Design of a PID control gains for a robotic manipulator under parametric uncertainties by using DE-SQP algorithm

Miguel G. Villarreal-Cervantes

Abstract — As the demand of robotic manipulator with an adequate performance under the effect parametric uncertainties is increased, the fixed PID controller requires an new approach to handle with such uncertainties. In this paper the formal formulation of a robust fixed PID control design is proposed as a constrained dynamic optimization problem. Hybrid algorithm is used to solve the problem. The empirical results show the importance of using heuristic approaches in real world optimization problem.

Keywords: Differential evolution, heuristic algorithms, SQP algorithm, fixed PID control.

1. Introduction

The proportional-integral-derivative (PID) controller is widely used in industrial applications in spite of advanced control techniques [1]. The popularity and widespread use of PID controller is attributed to its structural simplicity, performance characteristics and the application of a broad class of dynamic systems.

Several works deal with the linear fixed PID controller under parametric uncertainties called robust fixed gain PID controller. Those tuning strategies are mainly based on optimization approaches. Newton-Raphson algorithm is proposed in [2] to find the proportional-integral (PI) control gains satisfying \( H_\infty \) specifications. The initial condition of the Newton-Raphson algorithm is a crucial factor to find good solutions. In [3], the \( H_\infty \) static output feedback control technique is proposed to transform a robust PID control method in order to find the optimal PID controller gains in a multi-machine power system. An iterative linear matrix inequalities technique is used to solve the optimization problem. Another approach to deal with the robust PID controller is proposed in [4]. This is based on probabilistic collocation method (PCM) with stochastic parametric uncertainties. Nonlinear programming method is used to compute the optimal PID parameters. In that work the PCM method is compared with the Monte Carlo (MC) Method. The PCM method showed similar performance with less computational effort than the MC method. In [5] presents a robust PID tuning method based on the plant step response experimentally obtained, such that, the transfer function of the plant is not required.

In order to deal with the highly non-convexity \( H_\infty \) problem, the higher order linear systems and with the convergence to suboptimal solution in the robust fixed PID controller problem, meta-heuristic algorithms have been used to solve the problem. In this aspect, augmented Lagrangian particle swarm optimization is proposed in [6]. In [7], a heuristic Kalman algorithm (HKA) is proposed to solve unconstrained optimization problem with penalty function and \( H_\infty \) performance function. Nevertheless, the algorithm used to solve the optimization problem depends on the problem at hand [8]. There are two main approaches to be considered in the selection of the algorithm and they depend on how the search direction is computed: 1) The gradient based algorithms [9], [10] and 2) the meta-heuristic based algorithms [11].

The gradient based algorithms (GBA) such as sequential quadratic programming (SQP) algorithm, depend on the initial condition and the convexity of the optimization problem [12]. The drawbacks in the GBA are the convergence to local solution near the initial one when the optimization problem is multi-modal and it can not be used in discontinuous optimization problem. Hence, gradient based algorithms do not ensure global optimum and they have limited application.

On the other hand, meta-heuristic based algorithms (MHBA) have been widely used to solve real world optimization problems, [13], [14], [15]. Differential evolution (DE) [16] is an MHBA which can be used to solve continuous, discontinuous, linear, nonlinear, dynamic and static optimization problems. DE performs mutation based on the distribution of the solutions in the current population such that, the search direction depends on the location and selection of individuals. The simplicity, easy of implementation, reliability and high performance made the DE algorithm the most used algorithm in real world optimization problem and one of the algorithms used to compare the performance of a proposal algorithm. Nevertheless, some modifications to the original DE schemes have been done to enhance the explorative and exploitative performance. In [17], the modified versions of the DE are classified into two classes: i) DE integrating an extra component and ii) Modified structures of DE. The first class of DE is of the interest in this paper.

There are several DE variants but the most popular is called DE/rand/1/bin where just one difference (of two randomly chosen individuals added to another solution) is calculated. In this paper, the exploitative performance of the DE/rand/1/bin is enhance by adding an extra component
in the DE search. The exploitative pressure is done by combining a SQP approach.

