mathematics

MDPI

Article

Modelling an Industrial Robot and Its Impact on Productivity

Carlos Llopis-Albert *, Francisco Rubio and Francisco Valero

Centro de Investigación en Ingeniería Mecánica (CIIM), Universitat Politècnica de València—Camino de Vera s/n, 46022 Valencia, Spain; frubio@mcm.upv.es (F.R.); fvalero@mcm.upv.es (F.V.)
* Correspondence: cllopisa@upvnet.upv.es

Abstract: This research aims to design an efficient algorithm leading to an improvement of productivity by posing a multi-objective optimization, in which both the time consumed to carry out scheduled tasks and the associated costs of the autonomous industrial system are minimized. The algorithm proposed models the kinematics and dynamics of the industrial robot, provides collision-free trajectories, allows to constrain the energy consumed and meets the physical characteristics of the robot (i.e., restriction on torque, jerks and power in all driving motors). Additionally, the trajectory tracking accuracy is improved using an adaptive fuzzy sliding mode control (AFSMC), which allows compensating for parametric uncertainties, bounded external disturbances and constraint uncertainties. Therefore, the system stability and robustness are enhanced; thus, overcoming some of the limitations of the traditional proportional-integral-derivative (PID) controllers. The trade-offs among the economic issues related to the assembly line and the optimal time trajectory of the desired motion are analyzed using Pareto fronts. The technique is tested in different examples for a six-degrees-of-freedom (DOF) robot system. Results have proved how the use of this methodology enhances the performance and reliability of assembly lines.

Keywords: adaptive fuzzy sliding mode control; controller; multi-objective optimization; robotics; trajectory planning; Pareto frontier; trade-offs; productivity assessment

check for updates

Citation: Llopis-Albert, C.; Rubio, F.; Valero, F. Modelling an Industrial Robot and Its Impact on Productivity. *Mathematics* **2021**, *9*, 769. https://doi.org/10.3390/math9070769

Academic Editor: António M. Lopes

Received: 28 February 2021
Accepted: 29 March 2021
Published: 1 April 2021

1. Introduction

Trajectory planning has applications in several fields, for instance in control theory, artificial intelligence or robotics. Optimal trajectory planning of industrial robots in assembly line is a key topic to boost productivity in a variety of manufacturing activities [1,2]. There is a wide range of algorithms in the literature to achieve the minimum time for efficient robot task [3]. An exhaustive review of such approaches can be found in [4,5].

Some approaches deal with the robot trajectory planning by seeking jerk-optimal trajectories [6]. Jerks play a major role to achieve accurate results and reduce vibrations. In addition, they have an influence on the control system and the premature wear of joints and bars. These techniques enable a decrease in errors regarding the optimal trajectory. Stresses in the driving motors and in the structure of the robot are also diminished. Another technique relies on minimizing the energy consumed and the torque rather than the jerk or trajectory time [7]. With robotics advances, trajectory planning covers more complex environments, for example, uncertainties [8], multiple robots [9], the dynamics of the system [10–12], optimal reconfiguration of a robot for forward singularities avoidance [13,14] and fuzzy control [15]. Furthermore, fuzzy control has been gaining attention in recent years [16–18]. This is because, in the last decades, the algorithms that tackled the robot trajectory planning had the drawback of the discontinuous behavior of some kinematic variables and torques. They induce dynamic incompatibilities while the trajectory is performed. These disadvantages were eluded by using smooth trajectories, for instance, spline functions. To efficiently plan the motion is vital to compute the system dynamics. However, due to the parameter uncertainties and the highly nonlinear dynamic parameters, there is a difference between the desired motion and the real motion obtained with these algorithms [19]. The discrepancies range from time-varying loads, parameter variations, errors in mounting the robot arms, malfunctioning of actuator forces or torques, overload or excess voltage, noise and disturbances. They are translated into loss of productivity and reliability in assembly lines, which may lead to robot downtimes and expensive unplanned stoppages for repair and maintenance works. To tackle this problem, robot controllers must provide real-time feedback to reduce this discrepancy. This real-time feedback provides the driving motors with the appropriate input forces and torques to perform the desired motion. Note that these inputs are different than the nominal ones for compensating for the above-mentioned uncertainties.

This paper uses an algorithm based on the real dynamics of assembly line robots to obtain the desired motion (i.e., the optimal trajectory planning) and links the performance of the robot to the productivity by means of the working times obtained, while avoiding the uncertainties related to the control system by using an adaptive fuzzy sliding mode control (AFSMC). This overcomes some of the limitations of the traditional proportional-integral-derivative (PID) controllers when applied to the dynamics of the robotic system. AFSMC can work under certain degrees of uncertainty related to the dynamic parameters of the system, thus giving a suitable performance and reducing idle time. This methodology has been exhaustively used in the literature [20,21]. This approach, founded on applied fuzzy logic, also overcomes two of the main drawbacks of pure sliding mode controller, such as nonlinear equivalent dynamic approach and chattering phenomenon in unknown dynamic parameters [19]. Further advantages of the AFSMC are to diminish the number of fuzzy rules and to enhance stability and robustness of the algorithm, while the main drawback is the definition of the sliding surface slope coefficient [22].

