

An optimization algorithm for a better PID controller

- Luu Thanh Tung
- Le Nhan

University of Technology, VNU-HCM

(Manuscript Received on July 28th, 2014; Manuscript Revised October 7th, 2014)

ABSTRACT

A controller of a manipulator has studied and discussed for many years. However, many problems in controlling the precise position of the end effector are still continuing to be studied. To solve the precision of the Robot, two problems are attended. The first thing is to find the accuracy model of dynamics. The second thing is a controller for control law. However, it is so difficult to find an accurate model or differential equations of

motion which is similar to the true manipulator. In addition, some unknown influences on the manipulator will make the accurate differential equations unworthy. Thus, a control algorithm will be introduced with PID controller which coefficients K_p , K_d , K_i are compensated by compensator found from optimization algorithm. With the new algorithm, the results have proved the stability and precision are better.

Key words: Manipulator, PID, Optimization.

1.INTRODUCTION

In the recent years, the control of manipulators with high stability and precision is required highly. The reason for this requirement is that the industrial market needs higher precision products. To control accurately the manipulator, an exact dynamic equation must be shown. However, due to factors such as friction, saturation of actuators, and highly varying loads, the manipulator model becomes uncertain. Many approaches have proposed to solve the problems of control the manipulator with the uncertain nature of tracking. In [1], adaptive controllers assume that the unknown nonlinear dynamics of the system is linearly parameterized. Therefore, there are some errors in the controller that makes the tracking

little inaccuracy. To solve this problem, some approaches [2], [3] have ideas to linearize in certain intervals. In each interval, they will use technics such as neural network, Fuzzy, etc. to choose the K_p , K_i , K_d so that the Robot arm will track the reference line accurately. In [4], the authors use the robust controller to control the Robot arms. In this research, a compensator is used to compensate the lag into the robust controller. As a result, the tracking errors decrease when it is compared with (to?) the traditional robust controller. The approaches above have same the purpose that they change the controller to adapt the change of external environment. In these cases, the Robot models are not accurate. In [5], the exact robot model is found by Genetic Algorithm (GA). It is shown that GAs are able to

find robot parameters effectively even if the robot has low resolution position encoders. With approaches showed, they will go 2 ways to obtain the good tracking. In the first way, parameters consisting of mass, centroid, mass moment of inertia, etc. will be determined. Based on the parameters, the controller will be designed. This way will be only good for simulation because some factors pre-unknown can appear in reality. In the second way, the controller will be designed to track the trajectory well. It does not mean a robust controller. That is the controller will change due to external factors. The changes will be controlled by some algorithms such as fuzzy, neuron.

In this paper, we will introduce a method to control a manipulator with 6 degrees of freedom. This is an old manipulator and restored for gluing the shoe sole. Thus, the robot parameters are not pre-known exactly. This method still uses a

traditional PID controller, and then we use the optimization algorithm to obtain a better controller to track the trajectory. The paper will be divided into 6 sections. The next section will introduce how to find the state equation of the manipulator. The optimization algorithm is then shown and after that the experiments with real manipulator are done. The last section is discussions and conclusions of this study.

2.PARAMETERS OF THE MANIPULATOR

In a general case, the Lagrange dynamic equation of a manipulator can be written as follows.

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau + d' \quad (1)$$

where q is $n \times 1$ vector of joint position, \dot{q} is $n \times 1$ vector of joint velocity, $D(q)$ is inertia matrix.

In our experiment, the manipulator “MOTOMAN-SV3X” is used (Fig. 1).