In the works mentioned above, the robust fixed gain PID controller approaches are stated to linear systems. In this paper a new robust fixed gain PID design approach to handle nonlinear systems is proposed. The importance of using an evolutionary and gradient based algorithm is discussed.

The rest of the paper is organized as follows: The robust fixed gain PID design approach is formulated in Section 2. In Section 3 the DE algorithm is shown. The result and discussion are given in Section 4. Finally, in Section 5 the conclusions are drawn.

2. Robust PID control design problem statement

The purpose of the robust PID control design is to obtain the gains where the performance of the system remains inside the design specifications in spite of unknown parameter variations. So, the robust integrated design is formulated as a nonlinear mono-objective dynamic optimization problem where the control parameters \( p \in \mathbb{R}^n \) are optimized by minimizing the robust performance index \( \Psi \in \mathbb{R} \) (1), subject to the dynamic model of the feedback system and the sensitivity vector of the states (2), static and dynamics inequality constraints (3), static and dynamics equality constraints (4) and lower and upper limits in the input vector \( u \) (5). The state vector is represented by \( x \in \mathbb{R}^n \) with \( x(0) = x_0 \) is the initial state vector for the nonlinear differential equation \( \dot{x} = f(x, p, u, t) \) (dynamic model of the system), \( S_\xi = \frac{\partial S}{\partial \xi} \in \mathbb{R}^n \) is the sensitivity vector of the state vector \( x \) with respect to the unknown parameters \( \xi \), \( t \) is the time variable and \( u = f_u(x, p, \xi, t) \in \mathbb{R}^m \) is the input vector of the dynamic system. The unknown parameters \( \xi \in \mathbb{R} \) can vary from their nominal one \( \xi = \xi_0 + \Delta \xi \), \( g(x, p, t) \) and \( h(x, p, t) \) are static/dynamic inequality or equality constraint vectors, respectively. The function \( L \) is a nonlinear one of class \( C^1 \). In this paper the function \( L \) is called goal function and the term \( \frac{\partial L}{\partial \xi} \) i.e., the sensitivity of the goal function to the unknown parameters \( \xi \in \mathbb{R} \), is called sensitivity of the goal function.

\[
\begin{align*}
\text{Min}_p \quad & \Psi = \text{Min}_p \quad \frac{\partial J^2}{\partial \xi} \\
\bar{J} = & \int_0^{t_f} L^2(x, p, \xi, u, t) dt \\
\text{subject to:} \\
1.- & \quad \text{Dynamic model of the system and the sensitivity vector of the state, with } x(t_0) = x_0, \quad S_\xi(t_0) = 0 \\
2.- & \quad \text{Static and dynamic inequality constraint vector:} \\
3.- & \quad \text{Static and dynamic equality constraint vector:} \\
4.- & \quad \text{Lower and upper limits in the input vector } u:
\end{align*}
\]

It is important to remark that the minimization of the robust performance function \( \Psi = 4L^2 \left( \frac{\partial L}{\partial \xi} \right)^2 \) implies the minimization of both terms, the goal function \( L \) and the sensitivity of the goal function \( \frac{\partial L}{\partial \xi} \).

In the next subsections, the robotic system, the design variables, the robust performance function and constraints for the robust PID control design of a particular problem are detailed.

2.1 Robotic system

The robust PID control design approach is applied to the 3R manipulator with a parallelogram five-bar mechanism. It is considered the mass of the end-effector (mass of the fifth link \( m_5 \) in Fig. 1) as the unknown parameters, i.e., \( \xi = m_5 \). We assume that a payload can be added to the tip of the end-effector, changing the mass of the fifth link.

The 3R manipulator with a parallelogram five-bar mechanism presents three degree of freedom in the joint space which provide the ability to move the tip of the end-effector in the plane \( X - Z \) with an orientation \( \phi \) with respect to the axis \( X \) of the inertial coordinate system \( X - Z \). The parallelogram five-bar mechanism, included into the 3R robot, achieves a higher precision and a higher stiffness than a 3R robot without the parallelogram five-bar mechanism [19]. The 3R manipulator is shown in Fig. ??, where \( q_1, q_1, q_4, l_l, l_e, I_i \in i = 1, 2, ..., 4 \) are the joint position, joint velocity, joint acceleration, mass, length, mass center length, inertia of the \( i \)-th link length, \( (\dot{x}, \dot{z}) \) and \( \phi \) are the Cartesian coordinate and the angular position of the manipulator’s end-effector, respectively.

