Journal of Business Research 68 (2015) 1429–1431

Contents lists available at ScienceDirect

Journal of Business Research

Improving productivity using a multi-objective optimization of robotic trajectory planning☆

Carlos Llopis-Albert *, Francisco Rubio [1], Francisco Valero [1]

Universitat Politècnica de València, Spain

ARTICLE INFO

Article history:
Received 8 April 2014
Received in revised form 15 October 2014
Accepted 12 November 2014
Available online 11 February 2015

Keywords:
Robotics
Multi-objective optimization
Trajectory planning
Pareto frontier
Scheduling

ABSTRACT

This study presents a methodology to tackle robot tasks in a cost-efficient way. It poses a multi-objective optimization problem for trajectory planning of robotic arms that an efficient algorithm will solve. The method finds the minimum time to perform robot tasks while considering the physical constraints of the real working problem and the economic issues participating in the process. This process also considers robotic system dynamics and the presence of obstacles to avoid collisions. It generates an entire set of equally optimal solutions for each process, the Pareto-optimal frontiers. They provide information about the trade-offs between the different decision variables of the multi-objective optimization problem. This procedure can help managers in decision-making processes regarding performing tasks, items to be manufactured or robotic services performed to meet with the current demand, and also, to define an efficient scheduling. It improves productivity and allows firms to stay competitive in rapid changing markets.

1. Introduction

An industrial robot is an automatically controlled, reprogrammable, and multipurpose manipulator that industrial automation applications use. A service robot is a robot that operates semi or fully autonomously to perform useful services for humans and equipment. World robots are rapidly growing in number in recent years. Process complexity deriving from automation requires efficient algorithms that control them to provide cost-efficient solutions (e.g., Kelly, Johnson, Dorsey, & Blodgett, 2004). Specifically, in recent years researchers are working hard in the trajectory planning of robot arms (e.g., Chen & Zhao, 2013; Chettibi, Lehtihet, Haddad, & Hanchi, 2002; Cho, Choi, & Lee, 2006; Gasparetto & Zanotto, 2010; Huang, Xu, & Liang, 2006; Suñer et al., 2007; Rubio et al., 2010; Rubio, Llopis-Albert, Valero, & Suñer, 2015). Furthermore, mathematical optimization techniques solve many engineering problems (e.g., Llopis-Albert & Capilla, 2010a, 2010b).

This study presents a new robotic technology to address robotic systems' cost-effectiveness through a multi-objective optimization problem for robotic arm trajectory planning, which an efficient algorithm solves. The method finds the minimum time trajectory to perform robot tasks while considering the physical constraints and the economic issues participating in the process. The methodology also allows analyzing the trade-off between the different decision variables through the Pareto-optimal frontiers. A solution belongs to the Pareto optimal frontier if an objective does not improve without adversely affecting at least one other objective. This methodology allows an immediate change, a quality improvement of the products, an increase in productivity, and a reduction of cycle times, which may increase opportunities to react to market developments and receptivity. The procedure overcomes the limitations of economic analysis methods that can currently assess robotic systems cost-effectiveness in production lines and robot services.

2. Multi-objective optimization

Many real-world design tasks involve complex multi-objective optimization problems of various competing design specifications and constraints that make a single design highly improbable. Therefore, a trade-off among the conflicting design objectives is necessary. A multi-objective optimization affects several non-commensurable and often competing objectives, cost functions, or performance functions within a feasible decision variable space. This study follows above optimization model because, for example, a minimum time trajectory to produce an item leads to lower costs in energy consumption. Therefore, a trade-off exists between executable time and costs. The multi-objective optimization problem solves the collision-free trajectory-

☆ The authors thank the funding of Science and Innovation Ministry of the Spain Government by means of the Researching and Technologic Development Project DPI2010-20814-C02-01 (IDEMOV).

* Corresponding author at: Departamento de Ingeniería Mecánica y de Materiales (DIMM), Universitat Politècnica de València, Camino de Vera s/n, 46022 Valencia, Spain.
E-mail addresses: cllopisa@upvnet.upv.es (C. Llopis-Albert), frubio@mcm.upv.es (F. Rubio), fvalero@mcm.upv.es (F. Valero).
[1] Centro de Investigación en Ingeniería Mecánica (CIIM), Universitat Politècnica de València, Camino de Vera s/n, 46022 Valencia, Spain.

http://dx.doi.org/10.1016/j.jbusres.2015.01.027
0148-2963/

planning problem of robotic arms while considering the economic issues participating in the process. The algorithm, according to previous works (Rubio, Valero, Suñer, & Mata, 2009; Rubio, Valero, Suñer, & Cuadrado, 2012; Rubio et al., 2015; Valero, Mata, & Besa, 2006), returns robot's minimum total traveling time. This time has to do with productivity and flexibility, because it accelerates operation or execution time of the process. Problem constraints are the torque, power, jerk (variables to do with work quality, accuracy, and equipment maintenance), and energy consumption (related to savings). Optimization problem constraints require a fulfillment because minimum-time algorithms have discontinuous values of acceleration and torques leading to dynamic problems during trajectory performance. The imposition of smooth trajectories can solve the problem by using spline functions in path and trajectory planning. The jerk constraint is crucial for working with precision and without vibrations, and affects control system and joints and bars' wearing. These methods enable the errors, the stresses (in robot's actuators and mechanical structure), and the resonance frequencies to shrink during trajectory tracking.

