This is the author's version of an article that has been published in the MHI 2019 proceedings.
Changes were made to this version by the publisher prior to publication.
The final version of record is available at https://dx.doi.org/10.1007/978-3-662-59317-2_22

Optimization of a P/PI Cascade Motion Controller for a 3-DOF Delta Robot

Ahmed Zidan[1], Daniel Kaczor[1], Svenja Tappe[1] and Tobias Ortmaier[1]

Gottfried Wilhelm Leibniz Universität Hannover, Institute of Mechatronic Systems
Appelstraße 11a, 30176 Hannover
ahmed.zidan@imes.uni-hannover.de,
WWW home page: http://www.imes.uni-hannover.de

Abstract. An auto-tuning method for a Delta robot's P/PI cascade motion controller using multi-objective optimization algorithm is proposed. The implemented control structure consists of two controllers: A feedforward controller based on a model of the inverse dynamics of the robot, and a cascade P/PI controller to compensate for unmodeled effects. The auto-tuning is achieved in the sense of optimizing the control parameters in three stages. In the first stage, the feedback control parameters are optimized after neglecting the feedforward control term. The goal is to minimize the position error in tracking an excitation trajectory, which is used as well to identify the dynamic model parameters in the second stage. After that, the feedforward compensation term is computed offline based on the desired trajectory. In the final stage, the P/PI parameters are optimized again after adding the feedforward controller. Experimental results on an industrial 3-dof Delta robot validates the efficiency of the proposed method.

Keywords: Multi-objective optimization algorithms, P/PI control, Dynamics identification, Auto-tuning

1 Introduction

Parallel robots are implemented recently in a wide range of industrial applications. Their advantages over serial robots regarding high speed movements and accurate motion control made them the favorable choice for tasks with high dynamical movements such as pick and place.
The mostly used control structure for robot manipulators (both parallel and serial robots) consists of PID/PD controller combined with a compensation term, which aims to linearize the dynamics of the robot based on previous knowledge of robot dynamics. To add the compensation term, a partial or a full model of the inverse dynamics is required. Based on this, the modeled robot dynamics can be compensated in two different ways: In the first method, a simple feedforward controller is determined based on the desired values of position, velocity, and acceleration [7]. In the second method, a control structure known as computed torque control is implemented, where the compensation term is determined using real measurements of the position and velocity and the desired value of

the acceleration [8]. While the computed torque control is a more effective control structure in theory, it requires high computational power compared to the feedforward controller, which can be defined offline and then added to the PID controller.

Regardless of the chosen control structure, two steps are essential in designing an adequate controller: Firstly, modeling the inverse dynamics of the robot and, secondly, tuning the PID controller gains. Because of the complexity of the robot's dynamics, it is very hard to determine a perfect model that count for all components affecting the robot movement, such as friction, joints and links flexibilities and external loads. These model uncertainties are left for the PID controller to compensate for.

PID controllers for robot manipulators are tuned in most cases manually by trials and errors, which cannot guarantee having the best set of gains especially for robots with high degree of freedom and significant coupling effects between its links. Recently, global optimization methods are used to solve controller tuning problems depending on iterative algorithms that optimize the controller parameters with respect to a predefined objective function. In the field of robot manipulators, several optimization algorithms have been implemented as an auto-tuning method of PID controllers (e. g. Genetic Algorithms (GA) [4], Particle Swarm Optimization (PSO) [13], Differential Evolution (DE) [9]). Controller optimization in the given examples was done with respect to a single objective function, such as the integral of the absolute tracking error. However, minimizing the tracking error comes with the cost of generating control actions with extremely high variations which have dangerous impact on the joint motors. This problem can be solved by handling the auto-tuning as a multi-objective optimization problem. In this case, one objective function aims to minimize the trajectory tracking error, while the other function aims to minimize the variations of the controller output, and therefore, provide the controller with an additional stability margin. One of the works introducing this approach is [1], utilizing a multi-objective evolutionary algorithm (MOEA). In [6], a comparative study between different multi-objective optimization techniques has been introduced and an improved multi-objective particle swarm optimization (I-MOPSO) has been proposed. While in [14], a comparative study between multi-objective particle swarm optimization (MOPSO) and multi-objective cuckoo search (MOCS) has been introduced. MOCS is found to be more effective in preforming the auto-tuning task.