Let \( x = [x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9]^T = [q_1, q_2, q_3, q_4, q_1, \dot{q}_2, \dot{q}_3, \int_0^r e_1(\tau) d\tau, \int_0^r e_2(\tau) d\tau, \int_0^r e_3(\tau) d\tau]^T \in \mathbb{R}^9 \)
In addition, the trajectory tracking must be as insensitive as possible to variations at the end-effector’s payload. As the end-effector trajectory can be map to the joint space of the 3R manipulator by using the inverse kinematics [20], and the inverse kinematics does not depend of the control design variables (PID gains), then the joint position tracking error is chosen as the goal function (i.e., $L_{m}$). Considering the mass of the end-effector as the unknown parameters $\xi = m_5$ (mass of the fifth link $m_5$ in Fig. xxx), the robust performance function $\Psi$ is showed in (12). The unknown parameters can vary from the nominal one accordingly to the mass of the end-effector without payload and its variations $\Delta \xi$ is the mass of the added payload.

$$\Psi = \sum_{i=1}^{3} \frac{\partial J_i}{\partial \xi}^2 \bigg|_{\xi=\xi}$$ (12)

When the robust performance function (12) is optimized, the variations of the joint position error due to the changes at the unknown parameters $\xi = m_5$, as well as the joint position error $e_i = \bar{x}_i - x_i$, are simultaneously optimized within the same performance function, as it is observed in (6).

The desired end-effector position $(\hat{x}, \hat{z})$ and its orientation $\hat{\phi}$ are described by the hypocycloid trajectory and sinusoidal trajectory, given by (13)-(15), where $t$ is the time variable.

$$(\hat{x}, \hat{z}) = (0.2 + 0.08181 \cos(1.2566t) + 0.01818 \cos(5.6548t), 0.1 + 0.08181 \sin(1.2566t) - 0.01818 \sin(5.6548t), 0.4363 \sin(2.0943t))$$ (13-15)

In order to compute the desired state vector $\bar{x}_i$ from the end-effector desired trajectory (13)-(15), the inverse kinematic of the 3R manipulator must be obtained. The inverse kinematics is show in (16)-(18).

$$\bar{x}_1 = A \tan 2 \left( \frac{\hat{z}}{\hat{x}_m} \right) - \gamma$$ (16)

$$\bar{x}_2 = \bar{x}_1 + \bar{x}_1' + \pi$$ (17)

$$\bar{x}_3 = \hat{\phi} - \bar{x}_2 + \pi$$ (18)

where

$$\gamma = A \tan 2 \left( \frac{l_4 \sin(q_2')}{l_1 + l_4 \cos(q_2')} \right)$$
\[ q_2' = A \tan 2 \left( -\frac{\sqrt{u_1^2 t_1^2 - (t_m^2 + t_m^2 - t_2^2)^2}}{t_m^2 + t_m^2 - t_2^2} \right) \]

### 2.4 Constraints

The dynamic behavior of the 3R manipulator with a parallelogram five-bar mechanism with its PID control system is included as a dynamic constraint. Those are stated in (9) and (10).

