(H_1)\sqrt{2gH_1})] \\ H_2(t+1)-H_2(t)-[\frac{1}{A_2}(C_{12}\arctan(H_1-H_2)\sqrt{2g(H_1-H_2)}-C_0\arctan(H_2)\sqrt{2gH_2}) \\ Q_i(t+1)-Q_i(t)-[b_m\arctan(u)\sqrt{2gu(t)}-a_mQ_i(t)] \\ -\frac{200gC_{dl}}{\sqrt{2gH_1(t)}} \\ 18-u(t) \\ 2H_2(t)-200 \end{bmatrix}^T$$

$$and\ \ I^T=\begin{bmatrix} Q_i(t)-200 \\ -Q_i(t) \\ -200+Q_0(t) \\ -Q_0(t) \\ -H_1(t)+200 \\ H_1(t) \\ 2H_2(t)+e-r-200 \\ -2H_2(t)-e+r \\ -2u(t)+k_pe+k_Ix_3-18+u(t) \\ u(t)-k_pe-k_Ix_3-u(t) \\ a_m-18 \\ -a_m \\ b_m-18 \\ -b_m \end{bmatrix}$$

$$and\ \ A^T=\begin{bmatrix} Q_L \\ -C_{dl}\sqrt{2gH_1} \\ u(t) \\ -k_pe \\ -k_Ix_3 \\ e \\ -r \\ H_2 \\ 854 \end{bmatrix} \tag{69}$$

E. Augmented Lagrangian unconstrained presentation co-states calculation for optimal control:

$$
\begin{aligned}
&L_\mu(k,1) = -\frac{1}{2}H_1(k)^2 - \frac{1}{2}H_2(k)^2 + \frac{1}{2}u(k)^2 - a_1(k)*[H_1(t+1) - H_1(t) - [\frac{1}{A_1}(Q_i - C_{12}\arctan(H_1 - H_2)\sqrt{2g(H_1 - H_2)} - C_l \arctan(H_1)\sqrt{2gH_1})]*T_s]\\
&-r_k*[*[H_1(t+1) - H_1(t) - [\frac{1}{A_1}(Q_i - C_{12}\arctan(H_1 - H_2)\sqrt{2g(H_1 - H_2)} - C_l \arctan(H_1)\sqrt{2gH_1})]*T_s]^2\\
&-a_2(k)*[H_2(t+1) - H_2(t) - [\frac{1}{A_2}(C_{12}\arctan(H_1 - H_2)\sqrt{2g(H_1 - H_2)} - C_0 \arctan(H_2)\sqrt{2gH_2})]*T_s]\\
&-r_k*[H_2(t+1) - H_2(t) - [\frac{1}{A_2}(C_{12}\arctan(H_1 - H_2)\sqrt{2g(H_1 - H_2)} - C_0 \arctan(H_2)\sqrt{2gH_2})]*T_s]^2\\
&-a_3(k)*[Q_i(t+1) - Q_i(t) - [b_m \arctan(u)\sqrt{2gu(t)} - a_m Q_i(t)]*T_s] - r_k*[Q_i(t+1) - Q_i(t) - [b_m \arctan(u)\sqrt{2gu(t)} - a_m Q_i(t)]*T_s]^2\\
&-a_4(k)*[Q_L - C_{dl}\sqrt{2gH_1}] - r_k*[Q_L - C_{dl}\sqrt{2gH_1}]^2\\
&-a_5(k)*[e - r + H_2] - r_k*[e - r + H_2]^2\\
&-a_6(k)*[u - k_p e - k_I x_3] - r_k*[u - k_p e - k_I x_3]^2 - h_{n_1}*[Q_i(k)] - r_k*[Q_i(k)]^2\\
&-h_{n_2}*[200 - Q_i(k)] - r_k*[200 - Q_i(k)]^2\\
&-h_{n_3}*[Q_0(k)] - r_k*[Q_0(k)]^2\\
&-h_{n_4}*[200 - Q_0(k)] - r_k*[200 - Q_0(k)]^2\\
&-h_{n_5}*[H_1(k)] - r_k*[H_1(k)]^2\\
&-h_{n_6}*[200 - H_1(k)] - r_k*[200 - H_1(k)]^2\\
&-h_{n_7}*[H_2(k)] - r_k*[H_2(k)]^2\\
&-h_{n_8}*[200 - H_2(k)] - r_k*[200 - H_2(k)]^2\\
&-h_{n_9}*[u(k)] - r_k*[u(k)]^2\\
&-h_{n_{10}}*[18 - u(k)] - r_k*[18 - u(k)]^2\\
&-h_{n_{11}}*[a_m] - r_k*[a_m]^2\\
&-h_{n_{12}}*[18 - a_m] - r_k*[18 - a_m]^2\\
&-h_{n_{13}}*[b_m] - r_k*[b_m]^2\\
&-h_{n_{14}}*[18 - b_m] - r_k*[18 - b_m]^2
\end{aligned}
$$