The inverse dynamics identification was a subject of many researches in the previous decades. Important contributions have been introduced regarding defining the base parameter set of the model [5], and optimizing the excitation trajectory of these parameters [3], among others. A general overview on dynamics identifications of robot manipulators can be found in [10].

In this work, an auto-tuning of a cascade P/PI controller in combination with a feedforward controller for a 3-dof Delta robot is introduced. The main contribution of this work is that the auto-tuning and the identification are combined in one overall process, where the P/PI control parameters are optimized in two

stages: First, neglecting the feedforward control, the goal is to achieve a reasonably accurate tracking of the excitation trajectory, which leads in turn to an accurate identification of the inverse dynamics. Secondly, the identified model is implemented as a feedforward control and the auto-tuning is repeated in order to achieve the optimal trajectory tracking under consideration of the feedforward control.
The rest of the paper is organized as follows. The second section explains the auto-tuning concept with a review of the chosen optimization algorithm. The third section describes the dynamics identification process. The fourth section introduces the proposed controller optimization method. Finally, the experimental results are demonstrated and final conclusions are given.

2 Auto-tuning of PID controller using a global optimization algorithm

Optimization algorithms are a very effective techniques in performing tuning of control parameters for complex nonlinear systems, where it is hard to obtain an accurate model of the controlled plant and to estimate the control parameters analytically. The main advantage of these algorithms is that they handle the system as a black box, and perform the tuning through an iterative procedure. The parameters are set depending on one or more objective functions that evaluate the performance of the controller. The most important objective for a motion controller is to achieve the accuracy in tracking the desired position, but it is also essential to guarantee other factors related to the stability and the safety of the controlled system. For this sake, searching for the optimal control parameters must be done under a number of predefined constraints.
In the case of robot manipulators, the tuning requires setting the parameters, performing the desired movement, evaluating the performance, and then repeating this procedure for a number of iterations. The constraints in this case are to avoid generating torques on the joint motors higher than its saturation limits, and also to avoid unstable movements that exceed a higher limit of position error. It is also important to avoid exciting high oscillations, which can be amplified by inappropriate gain values. In [13], a practical approach is proposed to handle these constraints, where the movement of the robot is observed in real time and stopped immediately as soon as one of the constraints is violated.
In this work, the auto-tuning is handled as a multi-objective optimization problem with two objective functions. The first one is the integral of the absolute error, which is obtained in practice as given by

$$IAE = \sum_{i=1}^{M} \sum_{j=1}^{N} |e_j(i)| \Delta T \,, \tag{1}$$

with e_j being the position error signal of joint j, ΔT being the sampling time, M being the total number of measurement samples during the robot movement, and N being the number of joints. The goal is to minimize this function in order

to achieve the best accuracy of the trajectory tracking. The second objective is defined to measure the variations of control action (i.e. motor torque) as given by

$$CAV = \sum_{i=1}^{M-1} \sum_{j=1}^{N} |\tau_j(i+1) - \tau_j(i)|\,, \tag{2}$$

where τ_j is the motor torque of joint j. Minimizing this function helps the controller to gain a stability margin and reduces the possibility of inducing high oscillations. The two objective functions contradict each other in the sense that reducing one leads most likely to increasing the other. Therefore, a set of optimal solutions is found at the end of the searching called the Pareto solutions or the Pareto frontier. Among these solutions only one will be chosen to be the controller gains. A reasonable choice is the solution that produces the best compromise between the two objectives. This solution is given by

$$bcs = \min\left(\frac{IAE(P) + CAV(P)}{2}\right), \tag{3}$$

with P being the resulting set of Pareto solutions.
The auto-tuning is performed in this work using the cuckoo search algorithm, which is briefly discussed in the next subsection.