In real application, the input continuous torque applied to the manipulator joints has physical limits. If such limits are not considered, the PID control system may become unstable in real application which often incur a physical damage of the manipulator. Hence, the bounds of the input torque vector \( u \), is considered as a dynamic constraint (19), where \( u_{\text{BOUND}} = 3N\text{m} \) is the input toque bound.

\[ g_i(t) : |u_i(t)| - u_{\text{BOUND}} \leq 0 \quad \forall i = 1, 2, 3 \quad (19) \]

Other important constraint is the maximum response of the actuators at each sampling time. This constraint is stated in (20), (21), \( \forall i = 1, 2, 3 \). Without including that constraint, the robotic manipulator can reach singular configurations. Singular configurations can provide uncontrollable movements. Then, if that constraint is not fulfilled, the dynamic behavior of the closed-loop system must be stopped, i.e., the performance function is not evaluated and it is assigned a maximum value.

\[ g_{i+3}(t) : |q_i(t + \Delta t) - q_i(t)| - 0.05 < 0 \quad (20) \]
\[ g_{i+6}(t) : |\dot{q}_i(t + \Delta t) - \dot{q}_i(t)| - 11.98 < 0 \quad (21) \]

### 2.5 Optimization problem formulation

The optimization problem for the robust PID control design of the 3R manipulator consists on finding the optimal design parameter vector \( p^* \) (11) such that both the sensitivity of the joint position error and the joint position error (12) are simultaneously minimized subject to dynamic constraint (23) of the nonlinear differential equation that describes the dynamic behavior of the system (8) with the control signal \( u \) (10), the dynamic inequality constraint vector \( g(t) \in \mathbb{R}^p \) (24) that includes the bounds in the control signal (19) and the bounds in the maximum response of the actuators at each sampling time (20)-(21).

\[ \min_{p \in \mathbb{R}^p} \Psi \quad (22) \]

Subject to:

\[ \dot{x} = f(x, p, \xi, u) \quad (23) \]
\[ g(x, p, t) < 0 \quad (24) \]

### 3. Differential evolution

The DE algorithm is composed by four stages (see Fig. 2): Initialization, mutation, crossover and selection [16]. The main differences among the DE variants are in the mutation and crossover states (rows 9-17 in Fig. 2). The general convention used to name the DE variant is DE/x/y/z. DE means Differential Evolution, \( x \) represents the way that the vector is perturbed, \( y \) is the number of difference vectors considered for perturbation of \( x \), and \( z \) stands for the type of crossover being used (exponential or binomial). The most common DE variants are DE/rand/1/bin, DE/rand/1/exp, DE/best/1/bin, DE/rand/2/exp, DE/current-to-rand/1, DE/current-to-best/1, DE/current-to-rand/1/bin and DE/rand/2/dir. In this paper DE/rand/1/bin is used.

In the initialization stage, the initial population design vector \( x^i_{j,G=0} \) is randomly generated for all population, where \( G = 0 \) indicates the initial generation which the vector belongs, \( i \in [1, ..., NP] \) is a population index, \( j \in [1, ..., D] \) is the design parameter index. This stage is shown in (25) where \( \text{rand}_j(0, 1) \in [0, 1] \) is a uniformly distributed random number generator and \( b_{j,U}, b_{j,L} \) indicate the upper and lower limit for the \( j \)-th design parameter. For each initial design vector, its objective function is computed.

\[ \text{For} \ i = 1 \text{ to } NP \text{ Do} \]
\[ \text{For} \ j = 1 \text{ to } D \text{ Do} \]
\[ x^i_{j,G=0} = \text{rand}_j(0, 1)(b_{j,U} - b_{j,L}) + b_{j,L} \quad (25) \]
\[ \text{End For} \]
\[ \text{Evaluate } J(x^i_{G=0}) \]
\[ \text{End For} \]

Once the initial population vector is initialized, the mutation stage creates a mutant vector \( \vec{v}_{G} \) by the mutation and recombination of the individuals of the population. In the crossover stage, trial vector \( \vec{u}_{G} \) is generated when the target vector \( x^i_{j,G} \) is crossed with the mutant vector \( \vec{v}_{G} \). The mutation and the crossover stages depend on the DE variant. The mutation and crossover variants of the DE/rand/1/bin is shown as in 26.
\[ u_j^i = \begin{cases} \text{if } \text{rand}_j(0,1) < CR \text{ or } j = j_{\text{rand}} & \text{then } v_j^i = \vec{x}_j^G + F(\vec{x}_i^G - \vec{x}_j^G) \\ \text{otherwise} & \text{then } v_j^i = \vec{x}_j^G + F(\vec{x}_i^G - \vec{x}_j^G) \end{cases} \] 

(26)

The scale factor \( F \in (0,1) \) in the mutation process, is a positive real number that controls the influence of the selected individuals in order to generate the mutant vector. The indexes \( r_1, r_2 \) and \( r_3 \) are randomly chosen from the range \([1, NP]\). Those indexes can not have the same value.