Additionally, the developed algorithm is collision-free, considers the restraint of energy consumed and fulfils the physical characteristics of the robot (i.e., torque, power and jerks for each driving motor). By taking into account the energy consumed during the robot performance enables companies to design environmentally sustainable strategies towards compliance with governmental greenhouse gas (GHG) emission standards and for climate change mitigation and adaptation policies. Additionally, the lower energy consumption leads business to become more competitive in the market. Furthermore, a multi-objective optimization has been posed for improving the productivity of assembly lines in terms of the optimal time needed to carry out a certain task. This allows one to deal with the trade-offs among the economic issues of an assembly line (such as supply and demand of the items produced, costs, prices and manpower) and the time spent to produce an item (i.e., the desired motion). The trade-offs are analyzed by means of Pareto fronts. It is worthwhile to mention that there are few works tackling the analysis of the associated costs of the operation of the assembly line. Therefore, this work goes a step further regarding the current literature by presenting an algorithm that allows improving the company's production to be improved and, hence, its profits to also be improved.

The methodology has been applied to the trajectory planning of assembly line robots. Results have shown how the proposed algorithm enhances the performance and reliability of assembly lines, thus helping companies remain competitive in a changing economic context.

The rest of the manuscript is organized as follows. Section 2 presents a brief description of the adaptive fuzzy sliding mode control (AFSMC). Sections 3 and 4 introduce how the kinematics and dynamics of the industrial robot and the assessment of company's productivity are implemented in the multi-objective optimization algorithm. Section 5 demonstrates the validity of the methodology to a set of experimental case studies. Finally, Section 6 provides the conclusions of this research.

2. Adaptive Fuzzy Sliding Mode Control (AFSMC)

An in-depth review of the design of an adaptive fuzzy sliding mode controller (AFSMC) to implement the desired trajectory tracking was presented in [19]. Furthermore, this technique was compared with other control schemes including sliding mode control (SMC) and fuzzy sliding mode control (FSMC). For the sake of conciseness, the reader is referred to this work for an in-depth explanation of the technique, while a brief overview of the sliding mode control is subsequently presented.

It is a potent non-linear technique to devise a controller providing a desired state by minimizing the tracking error (e), i.e., the discrepancy between the actual motion of a robot manipulator and the desired motion obtained with optimal trajectory planning algorithms. Robot controllers solve this problem using a real-time feedback, which provide the driving motors with the appropriate input forces and torques to track the targeted motion.

The dynamic equation of an m-link industrial robot is:

$$M(q)\ddot{q} \ + C\big(q,\dot{q}\big)\dot{q} \ + G(q) \ + \ T_d = \tau \tag{1}$$

where $q = [q_1, \ldots, q_m]^T$ is an $m \ x \ 1$ vector of joint position, $M(q)$ is an $m \ x \ m$ inertial matrix, $C\big(q,\dot{q}\big)$ is an $m \ x \ m$ matrix of Coriolis and centrifugal forces, G(q) is an $m \ x$ 1 gravity vector, $\tau = [\tau_1, \ldots, \tau_m]^T$ is an vector of driving torques and T_d is the vector of generalized input on account of unmodeled dynamics or disturbances.

As the system functions and external disruption are unknown in real systems, the control law is normally complex to be obtained. Fuzzy logic systems are used to tackle non-linear unknown functions and device an updating law.

Tracking error is defined as:

$$e = q - q_d \tag{2}$$

where e is the tracking error vector defined as the deviation between the state or joint position vector (q) of the robot manipulator and the desired state vector (q_d) obtained with the optimal trajectory planning algorithm presented in the next section. The technique can be broken down into two stages. The first one is the choosing of an adequate sliding surface (SS). In the second stage, a control law is devised to guide the system state toward the SS while ensuring the system stability. Then, the sliding mode control law is made up of two main components:

$$\hat{\tau} = \hat{\tau}_{eq} + \hat{\tau}_{dis} \tag{3}$$

where the component $\hat{\tau}_{eq}$ is the equivalent controller founded on the dynamics modelling of the robot arm; and $\hat{\tau}_{dis}$ is the discontinuous component of the sliding mode controller based on a saturation continuous function, which is intended to diminish the undesirable chattering phenomenon [23]. The former ($\hat{\tau}_{eq}$) is the vector of the driving torques obtained, as explained in the next section.

A sliding surface s is defined as:

$$s = \dot{e} + \lambda e \tag{4}$$

where $\lambda = \text{diag}[\lambda_1, \ldots, \lambda_i, \ldots, \lambda_n]$ is a diagonal vector representing the SS slope and λ_i $(i = 1,$ …, n) is a positive constant. Furthermore, the controller is intended to keep the SS slope s near to zero.