3 Cuckoo search and Multi-Objective Cuckoo search

Cuckoo search (CS) is an optimization algorithm first introduced in [11]. It imitates the brood parasitism behavior of some cuckoo species. These cuckoo birds lay their eggs in the nests of other birds. The hosts nests are represented as the population of the algorithm, where the number of hosts will be fixed through the search. The cuckoo birds are represented by the new individuals generated in every iteration assuming that every cuckoo bird lays only one egg at a time. The new egg replaces the original egg in the host nest if it shows a better evaluation (cost value). In addition, the host nest has the ability to detect the intruder egg and get rid of it, which is modeled by a probability factor $pa \in [0$ $1]$. This factor is actually the only parameter that needs to be tuned manually for the algorithm.
In addition, CS algorithm uses a strong tool in generating new eggs based on the concept of Lévy flights [2], as follows:

$$x_i(t+1) = x_i(t) + \alpha \oplus L\acute{e}vy(\beta)\,, \tag{4}$$

where α depends on the difference between solution qualities, the product $\oplus$ is an entry-wise multiplication, and $L\acute{e}vy(\beta)$ is a function providing a random walk as follows:

$$L\acute{e}vy \sim u = t^{-1-\beta}\,, (0 < \beta \leq 2)\,. \tag{5}$$

In [12], the single objective CS algorithm is extended to handle optimization problems with multiple objectives. In order to achieve that, the analogy to the

cuckoo behavior is modified in which every objective function is represented by an egg in the nest, i.e. in k objectives problem, every nest will have k eggs, and each egg has its own quality (cost value). This approach is the one used in this work to perform the auto-tuning method.

4 Modeling and identification of the inverse dynamics

Modeling the dynamics of a parallel robot requires dividing it into separated kinematic chains and a moving platform. After that, the dynamical model of each of these subsystems is determined using one of the known methods such as Newton-Euler method or Lagrange method.
The resulting model can be simplified to a linear model with respect to a vector of base parameters $\boldsymbol{P}$ as given by:

$$\boldsymbol{\tau} = \boldsymbol{\Phi} \cdot \boldsymbol{P}\,, \tag{6}$$

with $\boldsymbol{\tau}$ being a vector of the measured joint torques, and $\boldsymbol{\Phi}$ being the regression matrix, whose elements are functions to the desired position, velocity, and acceleration.
Identifying the base parameters in this model can be done using the least squares method. The solution is given by:

$$\boldsymbol{P} = \boldsymbol{\Phi}^{\dagger} \cdot \boldsymbol{\tau} \tag{7}$$

with $\boldsymbol{\Phi}^{\dagger}$ being the pseudo-inverse of the regression matrix. This pseudo-inverse can only be determined if the regression matrix is well conditioned. Therefore, the desired trajectory must be chosen in a way that excites all the dynamical components of the robot and generates a well conditioned regression matrix. In this work, the excitation trajectory is defined using a Fourier series. The Fourier coefficients are defined with help of an optimization algorithm aims to minimize the condition number of the regression matrix as described in [3].

5 Optimization of P/PI cascade motion control with feedforward compensation

P/PI cascade motion control is an advanced PID position control. The main advantage of this controller is that it has two control loops, an outer loop to control the position, and an inner loop to control the velocity, which makes it more effective in compensating disturbances. This controller can be described by the following equations:

$$\boldsymbol{v}_{\mathrm{d}} = \boldsymbol{K}_{\mathrm{p}} \cdot (\boldsymbol{q}_{\mathrm{d}} - \boldsymbol{q})\,, \tag{8}$$

$$\boldsymbol{\tau}_{\mathrm{c}} = \boldsymbol{K}_{\mathrm{v}} \cdot (\boldsymbol{v}_{\mathrm{d}} - \dot{\boldsymbol{q}}) + \boldsymbol{K}_{\mathrm{i}} \cdot \int_0^T (\boldsymbol{v}_{\mathrm{d}} - \dot{\boldsymbol{q}})\mathrm{d}t. \tag{9}$$

With $\boldsymbol{K}_\mathrm{p}$, $\boldsymbol{K}_\mathrm{v}$, and $\boldsymbol{K}_\mathrm{i}$ being diagonal matrices of controller gains, $\boldsymbol{q}_\mathrm{d}$ and $\boldsymbol{q}$ being vectors of the desired and actual joint angles, $\dot{\boldsymbol{q}}$ being a vector of the actual angular velocities, and T being the total time duration of the movement. The actual velocity is estimated from the actual position using a first order filter. In this work, the cascade controller is combined with a feedforward compensation term based on the inverse dynamics model as shown in Fig. 1. For complex