The crossover probability \( CR \in [0,1] \) controls the influence of the parent vector in the generation of the offspring (higher values mean less influence of the parent vector, hence higher influence of the mutant vector).

Finally, the last stage involves a selection process between the trial vector \( \vec{u}_j^G \) and the target vector \( \vec{x}_j^G \). First, the performance function is evaluated with the trial vector and then an elitism procedure is done where the best of them pass to the next generation. In (27), the selection stage is shown.

\[
\begin{align*}
\text{Evaluate } & J(\vec{u}_j^G) \\
\text{If } & (J(\vec{u}_j^G) < J(\vec{x}_j^G)) \text{ Then} \\
& \vec{x}_{G+1}^i = \vec{u}_j^G \\
& J(\vec{x}_{G+1}^i) = J(\vec{u}_j^G) \\
\text{Else} & \\
& \vec{x}_{G+1}^i = \vec{x}_j^G \\
& J(\vec{x}_{G+1}^i) = J(\vec{x}_j^G) \\
\text{End if}
\end{align*}
\] 

(27)

4. Results and discussion

In this work, the DE/rand/1/bin is programmed in Matlab Release 7.9 on a Windows platform. Computational experiments were performed on a PC with a 2.8 GHz Core i7 Duo processor and 16 GB of RAM. Five independent runs are carried out. The parameters of the DE algorithm are proposed as follows: the population size \( NP \) consists of 100 individuals. The scaling factor \( F \) and the crossover constant \( CR \) are randomly generated in the interval \( 0.3 \leq F \leq 0.9 \) at each generation, and in the interval \( 0.8 \leq CR < 1 \) at each optimization process. The stop criterion is when the number of generations is fulfilled \( G_{\text{Max}} = 400 \).

4.1 Performance of the DE and SQP algorithm

In Table 1 the performance for five independent runs of the DE algorithm are shown. The term \( \text{mean}(J) \) and \( \sigma(J) \) are the mean and the standard deviation of the performance function of the population in the 400th generation. The best performance function value found in the last generation is placed in the column \( J^{*\text{DE}} \). \( J_{\text{Eval}} \) and \( J_{\text{NotEval}} \) are the number of times that the performance function is evaluated or not evaluated, respectively. The number of times that the performance function is not evaluated is when the constraint (21) is not fulfilled. The third column \( \text{Time}[hr] \) is the convergence time.

It is observed in Table 1 that all runs converge to a similar performance function (see column \( J^{*\text{DE}} \)) and all population in the last generation converge to a similar value, as indicate in the column \( \text{mean}(J) \) and \( \sigma(J) \). This indicates that the explorative performance of the DE/rand/1/bin algorithm makes the convergence of the algorithm is towards the same performance function value. On the other hand, there are several times that the performance function is not evaluated (see column \( J_{\text{NotEval}} \)). This reduces the convergence time to get the solution, hence the importance of including the constraint (21) in the optimization problem.

In order to enhance the explorative of the individuals in the DE algorithm, the best individual in different generations is used as the initial condition of the SQP algorithm. The stop criterion is when the number of iteration \( ITR \) is 100 is fulfilled. In Table 2 the performance of the SQP algorithm is shown. The column \( J_{IC} \) indicate the performance function value with the initial condition and the column \( J^{*\text{SQP}} \) is the performance function value when the number of iteration is satisfied. It is observed that the performance function value converges to suboptimal solutions near the initial ones, i.e., the SQP algorithm is highly sensitivity to the initial condition. This indicates that the optimization problem is a multimodal one. As expected, the convergence time of the SQP algorithm is less than the ones of the differential evolution. The SQP algorithm is also test with random initial condition but it does not converge to a solution (it is not an easy task to find good initial condition). For that reason the results with random initial condition are not shown.