The control input term ($\hat{\tau}_{dis}$) is computed as follows:

$$\hat{\tau}_{dis} = K \ \text{sgn} \ (s) \tag{5}$$

where $K = \text{diag}[K_{11}, \ldots, K_{ii}, \ldots, K_{nn}]$ is a diagonal positive matrix and K_{ii} $(i = 1,$ …, n) is a positive constant and $sgn \ (s)$ is the switching function, which is defined as:

$$\text{sgn} \ (s) = \begin{cases} \ \ 1 \ \ s > 0 \\ -1 \ \ s < 0 \\ \ \ 0 \ \ s = 0 \end{cases} \tag{6}$$

The sliding mode controller with a discontinuous component has the disadvantage of the chattering phenomenon, which may be caused by output oscillation. The boundary layer method can be used to reduce or eliminate this phenomenon, which is based on the replacement of the discontinuous technique by the saturation method with small surroundings of the switching surface. This replacement leads to an improvement in the error.

$$B(t) \ = \ \{x, |s(t)| \leq ø\}; ø > 0 \tag{7}$$

in which ø is the boundary layer thickness. In order to have a smoother control law, a saturation function $Sat(s/ø)$ is added to the control law:

$$\hat{\tau}_{sat} = K{\cdot}\mathrm{Sat}(s/ø) \tag{8}$$

with $Sat(s/ø)$ defined as:

$$\mathrm{Sat}(s/ø) \ = \ \begin{cases} 1 \quad (s/ø > 1) \\ -1 \quad (s/ø < 1) \\ s/ø \ \ (-1 < s/ø < 1) \end{cases} \tag{9}$$

when replacing Equation (9) into (1), the control output term is:

$$\hat{\tau} = \hat{\tau}_{eq} + K{\cdot}\mathrm{Sat}(s/ø) \ = \ \begin{cases} \tau_{eq} + K{\cdot}\mathrm{sgn}\ (s), & |s| \geq ø \\ \tau_{eq} + K{\cdot}s/ø, & |s| < ø \end{cases} \tag{10}$$

The system performance is sensitive to the SS slope, so that obtaining the optimum value of λ is a key factor when designing a controller, that is, for a large λ value the system has a very fast response and is unstable, and conversely, for small values the system has a very slow response but is very stable. The stability and convergence for a m-link robot can be demonstrated theoretically using Lyapunov stability theory [24]. As result, the technique has two main advantages: to reduce the number of fuzzy rule and to enhance the stability and robustness.

In addition, a MIMO fuzzy system can be broken into a set of multi-input single output (MISO) fuzzy systems [25]. Then, the design of the controller a MISO fuzzy rule is comprised of well-known fuzzy if-then rules [22]. Eventually, it provides an output control law defined as:

$$\tau = \ \hat{\tau} + \tau_{\mathrm{fuzzy}(S)} \tag{11}$$

where $\hat{\tau}$ is the nominal compensation term and $\tau_{\mathrm{fuzzy}(S)}$ is the output of the controller. Eventually, the main advantage of the AFSMC is to compensate for parametric uncertainties, bounded external disturbances and constraint uncertainties, which leads to greater reliability and stability.

3. Time from the Trajectory Planner

The developed algorithm is intended to provide the minimum execution time of a multitasking robot manipulator. First, the working space is generated based on the geometrical characteristics of the problem and by means of feasible adjacent configurations [12]. Second, a minimum weighted free-collision path between the initial and final configuration is obtained. Third, the dynamic characteristics of the robot are considered to obtain the minimum time trajectory between adjacent configurations that is compatible with the robot features and the actuator capabilities. The inverse dynamic problem is solved using the Gibbs–Appell equation of motion, yielding a well-structured set of equations that can be computed in real time. This approach is used instead of more usual dynamic formulations, due to its advantages such as the simplicity of the obtained expressions and an easy model expansion in order to include additional effects as rotor dynamics or the friction phenomenon. In this sense, for generating the optimal time trajectory (t) between two different configurations, a robot configuration $C^{j} = C^{j}\ (\alpha_i^{j}, p_k^{j})$ is defined unambiguously using the Cartesian coordinates of important points in the robot $\alpha_i^{j} = (\alpha_{xi}^{j}, \alpha_{yi}^{j}, \alpha_{zi}^{j})$.

Time to travel from C^{i} to C^{f} is calculated by solving an optimization problem.

The optimization procedure can be defined in this way:

$$\text{Find } q(t), \tau(t), t_f \text{ between each two configurations} \tag{12}$$

$$\text{Minimizing } \underset{\tau \in \Omega}{min}\, J = \int_0^{t_f} dt \tag{13}$$

where $\tau(t) \in R^n$ is the vector of the driving torques and Ω is the space state in which the vector of the driving torques exists.

The constraints of the optimization problem are subsequently presented.