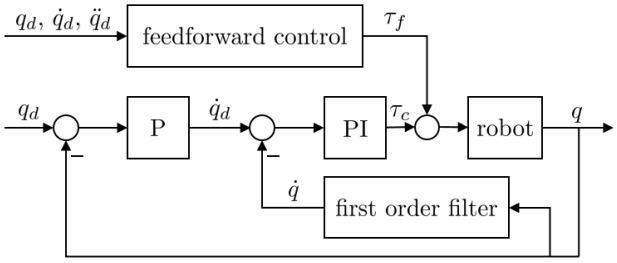

Fig. 1. Motion controller of the Delta robot

nonlinear systems such as robot manipulators, it is hard to obtain an accurate model of the inverse dynamics that counts for all dynamical forces affecting the robot motion. Which in turn complicates the procedure of tuning the P/PI cascade controller. The main contribution of this work is to propose a practical approach for optimizing this motion controller for a 3-dof Delta robot. However, it is justifiable to declare that this method can be applied to industrial robot manipulators in general. The auto-tuning procedure is performed in three steps:

1. Optimization of the P/PI controller using MOCS neglecting the feedforward controller as described in section 2. The desired trajectory used here is additionally used as excitation trajectory for the next step.
2. Identification of the inverse dynamics using the torques measurements of the joints as described in section 3. In this stage, the previously tuned P/PI controller is implemented.
3. Optimization of the P/PI controller as done in the first step but after adding the feedforward controller.

6 Experimental results

The proposed auto-tuning method is applied on a 3-dof Delta robot controlled by previously introduced motion controller. The used kinematic is an industrial Codian D4-1100 Delta robot designed for high dynamic pick and place applications (Fig. 2). The controller is applied using a standard industrial PLC and servo inverters. The MOCS algorithm is set with the following parameters:

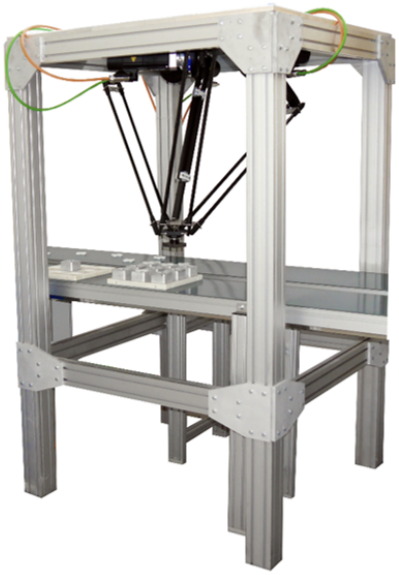

Fig. 2. Experimental setup with 3-dof Delta robot

- Number of nests = 10
- Maximal number of iterations = 15
- Searching space dimension = 9 (3 control parameters to be tuned for each of the 3 axes of the Delta robot). The optimization parameters are normalized depending on the maximum and the minimum gains provided from the manufacturer.

The whole number of executions of the desired movement is limited by the parameters of the MOCS to a total number of 150 for every tuning process. This equals about 1 hour of measurement time. After the first auto-tuning stage, the trajectory tracking is improved compared to the default control settings. The resulting position error signals are shown in Fig. 3. Measurements of the joints torques corresponding to the best achieved tracking accuracy in stage 1 are used to identify the inverse dynamics. Fig. 4 shows that the torque estimation generated by the inverse dynamics model is very close to the actual torque values. Finally, the determined model is implemented as a feedforward control, and the auto-tuning is performed again. A significant improvement is achieved compared to the first stage, which validates the efficiency of the proposed method. The position error signals after the final stage are shown in Fig. 5. At the end of the first and the last tuning stage, three Pareto solutions are generated. These solutions are given in Table 1. This number of solutions is relatively low compared to traditional multi-objective optimization problems. The main reason is that the effects of the control gains on the second objective function (variations of control