The co-states α_k are determined by backward integration of the adjunct state equation yielding:

$$
\begin{bmatrix}\alpha_1\\ \alpha_2\\ \alpha_3\end{bmatrix}_{k-1} = -2h_k\frac{\partial E_d}{\partial x_k} - F_d^T\lambda_k - h_k[\sum_{i=1}^{N}\nabla_{x_k}\Psi_{\mu s}(\sigma_k^j, s_i^{Dl}(x_k))]
$$

$$
-h_k[\sum_{j=1}^{N}\nabla_{x_k}\phi_{\mu g}(\rho_k^j, g_j^D(x_k, \upsilon_k, h_k))]
$$

where

$$x_{k+1} = f_{d_z}^D(x_k, \tau_k, h_k) \ , k = 0, \ldots\ldots, N-1$$

$$g_j^D(x_k, \upsilon_k, h_k) \le 0, \ j \in \{1, 2, \ldots\ldots j\}$$

$$F_d^{\ T} = I_d$$

$$E_d = H_1, H_2 \ and \ u$$

F. Implementation of the Two Tank System using Matlab:

The two tank system implementation has been done on matlab using the differential state equations representing the model of the two tank system (See Fig 8-10).

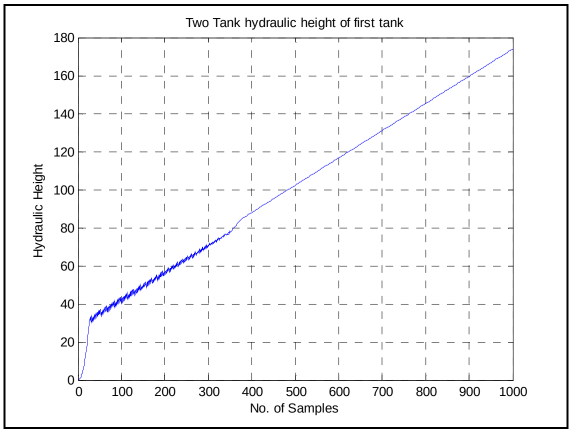


Figure 8. Two tank Implementation for the height of H_1 tank

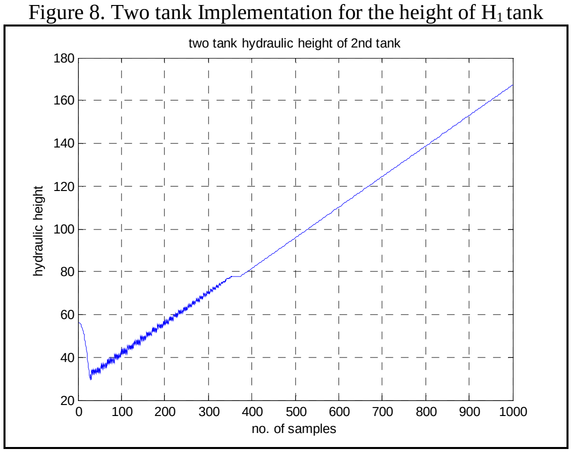


Figure 9. Two tank Implementation for the height of H_2 tank

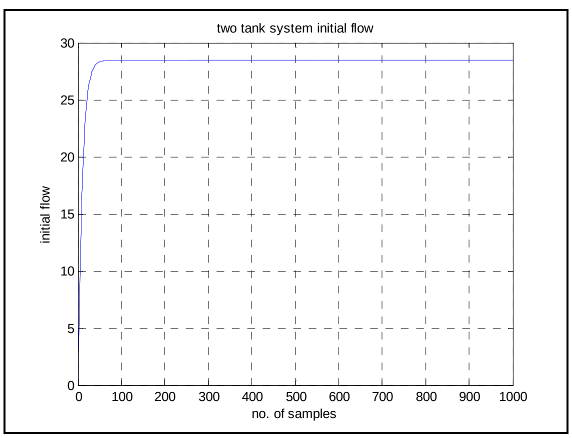


Figure 10. Two tank Implementation for the flow Q_i

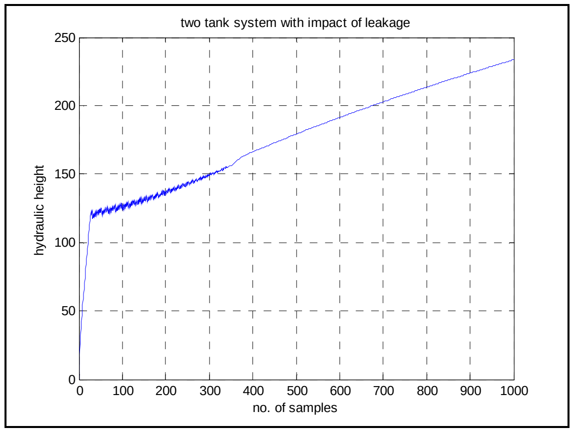
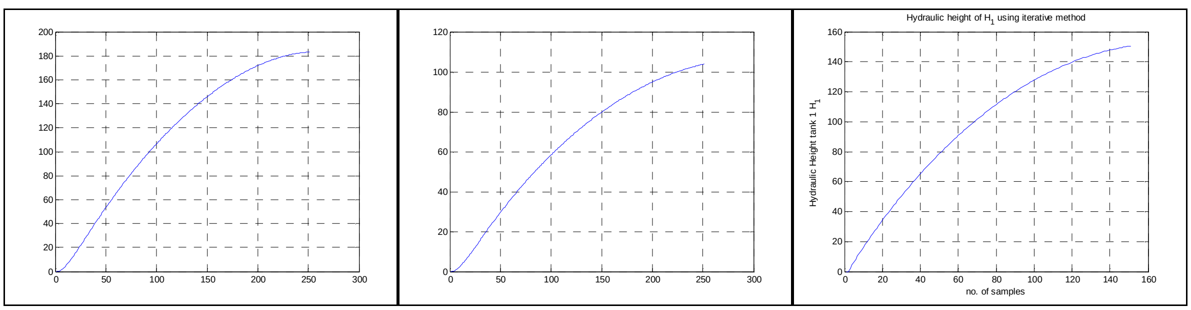


Figure 11. Two tank Implementation for the flow Q_L

G. Implementation of Augmented Lagrangian Iterative algorithm:

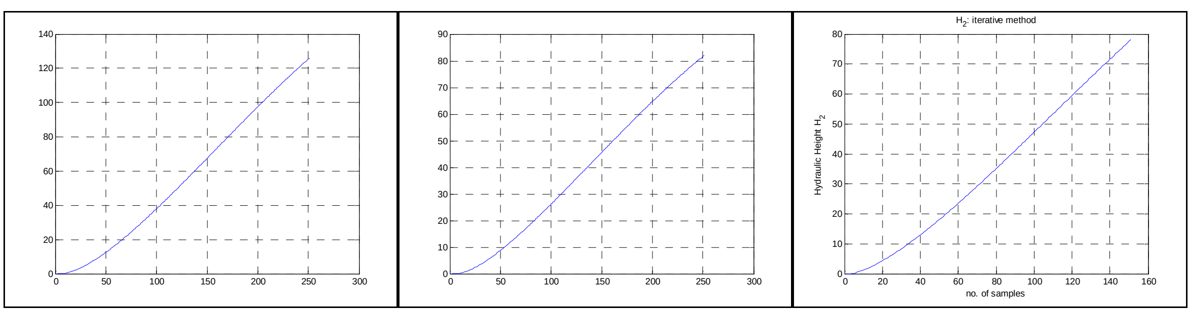
Generating the augmented Lagrangian Iterative Algorithm yields the following states of the system, the lagrangrange multipliers and augmented lagrangians,