(1) Robot dynamics

$$M(q(t))\ddot{q}(t) \ + C\big(q(t), \dot{q}\ (t)\big)\dot{q}(t) \ + g(q(t)) \ = \tau(t) \tag{14}$$

(2) Boundary conditions (position, velocity and acceleration) for intermediate configurations

$$\begin{aligned} q(t_{int-1}) \ &= q_{int-1}; q(t_{int}) \ = q_{int} \\ \dot{q}(t_{int-1}) \ &= \dot{q}_{int-1}; \dot{q}(t_{int}) \ = \dot{q}_{int-1} \\ \ddot{q}(t_{int-1}) \ &= \ddot{q}_{int-1}; \ddot{q}(t_{int}) \ = \ddot{q}_{int-1} \end{aligned} \tag{15}$$

(3) Boundary conditions for initial and final configuration

$$\begin{aligned} q(0) \ &= q_o; q\left(t_f\right) \ = q_f \\ \dot{q}(0) \ &= 0; \dot{q}\left(t_f\right) \ = 0 \end{aligned} \tag{16}$$

(4) Collision avoidance with obstacles

$$d_{ij} \geq r_j + w_i \tag{17}$$

where d_{ij} is the distance from obstacle j to industrial robot i; r_j is the characteristic radius of the obstacle (sphere, cylinder, or prism) and w_i is the radius of cylinder that wraps the arm i.

(5) Driving torque

$$\tau_i^{\min} \leq \tau_i(t) \ \leq \tau_i^{\max}\ \forall t \in \ [0, t_{min}],\ i = 1,\ \ldots,\ dof \tag{18}$$

(6) Power in the driving motors,

$$P_i^{min} \leq \tau_i(t)\dot{q}_i(t) \ \leq P_i^{max}\ \forall t \in \ [0, t_{min}],\ i = 1,\ \ldots,\ dof \tag{19}$$

(7) Jerk on the driving motors,

$$\dddot{q}_i^{\,min} \leq \dddot{q}_i(t) \ \leq \dddot{q}_i^{\,max}\ \forall t \in \ [0, t_{min}],\ i = 1, \ldots,\ dof \tag{20}$$

($\dddot{q}_i$ is the jerk of actuator i)

(8) Energy consumed

$$\sum_{j=1}^{m-1}\left(\sum_{i=1}^{dof} \varepsilon_{ij}\right) \leq E, \tag{21}$$

where ε_{ij} is the energy used by the driving motor i to move from configuration $c^{\,j}$ to $c_{\,j+1}$.

Torque and power in each driving motor are tabulated in Table 1.

Table 1. Power and torque for each driving actuator (PUMA 560 industrial robot).

Driving Actuator	Maximum Torque (N·m)	Minimum Torque (N·m)	Maximum Power (W)	Minimum Power (W)
1	140	−140	275	−275
2	180	−180	350	−350
3	140	−140	275	−275
4	80	−80	150	−150
5	80	−80	150	−150
6	40	−40	75	−75

The jerk considered is 500 $\mathrm{rad/s^3}$. The maximum energy considered is different for each solved example.

An optimization problem is solved to search for the minimum time. Total time is defined using time increment $\Delta t_j = t_j - t_{j-1}$, at each interval, so that the objective function is

$$t_{\min} = \sum_{j=1}^{m-1} \Delta t_j$$

It is worthwhile to mention that the minimum execution time is related to productivity issues; the minimum jerk is related to the quality of work, the accuracy and equipment maintenance, while the minimum energy consumed (or minimum actuator effort) is related to savings. As explained, this research follows a combination of such criteria, since the execution time, the energy cnsumed, and the jerk are taken into account. The optimization problem is solved using sequential quadratic programming techniques (SQP), which is an iterative method for constrained nonlinear optimization where the objective function and the constraints are twice continuously differentiable. The algorithm has been implemented in FORTRAN.

4. Assessment of the Productivity of an Assembly Line

The multi-objective optimization algorithm is tested in several cases, which allows the improvement of production scheduling, while determining the trade-offs among the business profits and the optimal production times of the assembly line.

The productivity is determined by performing an economic analysis found on the working times of robotic tasks. A multi-objective function that considers the kinematics and dynamics of robotize production lines is posed. The algorithm finds the minimum time (i.e., the optimal trajectory to perform a certain industrial task), while meeting the physical restraints of the problem. In addition, the algorithm also considers the economic aspects concerning to the process. They cover the price and cost of the manufactured items (including costs of raw materials, amortization, energy, maintenance, labour force, taxes to direct and indirect costs); supply and demand of the items produced; the current socio-economic situation and seasonal variations; the number of working hours, employees and shifts per year; and profits. Then, the algorithm is intended to maximize the profits of the firm, while minimizing the execution times of robot arms to perform robotic tasks.

The trade-offs among competing variables (i.e., optimal times and profits obtained with the items manufactured at the assembly line) are assessed by means of the Pareto frontiers. Pareto optimality allows one to deal with conflicting variables in a complex search space.

The economic objective function is calculated according to the following equation:

$$\text{Max } Be = \frac{1}{(1+\rho)^T}\left[\sum_{p=1}^{n} \left(Pr_p - Co_p\right)\cdot Nu_p(t)\right] \tag{22}$$

where

Be: is the economic objective function. It is the business profit to be maximized obtained from a certain product (EUR). It is calculated as the profit of the products manufactured at an assembly line minus full costs.

ρ: is the annual discount rate.

T: period of simulation.

Pr_p: is the market unitary price of the item p (€).

Co_p: is the unitary cost of the item p (€), which covers costs of raw materials, amortization, energy, maintenance, labour force, taxes to direct and indirect costs.