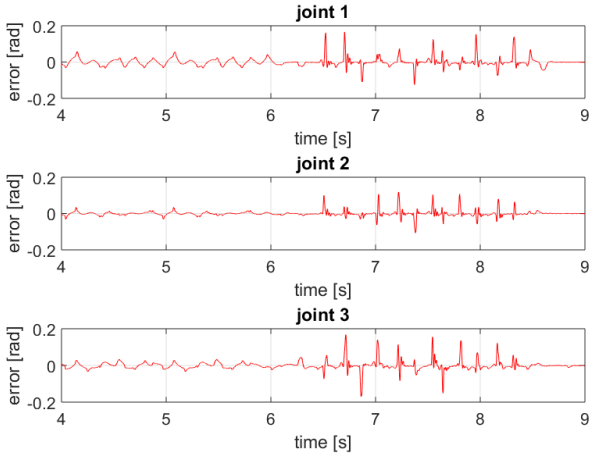

Fig. 3. Position error signals after the first auto-tuning stage

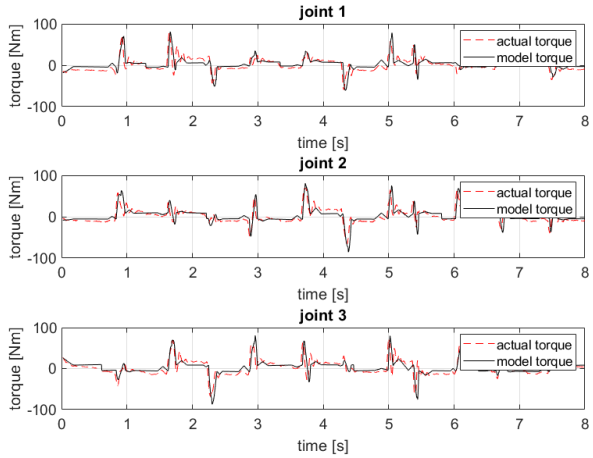

Fig. 4. Estimated torque signals compared to the actual torques

action) is relatively small. As a result, the Pareto frontier is narrow with respect to this objective function. However in general, controller gains may have higher impact on the motor torques variations if a larger search space is employed, therefore, it is essential to take the second objective function into consideration. In addition, choosing the controller gains based on the best compromise solutions enabled the optimization algorithm to obtain an accurate tracking and to

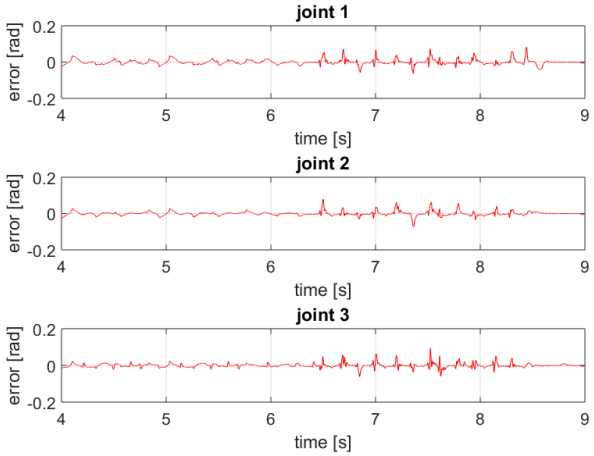

Fig. 5. Position error signals after the final auto-tuning stage

Table 1. Pareto Solutions for the first and final stage

	Pareto Solutions P/PI controller			**Pareto Solutions P/PI with Feedforward**		
	Solution 1	**Solution 2**	**Solution 3**	**Solution 1**	**Solution 2**	**Solution 3**
IAE	0.7066	0.5054	0.6253	0.445	0.3527	0.3469
CAV	0.9807	0.9888	0.9812	0.9788	0.9876	0.9877

Table 2. Optimal normalized controller gains for the first and final stage

	$\mathbf{K_{p1}}$	$\mathbf{K_{v1}}$	$\mathbf{K_{i1}}$	$\mathbf{K_{p2}}$	$\mathbf{K_{v2}}$	$\mathbf{K_{i2}}$	$\mathbf{K_{p3}}$	$\mathbf{K_{v3}}$	$\mathbf{K_{i3}}$
Stage 1	0.61	0.98	0.24	0.44	1	0.19	0.91	0.87	0.77
Stage 2	0.51	1	0.78	1	1	0.43	0.73	0.96	0.82

avoid exciting dangerous oscillations at the same time.The solutions that give the best compromise between the two objective functions are "Solution 2" for the first tuning stage and "Solution 3" for the final stage. The values of IAE corresponding to these two solutions indicates a 30% improvement of the tracking accuracy after adding the feedforward controller and repeating the optimization procedure. The optimal normalized gains are given in Table 2 for the first and the last tuning stage.