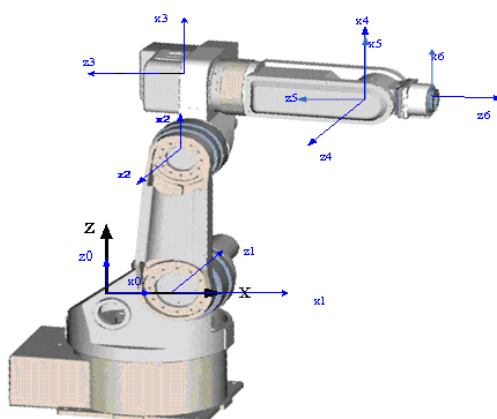
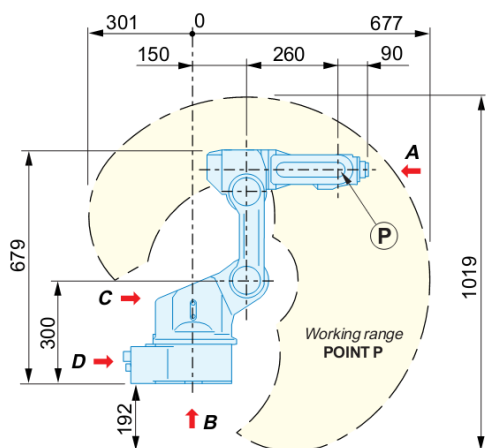


Figure 1. Robot MOTOMAN – SV3X

Because the manipulator is a secondhand one, the control driver and the motor have been changed. From [6,7], the dynamics of the controlled motors via gears is shown in Eq. (2).

$$Ri + L \frac{di}{dt} + k_b \dot{q}_m = u \quad (2)$$

Where i are vectors of armature current, u are vectors of applied armature voltage. R , L and k_b are matrices of armature assistance, armature

inductance and counter-electromotive force (EMF) constant of motor, respectively. The motor torque at each motor shaft is given by:

$$\tau_m = K_b I \quad (3)$$

There is set of gear in each joint and the relation between motor shaft position and joint position is as follows: $q_m = Nq$ (4)

where N is the matrix of gear ratio. Similarly, the relation between the motor torque and the applied torque at the joint is given by:

$$\tau = N\tau_m \quad (5)$$

From Eq. (1) to (4), we can obtain a new equation:

$$R_n \tau + L_n \dot{\tau} + k_{bn} \dot{q} = u \quad (6)$$

where $R_n = R(NK_b)^{-1}$, $L_n = L(NK_b)^{-1}$ and $k_{bn} = K_b N$. Then, insert Eq. (6) into Eq. (1), the combination becomes:

$$M(q)\ddot{q} + D(q, \dot{q}, \ddot{q}) = u + R_n d' + L_n d' \quad (7)$$

where $M(q) = L_n M'(q)$, $D(q, \dot{q}, \ddot{q}) = (R_n M'(q) + L_n (M'(q) + C(q, \dot{q}))) \ddot{q} + (R_n C(q, \dot{q}) + L_n \dot{C}(q, \dot{q}) + k_{bn}) \dot{q} + R_n G(q) + L_n \dot{G}(q)$.

3.CONTROLLING ROBOT ARM

The mission is to control a manipulator to follow a tracking pre-defined. The tracking curve is closed one. The curve is divided into several small parts with starting point and ending point. The control algorithm will control the robot arm from starting point to the ending point of each part and then the robot arm continue to move to the next part with a new starting point and ending point. With each part, we will have tracking error \tilde{q} which is defined as follows:

$$\tilde{q} = q - q_d \quad (8)$$

where q_d is position of ending point of each part in generalized coordinate. From Eq. (7) and (8), the state vector can be written as Eq. (9).

$$e = [\tilde{q}, \dot{\tilde{q}}, \ddot{\tilde{q}}] \quad (9)$$

The Eq. (7) can be rewritten as follows:

$$\begin{bmatrix} \tilde{q} \\ \dot{\tilde{q}} \\ \ddot{\tilde{q}} \end{bmatrix} = \begin{bmatrix} \ddot{\tilde{q}} \\ \dot{\tilde{q}} \\ -M^{-1}(q)D(q, \dot{q}, \ddot{q}) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ M^{-1}(q) \end{bmatrix} u + \begin{bmatrix} 0 \\ 0 \\ M^{-1}(q)(R_n d' + L_n d') - q_n^{(3)} \end{bmatrix} \quad (10)$$