$Nu_p(t)$: is a function which depends on the number of items manufactured per hour:

$$Nu_{\mathrm{p}}(\mathrm{t}) \ = \frac{W}{t(V_k)^{\mu}} \tag{23}$$

where V_k: is the set of tasks required to manufacture an item (p), i.e., the workload, where k represents the number of tasks.

$t(V_k) \ = \sum_{j \epsilon V_k} t_j$: time required to manufacture an item. It is defined as a polynomial function of $t_{\min}$.

μ: parameter to consider the current socio-economic situation and seasonal variations.

W: is the number of working hours per year.

The presented algorithm provides the minimum time $t_{\min p}$ to carry out a certain robot task with the aim to manufacture a product p. Other required tasks are defined as a constant value (t_j). The lower the time the greater the number of items manufactured per hour, thus increasing the firm total profit. Therefore, the cumulative time of all needed tasks is defined as:

$$t(V_k) \ = t_{\min p} + \sum_{j \notin S_{robot}}^{k} t_j \tag{24}$$

5. Experimental Case Studies for Methodology Validation

A set of 33 cases have been solved using the multi-objective optimization problem. The algorithm returns the optimal times that meets the kinematics and dynamics restraints of the industrial robot while avoiding collisions. The minimum times (i.e., the optimal trajectories) represent the desired motion of the robot arm. However, there are uncertainties that lead the actual motion away from the desired motion. These uncertainties range from time-varying loads, parameter variations, errors in mounting the robot arms, mechanical accidents, faulty maintenance, defective use, malfunctioning of actuator forces or torques, overload or excess voltage, noise and disturbances. To reduce these problems a AFSMC controller has been coupled to the industrial robot to come close the actual motion to the desired one. This is achieved by controlling the driving motors to produce the appropriate input forces and torques to track the desired motion, i.e., these inputs are different than the nominal ones to compensate for the above-mentioned uncertainties.

All cases are assessed using a serial chain machine for manufactured industries, the PUMA (Programmable Universal Machine for Assembly) 560 robot with six degrees of freedom with six joints and links. This robot is probably the mathematically best-described robot and, therefore, it is frequently used for research and educational purposes. Therefore, it is an excellent benchmark for testing new algorithms, while facilitating an easy comparison among algorithms for the scientific community. However, it is worthwhile mentioning that the methodology can be applied to any robot arm.

The cases present different initial and final configurations of the robot arm, thus leading to diverse locations of the end-effector. In addition, all cases are defined with the same quantities and considerations regarding the economic issues related to the assembly line and robot tasks. First, the unitary cost to manufacture an item is supposed to be 0.75 EUR (the energy cost is not considered). Second, the product market price is 0.95 EUR. For simplicity, only one item is considered. Third, when the energy cost is considered, the set of cases present disparate costs because the diverse working times. In this sense, the energy cost is defined as 0.0625 EUR/kWh (mean cost). This cost is aggregated to the cost of 0.75 EUR. It is worthwhile to mention that the cost of the consumed energy is negligible regarding the other costs (e.g., workforce or raw materials). Fourth, products are supposed to be manufactured in only one shift of 8 h and considering 365 working days per year. Then, profits are shown in a one-year period.

The time of the other robot tasks required to manufacture the product is supposed to be 85 s, which correspond to the summation of times shown in Equation (24), $\sum\limits_{j \notin S_{robot}}^{k} t_j$.

Table 2 shows the different optimal times (i.e., the desired motion) and physical constraints for the 33 cases reported by the optimization algorithm, which lead to different profits. The physical constraints represent a wide working range for the PUMA 560 industrial robot.

Table 2. Task execution time (s) obtained with the optimization algorithm for the set of cases. These times represent the desired motion and consider the physical restraints of the robot.

Case	Max. Jerk Constraint (rad/s^3)	Consumed Energy Constraint (J)	Execution Time (s) from the Optimization Algorithm	Consumed Energy (J)	Jerk in Third Actuator (rad/s^3)	
					Max.	Min.
1	Unconstrained	Unconstrained	3.79	140.1	1615	−841
2	Unconstrained	75	22.55	75	107	−28
3	5	Unconstrained	19.27	80.1	5	−5
4	5	75	25.76	75	5	−4
5	Unconstrained	Unconstrained	5.14	203.49	1098	−984
6	Unconstrained	200	5.15	200	1096	−980
7	Unconstrained	175	5.3	175	954	−872
8	Unconstrained	150	5.62	150	704	−627
9	Unconstrained	125	6.42	125	425	−418
10	Unconstrained	100	12.25	100	167	−145
11	Unconstrained	95	21.08	95	176	−37
12	5	Unconstrained	23.05	103.3	5	−5
13	5	95	26.35	95	5	−5
14	Unconstrained	Unconstrained	2.27	87.5	897	−944
15	Unconstrained	50	7.34	50	125	−147
16	5	Unconstrained	14.82	51.5	5	−5
17	5	50	17.94	50	2.8	−3.6
18	5	Unconstrained	18.28	42.14	5	−3.9
19	10	Unconstrained	14.51	42.39	10	−7.7
20	25	Unconstrained	10.69	43.1	25	−19.3
21	5	Unconstrained	18.28	42.14	5	−3.9
22	10	Unconstrained	14.51	42.39	10	−7.7
23	25	Unconstrained	10.69	43.1	25	−19.3
24	50	Unconstrained	8.49	43.92	50	−38.7
25	100	Unconstrained	6.74	45.36	100	−77.4
26	1000	Unconstrained	3.21	63.37	856.8	−776.2
27	Unconstrained	Unconstrained	2.41	88.46	1361.50	−1802.90
28	5	40	18.65	40	5	−4
29	Unconstrained	40	9.94	40	337	−13.2
30	Unconstrained	Unconstrained	3.08	106.2	972	−1009
31	Unconstrained	40	9.18	40	42.2	−37.8
32	5	Unconstrained	15.91	40.3	5	−5
33	5	40	15.93	40	5	−5