The profiles of height of tank 1, tank2 can be seen in figure 12-13.



(a) (b) (c)

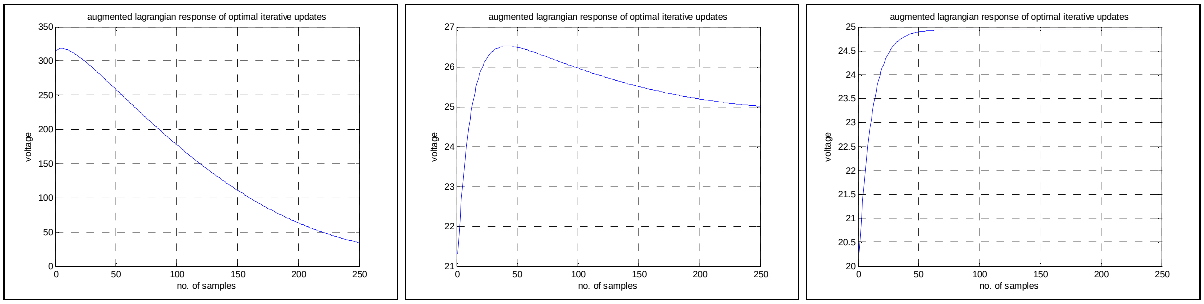
Figure 12. Iterative method Implementation for the height of H_1



(a) (b) (c)

Figure 13. Iterative method Implementation for the height of H_2

The profiles of u can be seen in figure 14(a-c). It can be seen that the iteration process is giving updated results in the control u profile.



(a) (b) (c)

Figure 14 (a-c). Iterative method Implementation for the profile of control u

H. Genetic Algorithm Based Implementation of Augmented Lagrangian

Because of highly diversified system, the standard functional technique of implementing the Genetic Algorithm was not successful, therefore, a generalized genetic algorithm has been designed to capture all the critical sides of implementing the Algorithm. The following are the steps for the Genetic Algorithm.

The following are the steps for the Genetic Algorithm.

H.I Genetic Algorithm Function

The following is the function of the genetic algorithm developed:

function[global best, cost convergence]=genetic(num genes, min gene, max genetic pop size, minimize, elite count, tour size, prob mut, max gen, alpha)

The implementation of the Genetic Function can be elaborated as follows:

[global best, cost convergence]=genetic(a, b, c, d, e, f, g, h, i, j)

where

- *a = number of genes*
- *b = row vector containing minimum values of genes*
- *c = row vector containing maximum values of gene*
- *d = population size*
- *e = '1' for minimize and '0' for maximize*
- *f = number of elite chromosomes (must be an even number)*
- *g = tournament size for parent chromosome selection*
- *h = probability of mutation*
- *i = maximum number of generations*
- *j = alpha for BLX-alpha crossover, typical values are 0.4 to 0.5*

Initializing best solution matrix with zeros:

best = zeros (max gen,num genes+1);

H.II Initial Population Generation

pop = repmat(min gene,pop size,1) + rand(pop size,1)(max gene - min gene);*

H.III Fitness Function Evaluation

```
for gen=1:1:max_gen

[cost]=costfunc(pop);

Now calculating fitness

if minimize == 1
   fitness = 1./cost;
else
   fitness = cost;
end
```