This is an equation of linear feedback with parameters known. However, the function $M()$ and $D()$ are estimated. Assume that $M()$ and $D()$ are replaced by $M_0()$ and $D_0()$, respectively. Following the study in [6], the control law u will be given:

$$u = D_0 + M_0 q_d^{(3)} + M_0 (-k_1 \tilde{q} - k_2 \dot{\tilde{q}} - k_3 \ddot{\tilde{q}} + u_0) \quad (11)$$

where $D_0 + M_0 q_d^{(3)}$ is linear terms. Terms k_1 , k_2 and k_3 are diagonal matrix that be designed to adapt with differences between true and nominal system. u_0 is an auxiliary control signal to reduce the effect of the differences.

The control law in Eq. (11) can be re-written by assigning:

$$v = -k_1 \tilde{q} - k_2 \dot{\tilde{q}} - k_3 \ddot{\tilde{q}} + u_0 \quad (12)$$

And then, the control law is given:

$$u = D_0 + M_0 v \quad (13)$$

Substituting Eq. (11) into Eq. (7), we obtain the equation:

$$\begin{bmatrix} \ddot{q} \\ \dot{q} \\ q \end{bmatrix}' = \begin{bmatrix} 0 & I & 0 \\ 0 & 0 & I \\ -k_1 & -k_2 & -k_3 \end{bmatrix} \begin{bmatrix} \ddot{q} \\ \dot{q} \\ q \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ I \end{bmatrix} u_0 + \begin{bmatrix} 0 \\ 0 \\ I \end{bmatrix} (f + d) \quad (14)$$

where $d = M^{-1}(q)(R_n d')$ and $f = M^{-1}[(D_0 - D) + (M_0 - M)v]$.

Making it simpler, Eq. (14) can be re-written:

$$\dot{e} = Ae + Bu_0 + Bf(x) + B(d(t)) \quad (15)$$

where

$$A = \begin{bmatrix} 0 & I & 0 \\ 0 & 0 & I \\ -k_1 & -k_2 & -k_3 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 0 \\ 0 \\ I \end{bmatrix} \quad (16)$$

From Eq. (15), we must design u_0 so that the effect of unknown plant f and disturbance d is minimized. In this case, we will choose u_0 as follows

$$u_0 = u_e + u_f \quad (17)$$

where u_e is output of the simulating controller that parameters are from Motoman Robot manufacturer. u_f is a compensator which is found by optimization algorithm.

4.OPTIMIZATION ALGORITHM

In this section, algorithm will find an optimal u_f . With an optimal u_f , effects of f and d to the system are minimized. To find an optimal u_f , we will measure the output of the simulating manipulator and then that of real one. The difference between the two outputs will be used to train u_f by optimization algorithm.

In a typical optimization problem, procedures include 3 parts [8] as follows:

The part one is to define an optimization or mathematical programming problem. It can be expressed as follows.

$$\text{Find } X = \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Bmatrix} \text{ which minimizes the function } y = F(X) \quad (18)$$

where X is n dimensional vector called the design vector. $F(X)$ is the objective function

The part one will help the designer understand which the objective is prioritized. Sometime, they have more than one objective function. In this case, to construct an overall objective function as a linear combination of the conflicting multiple objective functions.

Part 2 concerns the design of constraint. Here, restrictions must be satisfied to produce an acceptable design. In illustration, the constraints are expressed as inequalities or equations as follows.

$$g_j(X) \leq 0 \text{ with } j = 1, 2, \dots, m \text{ and } h_k(X) = 0 \text{ with } k = 1, 2, \dots, l \quad (19)$$

Part 3 is to solve the objective function $F(X)$ to find the design values X make $F(X)$ optimal with the constraints given.