Note that the robot tasks are performed in a few seconds and, hence, the time scale is presented in seconds. For instance, the case number 14, which presents the minimum execution time and has no restraints in either the jerk or the energy consumed, shows the maximum annual profit (Figure 1). Conversely, case 13 with the maximum execution time and significant physical constraints presents the minimum profits.

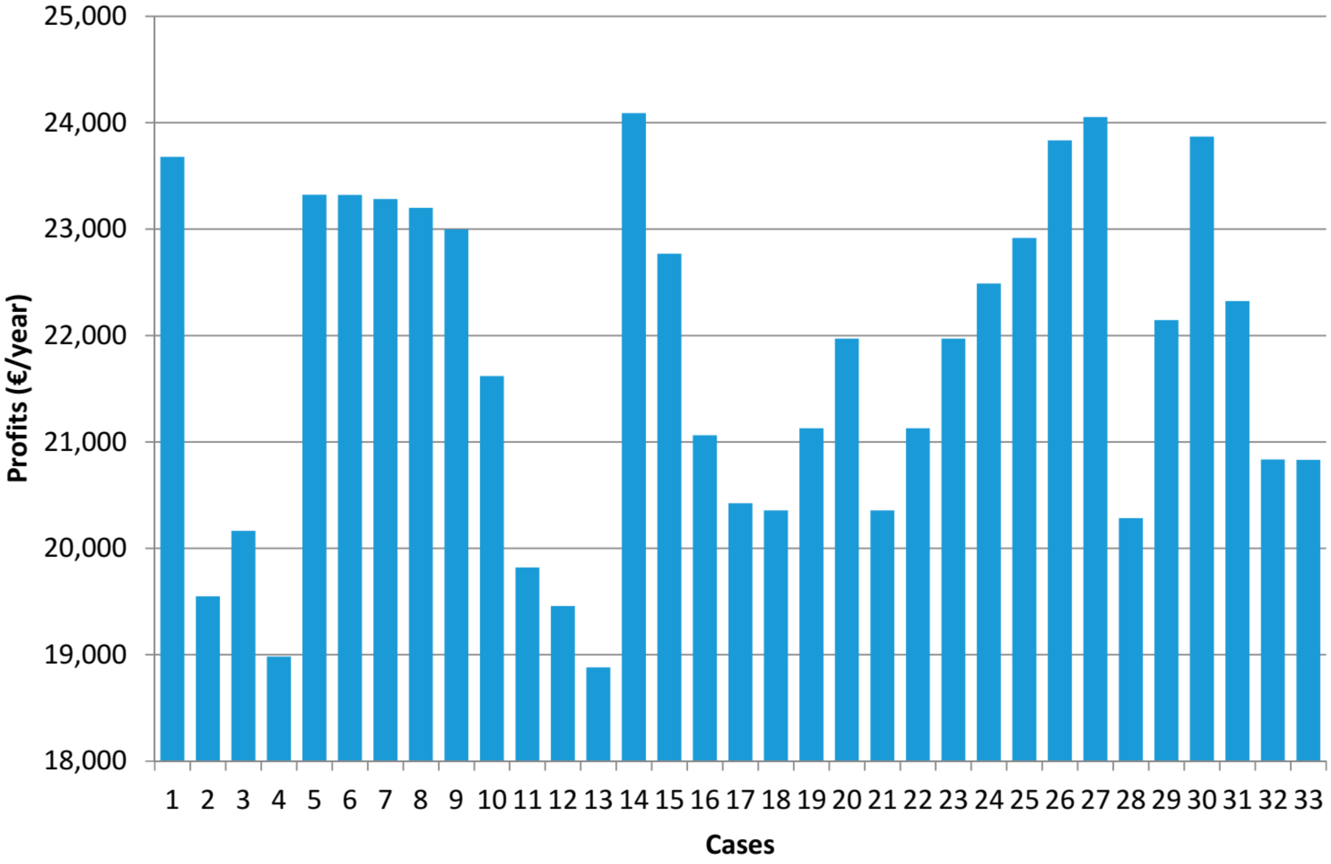

Figure 1. Annual revenue for each case based on the current demand.