H.IV Selection of elite chromosomes and best solutions

```
pop_sort = [pop fitness];
pop_sort = sortrows (pop_sort,(num_genes+1));

elite = pop_sort(end-elite_count+1:end,1:end-1);
best(gen,:) = pop_sort(end,:);
```

H.V Generation Formation

```
new_gen = zeros(pop_size,num_genes);
new_gen(end-elite_count+1:end,:) = elite;

for g_update=1:1:(pop_size-elite_count)/2
```

H.VI Tournament Selection

```
tour_index = 1+(pop_size-1)*rand(2,tour_size);
tour_index = round(tour_index);

parent_ind = max(tour_index');

preventing asexual reproduction

while parent_ind(1) == parent_ind(2)
   [a,b] = max(tour_index(2,:));
   tour_index(2,b) = 0;
   parent_ind(2) = max(tour_index(2,:));
end

selected parents are given by

parent(1,:) = pop_sort(parent_ind(1),1:end-1);
parent(2,:) = pop_sort(parent_ind(2),1:end-1);
```

H.VII Applying crossover

```
finding crossover point by generating a random integer between "1" and "number-of-genes-minus-one"
```

```
 c = 1+(num_genes-1-1)*rand;
 c = round(c);
child(1,:) = [parent(1,1:c) parent(2,c+1:end)];
child(2,:) = [parent(2,1:c) parent(1,c+1:end)];
```

applying BLX-alpha crossover

```
for blx=1:num_genes
    pblx = [parent(1,blx) parent(2,blx)]';
    pmax = max(pblx); pmin = min(pblx);
    I = pmax - pmin;
    child(1,blx) = (pmin-I*alpha)+(pmax+I*alpha - pmin-I*alpha)*rand;
    child(2,blx) = (pmin-I*alpha)+(pmax+I*alpha - pmin-I*alpha)*rand;

   saturating the value of child gene if child gene exceeds the
   constraint on the gene

    for child_check=1:1:2
        if child(child_check,blx) < min_gene(blx)
            child(child_check,blx) = min_gene(blx);
        end

        if child(child_check,blx) > max_gene(blx)
            child(child_check,blx) = max_gene(blx);
        end
    end

end
```

H.VIII Comparison

Comparing children with parents, and selecting the ones that are more fit to pass on to the next generation

```
comp = [parent; child];
[cost_comp]=costfunc(comp);
```

calcuating fitness

```
if minimize == 1
    fit_comp = 1./cost_comp;
else
    fit_comp = cost;
end

comp = [comp fit_comp];
comp = sortrows(comp,(num_genes+1));
```

selecting the two most fit out of the 4 parents and children

```
new_gen((2*g_update)-1:(2*g_update),:) = comp(end-1:end,1:end-1);
```

F.VII Mutation

```
finding mutation point by generating a random integer between "1" and "number-of-genes" for each child  if d  <=
probability of mutation, performing mutation

for mut = 1:1:2
d(mut) = rand;
if d(mut) <= prob_mut
   mut_point = 1+(num_genes-1)*rand;
   mut_point = round(mut_point);
   new_gen(2*g_update+mut-2,mut_point)=min_gene(mut_point)+rand*(max_gene(mut_point)-min_gene(mut_point));
end
end

end

pop=new_gen;
gen

end
```

H.IX Formulating Global Best and Cost Convergence

```
if minimize == 1
   cost_convergence = 1./(best(:,num_genes+1));
else
   cost_convergence = best(:,num_genes+1);
end

global_best = best(:,1:num_genes);
```

The follwing results are being obtained after the implementation of the Genetic Algorithm:

H.XI Optimization of the hydraulic Height for H_1 Tank

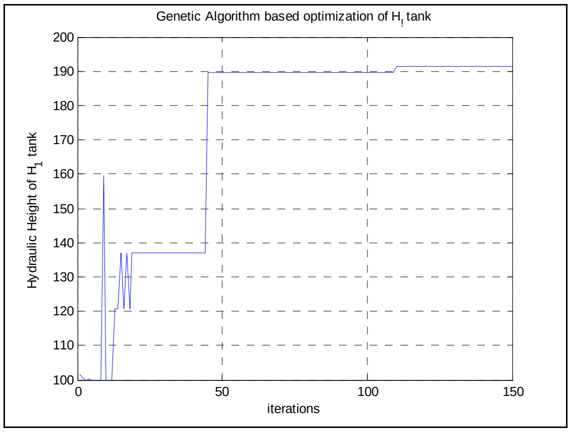


Figure 15. Genetic Algorithm Based Optimization for the height of H_1 tank

H.XII Optimization of the hydraulic Height for H_2 Tank

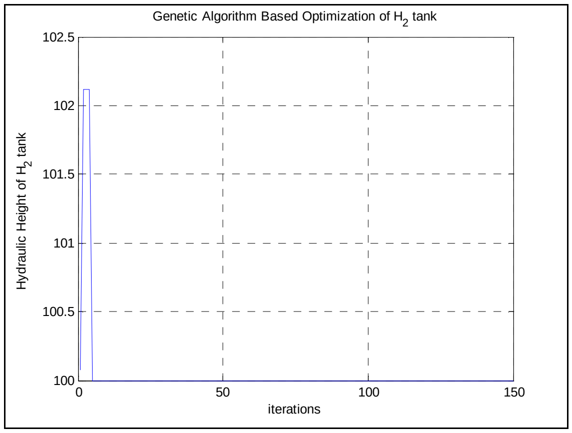


Figure 16. Genetic Algorithm Based Optimization for the height of H_2 tank

H.XIII Optimization of the Controller Voltage u for minimum energy:

Figure 17 shows the result of Genetically optimized augmented Lagrangian based controller voltage u for minimum energy:

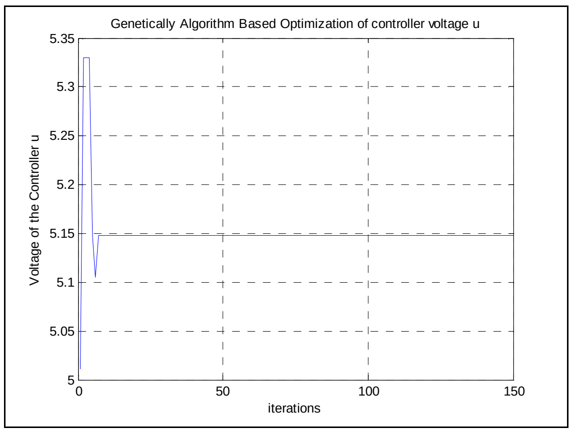


Figure 17 Genetic Algorithm Based Optimization for the controller voltage u

I. Particle Swarm Optimization Based Implementation of Augmented Lagrangian

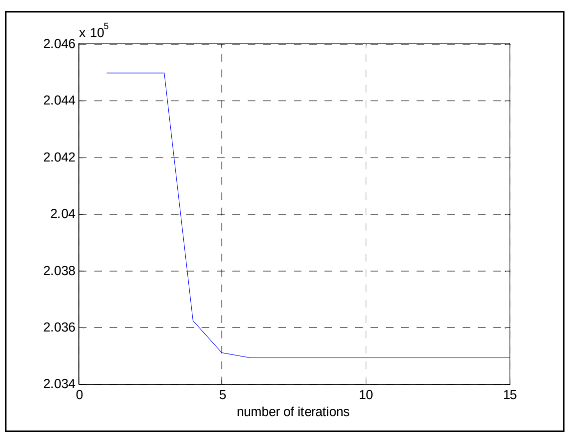


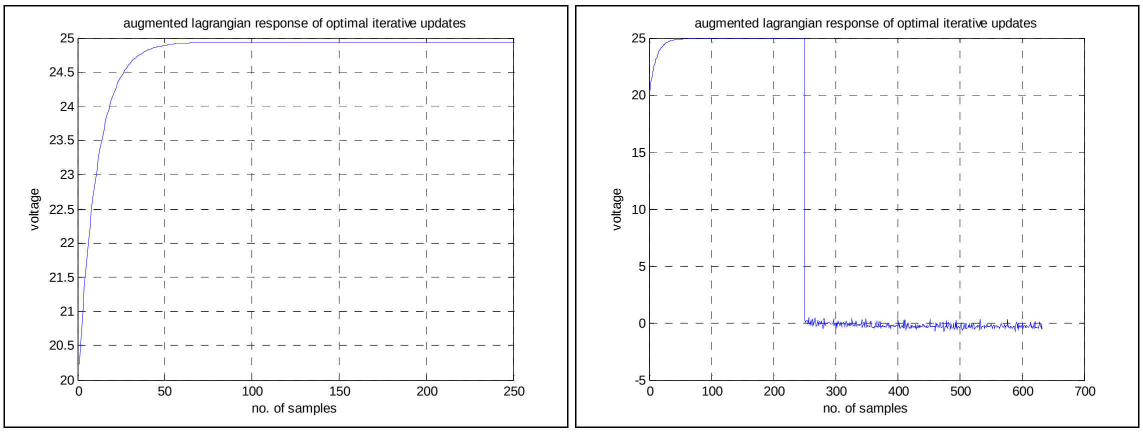
Figure 18. PSO-Based Cost Convergence for controller voltage u

Particle Swarm based Augmented Lagrangian is developed. The developed algorithm is applied on the above defined problem to search for optimal value of control input data clusters. The number of iterations is kept 15, population size is kept

75, cognitive and social parameters c_1 and c_2 are kept equal to 2, and constraints on the radii, as defined above, are observed strictly. The convergence of objective function is shown in figure 18. Cost function convergence to optimal or near optimal solution regardless of initial solution demonstrates the robustness of the algorithm.

VI. Comparison of Results between Iterative Method and Genetic Algorithm

For the gist of optimal control, the result is being compared for the methods of iteration based algorithm and PSO-GA-Based approach as can be seen in this section from Figure 19(a-b). It can be seen that both iterative based algorithm and PSO-GA based algorithm were able to provide a “controlled” u profile in the given number of iterations but the genetic algorithm has an upper edge than the iterative algorithm because it is providing an economical value of u.

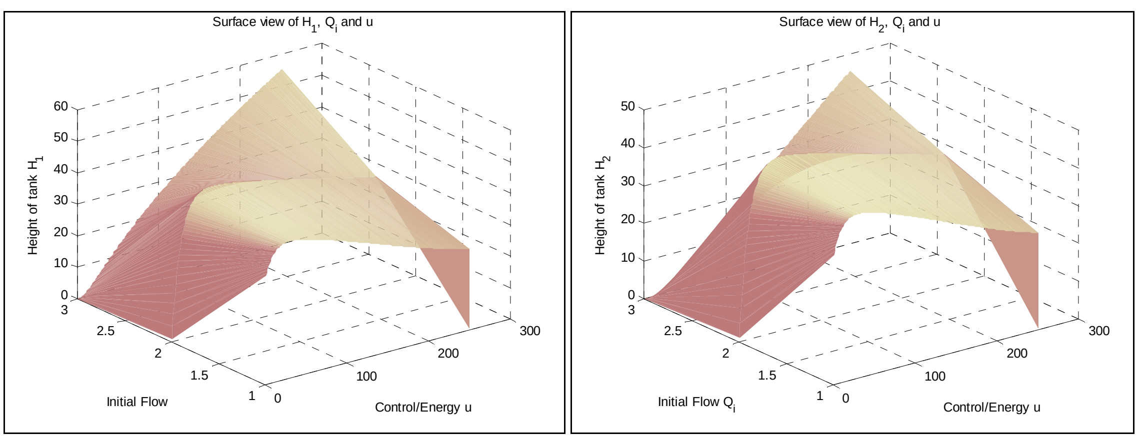


(a) (b)

Figure 19 (a) Iterative algorithm based optimal control and (b) PSO-GA-Based Optimal control

VII. Surface Plots for Analysis

For the analysis of the effect of control/energy u, on the height of first tank, height of second tank, and the initial flow, the result is being shown as can be seen in this section from Figure 20(a-b).