In the optimization problem to find the optimal u_f , we assigned u_f in position of X , the difference of two output called e_f was in position of $F(X)$. The constraints in this problem are unnecessary. To minimize the effect of f and d , we must minimize function e_f . The algorithm to do this work is shown in Fig. 2.

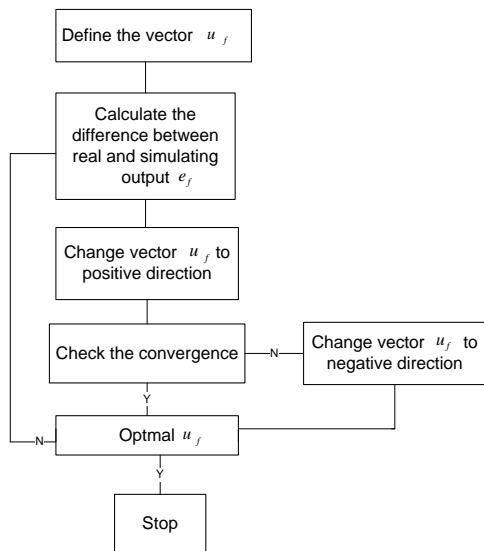


Figure 2. Flowchart of finding the u_f by using optimization algorithm

The first work of the algorithm is to assign u_f a certain value. In our experiment, u_f is assigned the value of u_e , where u_e is output of simulating controller with PID. Next, we run the robot and record the output from encoder, then calculate e_f . If $e_f \gg 0$, we change vector u_f with two ways positive and negative direction. With each direction, we make the experiment and simulation and then make the difference e_f . Which u_f makes e_f smaller will be chosen for next step. The loop will stop when $e_f \approx 0$. When the loop stops, the optimal u_f is achieved.

5. EXPERIMENTAL RESULTS

As described in section 4, experiment includes test on the real manipulator and simulation with parameters of the real manipulator too. The real manipulator is shown in Fig. 3.



Figure 3. Real Robot and tracking line

Firstly, we used PID controller and then control the real manipulator with a distance 20mm. The result of the tracking is the solid line in Fig. 4. The dashed line in Fig. 4 is the result of simulation (Fig. 5) with same PID controller. The maximum difference between two outputs is assigned e_f .

Because $e_f \gg 0$, the loop will start. This time, the controller of the real manipulator is assigned PID and a compensator u_f as Eq. (17) and the controller of simulating manipulating is not changed. In this case, u_f is found by the optimization algorithm in section 4.

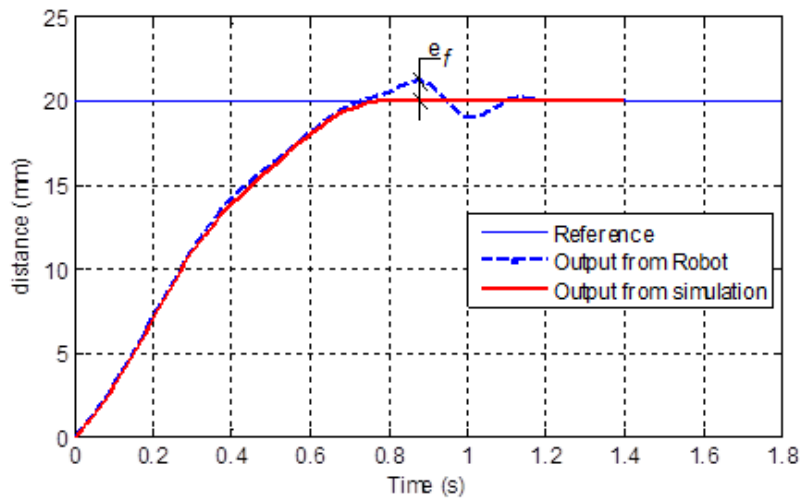


Figure 4. Output from simulation and real robot

After each loop, the output result of real manipulator will be recorded and compare with that of simulating manipulator using PID. The results of experiment will be shown in Fig. 6a to 6d. After each loop, we change vector u_f to the

optimal direction. Value e_f is from 2.2 mm in first experiment (Fig. 4) to 0.1 mm in the last experiment (Fig. 6d). With this result, we can almost obtain the controller for a manipulator with an output same as simulation.