The economic objective function is the following:

$$\text{Max } B = \frac{1}{(1+r)^T}\left[\sum_{p=1}^{n}\left(P_p - C_p\right)\cdot N_p\right] \tag{1}$$

where B is the objective function that requires maximization and represents the current value of the net benefit from a generic robot task (€), which is the revenue of the items or services performed minus total costs; P_p is the market unitary price of the item or service p (€); r is the annual discount rate; T represents the number of years; C_p is the unitary cost to perform the task p (€), ranging from costs of raw materials, energy, amortization, labor force, and maintenance, taxes to direct and indirect costs; and $N_p(t)$ is a function accounting for the number of tasks per hour:

$$\mathrm{N_p(t)} = \frac{K}{t(S_k)^{\mu}}. \tag{2}$$

Tasks' set S_k to perform the item or service (p) constitutes the work load, where k represents tasks' number. In $t(S_k) = \sum_{j \epsilon S_k} t_j$ (task time), K is a constant that has to do with the current number of working hours per year. Parameter μ refers to economic environment and market seasonality.

The robot arm develops each of these tasks, using certain time to describe the optimal trajectory. Then the algorithm returns robot arm's minimum time to perform the task p (t_{minp}), while considering the time of the other tasks as constant. The lower the time that robot uses to perform its task, the greater the number of tasks per hour. Then, the cumulative time of all tasks is the following:

$$t(S_k) = t_{minp} + \sum_{j \notin S_{robot}}^{k} t_j. \tag{3}$$

3. Results of the application of methodology to different examples

This study applies multi-objective optimization methodology to different examples following those by Rubio et al. (2012). This study uses as a model the PUMA 560 robot, which stands for Programmable Universal Machine for Assembly.

Five examples provide positive results with sequences between 32 and 57 intermediate configurations between the initial and final ones, using different physical working environments (see Rubio et al., 2012). The robot uses different working constraint values for each actuator. Table 1 presents algorithm results, that is, the execution time for the robot to perform robot task trajectory.

Subsequently, this study analyzes the economic issues regarding robot tasks. The study supposes a task cost of 0.8 € (without considering energy cost) and a price of 1€ for the five examples. When the study considers energy consumption cost, the examples have different costs. This research defines and adds a cost of 0.0676 €/kWh to the total costs of 0.8 €. For clarity purposes, only one shift of 8 h appears (365 working days in a year), and benefits B cover a period of one year. Different number of items or services per year arise for each case, because they present different minimum execution times (t_{min}). The time of the other tasks to perform the item or service (i.e., the summation of times in Eq. (3), $\left(\sum_{j \notin S_{robot}}^{k} t_j\right)$ is 90 s. Therefore, different cases present different benefits. For instance, the case 3_s_s, which has no constraints in both the jerk and the energy consumed, presents the maximum benefits per year (23243 €). Conversely, case 2_5_95, with severe physical constraints, shows the minimum benefits (22962 €).

As an additional example, this study considers the performance of three different services. This exercise illustrates benefit loss because of not using efficient algorithms. The Pareto frontiers represent this benefit loss for the three different services. The services differ in their cumulative performance time but share the same execution time of the robot arm (t_{minp}). Then the study uses the minimum trajectory time of case 3_s_s for all items, i.e., 2.27 s. The cumulative time of the Service 1 = 90 s; Service 2 = 100 s; and Service 3 = 80 s. These services also differ in the total costs (without considering energy costs), prices, and values of the parameter μ, which intends to simulate different economic environments and market seasonality. Then, the total cost of Service 1 = 0.8 €; Service 2 = 0.82 €; and Service 3 = 0.84 €, while the prices are Service 1 = 1.0 €; Service 2 = 1.05 €; and Service 3 = 1.02 €. The parameter μ takes the values of 0.6, 0.5, and 0.55, respectively for each service. In this case, $t(S_k)$ is a cubic function of t_{minp}.

In this scenario where the market conditions do not change and without using the efficient algorithm a benefit loss exists. This is because robot arm may present higher execution times. Instead, the multi-objective optimization algorithm allows obtaining the Pareto frontiers (i.e., the minimum time to perform the robot tasks), while providing information about decision variable trade-offs.

The trade-off between the benefits and the execution time for the case 3_s_s (i.e., the Pareto frontier) appears in Fig. 1. Note that the results obtained with the optimization procedure lead to lower working times and therefore greater annual revenues.

Table 1
Execution times (s) for the different examples solved with physical constraints.

Case	Execution time (s)	Case	Execution time (s)
1_s_s	3.79	4_5_s	18.28
1_s_75	22.55	4_10_s	14.51
1_5_s	19.27	4_25_s	10.69
1_5_75	25.76	4_5_s	18.28
2_s_s	5.14	4_10_s	14.51
2_s_200	5.15	4_25_s	10.69
2_s_175	5.3	4_50_s	8.49
2_s_150	5.62	4_100_s	6.74
2_s_125	6.42	4_1000_s	3.21
2_s_100	12.25	4_s_s	2.41
2_s_95	21.08	4_5_40	18.65
2_5_s	23.05	4_s_40	9.94
2_5_95	26.35	5_s_s	3.08
3_s_s	2.27	5_s_40	9.18
3_s_50	7.34	5_5_s	15.91
3_5_s	14.82	5_5_40	15.93
3_5_50	17.94		

Nomenclature used. Case: numberexample_*X*_*Y*. The first number indicates the example solved, the *X* position indicates the value of a physical constraint-jerk- and the *Y* position indicates the value of energy consumed. Letter *s* in any position means without that constrain.