The profits for all cases that returns the algorithm are shown in Figure 1. The mean profit for the 33 cases analysed is 21,730.10 EUR/year (with the present demand) and the standard deviation is 1588.6 EUR/year.

If diverse prices, costs and number of products manufactured per year had been used greater discrepancies in profits for the set of cases had been achieved.

An additional analysis is performed using the Pareto approach to show the loss of profits when optimization algorithms are not used. This is illustrated through case 17, in which three different items are assembled in a certain production line. They present different cumulative production times but same working times ($t_{\min p}$ of case 17) of the robot arm. The items also present different total costs, apart from energy costs, prices and parameter μ values (which is intended to reproduce the economic situation and seasonal variations). The total cost of item 1 is supposed to be 0.73 EUR, item 2 is 0.75 EUR and item 3 is 0.78 EUR; the prices are 0.95 EUR for product 1, 1.2 EUR for product 2 and 1.02 EUR for product 3; and the time of other tasks required to manufacture each of the 3 products are 85, 98 and 87 s, respectively. The values for each item of the parameter μ are 0.54, 0.56 and 0.58, respectively. In this work, $t(V_k)$ is defined as a polynomial function of $t_{\min p}$. This function has been calibrated to reflect the market environment, for instance, the current demand of each item.

The loss of profits is due to higher working times that the industrial robot may present (Figure 2). Furthermore, the algorithm provides the Pareto frontiers for 3 different items, which enables to obtain the trade-offs of the conflicting variables (i.e., profits and working times). Therefore, the methodology allows quantifying the opportunity cost of no using the presented methodology. The Pareto frontier solutions have optimal objective function values while meeting all constraints. Moreover, the concept of non-domination sorting can be used to determine which product is better to manufacture, in terms of profits, for a specific execution time of the industrial robot (with a certain maintenance, time-varying loads, parameter variations, disturbances, etc.). Figure 2 shows that the non-dominated subset (i.e., solutions with greater profits) is depicted by the Pareto frontier of the item 3 and times lower than 3.41 s, while for times higher than that it is better to manufacture the item 2 (i.e., the Pareto frontier of item 2 dominates the frontiers of other two items).

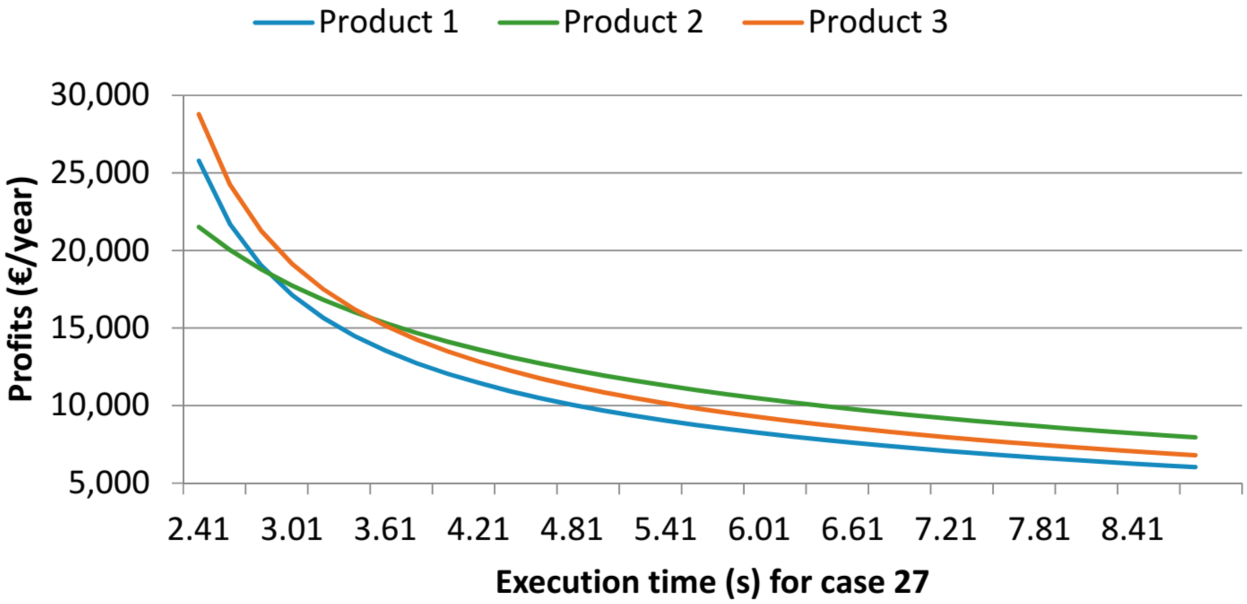

Figure 2. Pareto frontiers obtained with the optimization algorithm for case 27 for the three different products manufactured.