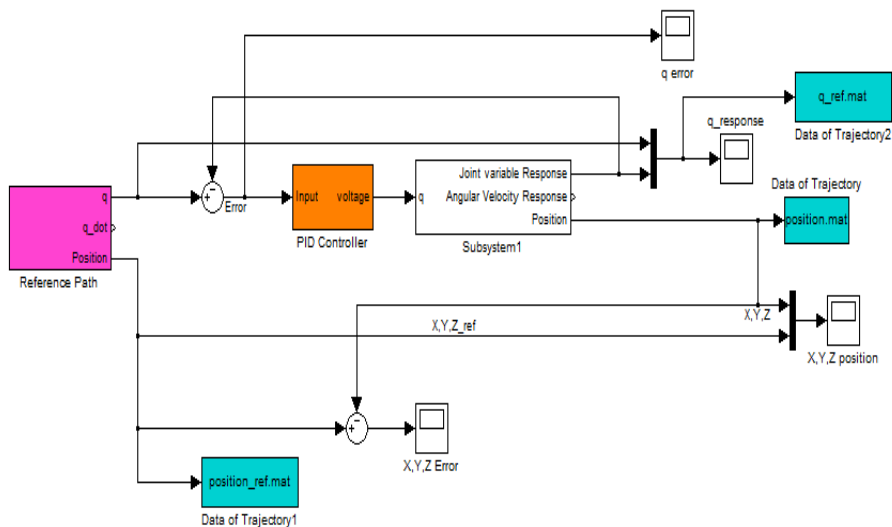


Figure 5. Design of PID controller

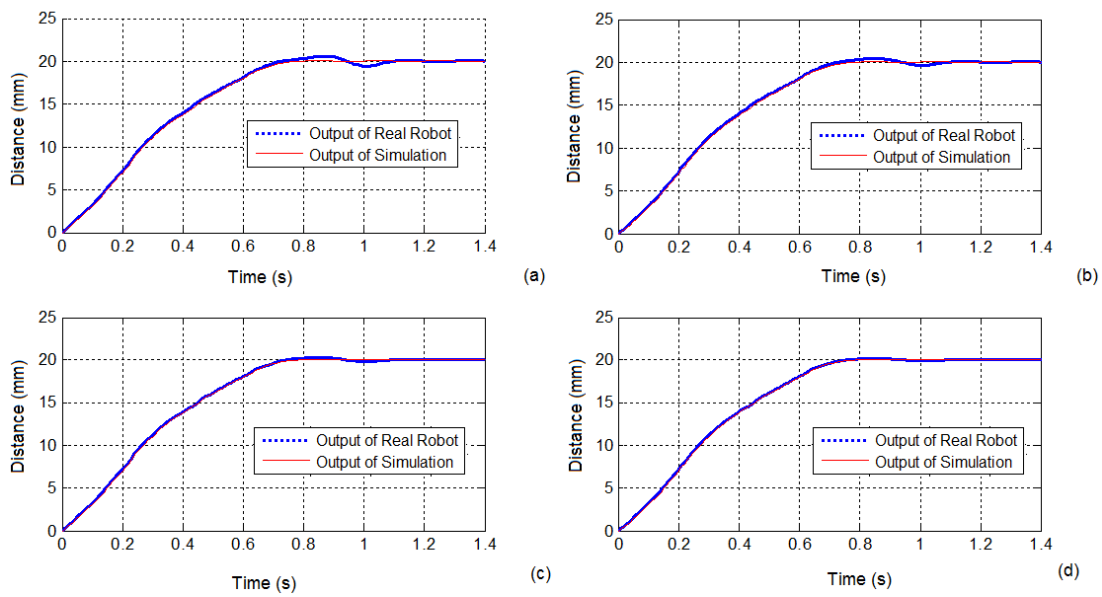


Figure 6. Result of two outputs, real Robot and simulation. u_f change from 0 to optimization, figure (a) to (d).