7 Conclusion

A novel auto-tuning method for a P/PI cascade motion controller of a 3-dof Delta robots was proposed. The auto-tuning is achieved with the help of a

global optimization algorithm. The problem is described as a multi-objective optimization problem and solved with multi-objective cuckoo search algorithm (MOCS). Auto-tuning is combined with a dynamic identification method to obtain an accurate model of the inverse dynamics. This model is implemented as a feedforward controller added to the P/PI control loop. The auto-tuning of the control parameters combined with the dynamics identification improves the trajectory tracking performance significantly, which is proved by the presented experimental results.

References

1. Ayala, H.V.H., dos Santos Coelho, L.: Tuning of pid controller based on a multiobjective genetic algorithm applied to a robotic manipulator. Expert Systems with Applications **39**(10), 8968–8974 (2012)
2. Barthelemy, P., Bertolotti, J., Wiersma, D.S.: A lévy flight for light. Nature **453**(7194), 495 (2008)
3. Do Thanh, T., Kotlarski, J., Heimann, B., Ortmaier, T.: Dynamics identification of kinematically redundant parallel robots using the direct search method. Mechanism and machine theory **55**, 104–121 (2012)
4. Kim, E.J., Seki, K., Iwasaki, M., Lee, S.H.: Ga-based practical auto-tuning technique for industrial robot controller with system identification. IEEE Journal of Industry Applications **1**(1), 62–69 (2012)
5. Mayeda, H., Yoshida, K., Osuka, K.: Base parameters of manipulator dynamic models. IEEE Transactions on Robotics and Automation **6**(3), 312–321 (1990)
6. Pierezan, J., Ayala, H.H., da Cruz, L.F., Freire, R.Z., Coelho, L.d.S.: Improved multiobjective particle swarm optimization for designing pid controllers applied to robotic manipulator. Computational Intelligence in Control and Automation (CICA), 2014 IEEE Symposium on pp. 1–8 (2014)
7. Santibañez, V., Kelly, R.: Pd control with feedforward compensation for robot manipulators: analysis and experimentation. Robotica **19**(1), 11–19 (2001)
8. Slotine, J.J.E., Li, W.: On the adaptive control of robot manipulators. The international journal of robotics research **6**(3), 49–59 (1987)
9. Villarreal-Cervantes, M.G., Alvarez-Gallegos, J.: Off-line pid control tuning for a planar parallel robot using de variants. Expert Systems with Applications **64**, 444–454 (2016)
10. Wu, J., Wang, J., You, Z.: An overview of dynamic parameter identification of robots. Robotics and computer-integrated manufacturing **26**(5), 414–419 (2010)
11. Yang, X.S., Deb, S.: Cuckoo search via lévy flights pp. 210–214 (2009)
12. Yang, X.S., Deb, S.: Multiobjective cuckoo search for design optimization. Computers & Operations Research **40**(6), 1616–1624 (2013)
13. Zidan, A., Kotlarski, J., Ortmaier, T.: A practical approach for the auto-tuning of pd controllers for robotic manipulators using particle swarm optimization. 14th International Conference on Informatics in Control, Automation and Robotics pp. 34–40 (2017)
14. Zidan, A., Tappe, S., Ortmaier, T.: A comparative study on the performance of mopso and mocs as auto-tuning methods of pid controllers for robot manipulators. In: Proceedings of the 15th International Conference on Informatics in Control, Automation and Robotics - Volume 1: ICINCO,, pp. 240–247. INSTICC, SciTePress (2018). DOI 10.5220/0006899802500257