(a) (b)

Figure 20 Surface view for (a) Height of tank 1, initial flow and control/energy and (b) Height of tank 2, initial flow, and control/energy

VIII. Conclusion

In this paper, we presented an optimal control approach to a constrained optimization non-linear problem to the fault diagnosis problem, based on a combination of strategies like augmented lagrangian and PSO-GA-Based approach. This optimal control approach ensures the optimal height of water at minimum energy level. As such, this augmented lagrangian based approach can be made an effective part of an overall approach that tackles both optimal control of the system and optimization of the non-linear constraints. For this technique, PSO-GA-Based approach has been used. The effectiveness of this scheme has been evaluated on a benchmarked laboratory scaled two tank system.

ACKNOWLEDGMENT

The authors wish to acknowledge the support of KFUPM in carrying out this work under the grant of SB100017.

REFERENCES

[1] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. Automatica, 35(3):407 – 427, 1999.
[2] M. P. Avraam, N. Shah, and C. C. Pantelides. Modelling and optimisation of general hybrid systems in the continuous time domain. Computers chem. Engineering, S221–S228, 1998.
[3] V. S. Vassiliadis, R. W. H. Sargent, and C. C. Pantelides. Solution of a class of multistage dynamic optimization problems. 1. Problems without path constraints, 2. Problems with path constraints. Ind. Eng. Chem. Res., 33:2111–2122, 2123–2133, 1994.
[4] A. I. Ruban, Sensitivity coefficients for discontinuous dynamic systems. Computer and Systems Sciences International, 36:536–542, 1997.
[5] Stoer J. and Bulirsch R. (1980): Introduction to Numerical Analysis.—New York: Springer.
[6] Robinson S.M. (1980): Strongly regular generalized equations.— Math. Oper. Res., Vol. 5, No. 1, pp. 43–62.
[7] Alt W. (1990a): Lagrange-Newton method for infinite dimensional optimization problems. — Numer. Funct. Anal. Optim., Vol. 11, No. 3/4, pp. 201–224.
[8] Alt W. (1990b): Parametric programming with applications to optimal control and sequential quadratic programming.—Bayreuther Math. Schriften, Vol. 34, No. 1, pp. 1–37.
[9] Alt W. (1990c): Stability of solutions and the Lagrange-Newton method for nonlinear optimization and optimal control *problems*. — (Habilitationsschrift), Universität Bayreuth, Bayreuth.
[10] Alt W. and Malanowski K. (1993): The Lagrange-Newton method for nonlinear optimal control problems. — Comput. Optim. Appl., Vol. 2, No. 1, pp. 77–100.
[11] K.A. De Jong and W.M. Spears, Using genetic algorithms to solve NO-complete problems, in Proceedings of the Third International Conference on Genetic Algorithms, ed. By J. D. Schaffer. Morgan Kaufmann: 1989.
[12] D.E. Goldberg, K. Zakrzewski, B. Sutton, R. Gadient, C. Chang, P. Gallego, B. Miller and E. Cantu-Paz, Genetic Algorithms: A Bibliography, illegal Report no. 97011, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign, 1997.
[13] Eberhart, R., Shi, Y. (1998). Parameter selection in particle swarm optimisation. In *Evolutionary Programming VII*. pages 591-600.
[14] Khoukhi A et al, Luc Baron, Marek Balazinski, "Constrained multi-objective trajectory planning of parallel kinematic machine", Robotics and Computer Integrated Manufacturing, (2008).
[15] Khoukhi A et al, "Dynamic modeling and optimal time-energy off-line programming for mobile robots, a cybernetic problem. Kybernetes 2002; 31(5–6): 731–5.
[16] Khoukhi A et al, Baron L, Balazinski M., "Constrained multi-objective off-line programming of parallel kinematic machines, Technical Report, E´cole Polytechnique de Montreal, CDT-P2877-05,70Pages; May2007.