6.CONCLUSION

This paper introduces an algorithm to find a better controller from the traditional PID controller. The new controller considers a compensation for PID controller. The new controller will decrease the effect of unknown facts. From the experiment to control a

manipulator, the output due to new controller is almost same as the simulation result. When all curves are applied to the new controller, the compensator for each part of the curve is maybe different. Thus, we must make the experiment for each part to find the best compensator.

Một thiết kế tối ưu cho một bộ điều khiển PID tốt hơn

- Lưu Thanh Tùng
- Lê Nhân

Trường Đại Học Bách Khoa, ĐHQG-HCM

TÓM TẮT

Một bộ điều khiển của một tay máy đã được nghiên cứu và thảo luận trong nhiều năm. Tuy nhiên, nhiều vấn đề trong việc kiểm soát vị trí chính xác của các đầu cuối vẫn tiếp tục được nghiên cứu. Để giải quyết sự chính xác của robot, hai vấn đề cần quan tâm. Điều đầu tiên là phải tìm ra mô hình chính xác của phương trình động lực học. Điều thứ hai là một bộ điều khiển theo luật đã định. Tuy nhiên, điều rất khó khăn để có thể tìm ra một

mô hình chính xác hoặc phương trình vi phân chuyển động chính xác cho tay máy thực sự. Ngoài ra, một số ảnh hưởng không thể xác định được sẽ tạo ra sự khác biệt giữa lý thuyết và thực tế. Do đó, một thuật toán điều khiển sẽ được giới thiệu với bộ điều khiển PID có hệ số K_p , K_d , K_i được bù đắp bằng hệ số bù tìm thấy từ thuật toán tối ưu hóa. Với các thuật toán mới, các kết quả đã chứng minh sự ổn định và độ chính xác tốt hơn.

REFERENCES

- [1]. Mirela TRUȘ CĂ, Gheorghe LAZEA, "Adaptive controller for robot arms", Journal of Control engineering and applied informatics, Vol. 4, No. 3, pp. 13 – 18, 2002.
- [2]. Hassan Emara, Abdel Latif Elshafei, "Robust robot control enhanced by a hierarchical adaptive fuzzy algorithm", Journal of Engineering Applications of Artificial Intelligence, Vol. 17, pp. 187 – 198, 2004.
- [3]. Tzyh-Jong Tarn, Antal K. Bejczy, Xiaoping Yun, and ZUOfeng Li, "Effect of Motor Dynamics on Nonlinear Feedback Robot Arm Control", IEEE transactions on robotics and automation, Vol. 7, No. 1, 1991.
- [4]. Chin Su Kim, Kang Woong Lee, "Robust control of robot manipulators using dynamic compensators under parametric uncertainty", International Journal of Innovative Computing, Information and Control. Vol. 7, No. 7(B), 2011.
- [5]. Alexei ZAKHAROV, Sándor H ALÁSZ, "Parameter identification of a robot arm using genetic algorithms", Periodica polytechnica serby el. Eng, Vol. 45, No. 3–4, pp. 195–209, 2001.
- [6]. Hassan Emara, Abdel Latif Elshafei, "Robust robot control enhanced by a hierarchical adaptive fuzzy algorithm", Engineering Applications of Artificial Intelligence 7, pp. 187 – 198, 2004.
- [7]. Mark W. Spong, Seth Hutchinson, M. Vidyasagar, Robot Modeling and Control, John Wiley & Sons, Inc.
- [8]. Singiresu S. Rao, Engineering Optimization, John Wiley & Sons, Inc, 2009.