4.2 Optimal design

The design performance with the design parameters resulting from the SQP algorithm is compared with the design performance with the design parameters given by a trial & error procedure. In Table 3 the design parameters obtained by the SQP algorithm and by the trial & error procedure are shown.

In Fig. 3 and Fig. 4, the trajectory tracking with the robust fixed PID control design and the trial & error PID design are shown. With no load, both designs track the desired trajectory. Nevertheless, with load, the robust fixed PID control design tracks the desired trajectory, while the trial & error PID design a trajectory tracking error is observed.
Table 1: Experimental results of the DE/rand/1/bin algorithm.

<table>
<thead>
<tr>
<th>Run</th>
<th>mean(J)</th>
<th>σ(J)</th>
<th>J*DE</th>
<th>J_Eval</th>
<th>J_NotEval</th>
<th>Time[hr]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5.46e−13</td>
<td>3.76e−16</td>
<td>5.45e−13</td>
<td>20044</td>
<td>20056</td>
<td>35.63</td>
</tr>
<tr>
<td>2</td>
<td>5.45e−13</td>
<td>2.92e−16</td>
<td>5.44e−13</td>
<td>27481</td>
<td>12619</td>
<td>48.61</td>
</tr>
<tr>
<td>3</td>
<td>5.46e−13</td>
<td>3.95e−16</td>
<td>5.45e−13</td>
<td>20268</td>
<td>19832</td>
<td>35.18</td>
</tr>
<tr>
<td>4</td>
<td>5.45e−13</td>
<td>4.49e−16</td>
<td>5.44e−13</td>
<td>23750</td>
<td>16350</td>
<td>40.13</td>
</tr>
<tr>
<td>5</td>
<td>5.45e−13</td>
<td>2.72e−16</td>
<td>5.44e−13</td>
<td>29845</td>
<td>10255</td>
<td>51.97</td>
</tr>
</tbody>
</table>

Table 2: SQP algorithm with different initial condition given by different generations in the DE algorithm.

<table>
<thead>
<tr>
<th>Generation</th>
<th>J_{IC}</th>
<th>J^{SQP}</th>
<th>Time[hr]</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>28.52e−13</td>
<td>10.8e−13</td>
<td>2.96</td>
</tr>
<tr>
<td>200</td>
<td>11.07e−13</td>
<td>8.39e−13</td>
<td>3.11</td>
</tr>
<tr>
<td>300</td>
<td>5.50e−13</td>
<td>5.50e−13</td>
<td>12.10</td>
</tr>
<tr>
<td>400</td>
<td>5.47e−13</td>
<td>5.44e−13</td>
<td>11.98</td>
</tr>
</tbody>
</table>

Fig. 3: Trajectory tracking of the end-effector of the robotic manipulator without load.

Fig. 4: Trajectory tracking of the end-effector of the robotic manipulator with load of 0.1247 Kg.

Fig. 5: Control signal for the trajectory tracking without load.

satisfying the input torque bounds. Hence, the proposed robust fixed PID control design is validated.

5. Conclusion

In this paper the formulation of the robust fixed PID design for a robotic manipulator is proposed as an optimization problem. The optimization problem is solved by combining the DE and SQP algorithm. The empirical results show that the combination of an evolutionary algorithm with a gradient one make a good exploration and exploitation
of the individuals in the search space. The evolutionary algorithm is highly useful when the SQP algorithm presents high sensitivity to the initial condition such that the searching of the initial condition is not an easy task.

From an engineering design point of view, a heuristic approach must be considered first and then a gradient approach must be considered first and then a gradient approach in order to finely search in the complete space and hence finding the best design.

The trajectory tracking performance of robust fixed PID control design is better than the one of trial & error PID design when the robotic manipulator handle different loads satisfying the input torque bounds. Hence, the proposed robust fixed PID control design is numerically validated.

Acknowledgment

The author acknowledges support from COFAA and SIP of the Instituto Politécnico Nacional through project no. SIP-20140926, and the support from the Consejo Nacional de Ciencia y Tecnología (CONACyT) through project no. 182298.

References