Another exercise has been conducted using a Gaussian function for the cost (C) and the price (P), which are defined by taking into account actual economic fluctuations. This exercise is intended to simulate hypothetical market deviations with regard to costs and prices, with the aim of showing how applying the methodology firms will be able to better adapt to that fluctuations. C $(C = C_p * N_p)$ provides the cost to produce p items (N_p). P_p provides the profit as shown in Equation (22), i.e., the item price multiplied by the number of items manufactured. This exercise has been applied for product 2 and case 17. The statistics of the Gaussian function are defined as follows: the mean is that previous defined for product 2 (i.e., m_c = 0.75 EUR for the cost and m_p = 1.02 EUR for the price) and the standard deviation is $m/4$. The Gaussian functions are sampled and then these values are used in the algorithm. The profits due to fluctuations are shown in Figures 3 and 4. The methodology allows translating market fluctuations directly into profits, which allows making informed decisions about production, for instance, about what items should be manufactured and, moreover, to do it with an efficient scheduling of the robot tasks. In this way, the production is significantly improved.

Another exercise is carried out by simulating a rise in the current demand of a 5%. For the case 6, a total of 122,435 items should now be manufactured per year in shifts of 8 h. The goal is to draw any meaningful conclusions regarding the number of extra working hours to face the rise in the demand. The algorithm reports that an additional 3065 h per year are required to meet the rise in the demand. This result can be used by managers to come up with a better production planning and inventory management, by adjusting the shifts needed in relation with the demand of the products, the possibility of hiring new workers or increasing the stock, the instalment of a new assembly line with robot arms and avoiding robot downtimes and expensive unplanned stoppages for repair and maintenance works, etc. As a result, it helps to devised flexible manufacturing systems to achieve greater profitability and to decide which products should be manufactured and the efficient scheduling to produce them.

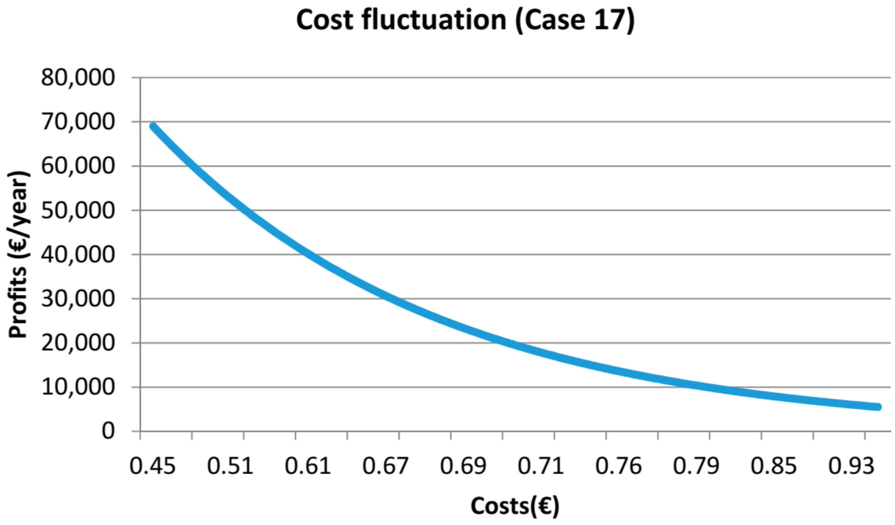

Figure 3. Profits obtained for the product 2 and case 17 versus costs.

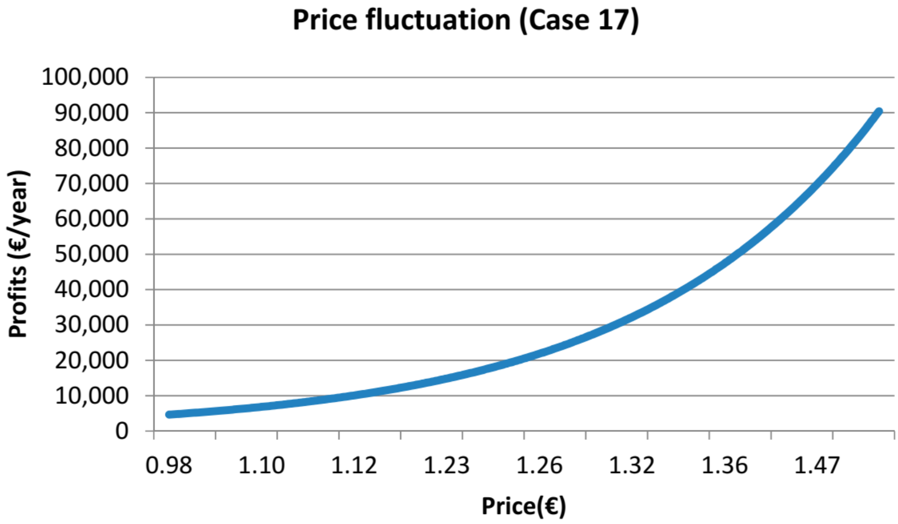

Figure 4. Profits obtained for product 2 and case 17 versus price.

6. Conclusions

This paper is focused on a cost-efficient solution for robotic assembly lines, highlighting an economic analysis to the robotic operation. The methodology allows the company's profit to be quantified by minimizing the time to complete the robot's scheduled tasks and the downtime of the assembly line. This is achieved by coupling an AFSMC controller with a multi-objective optimization algorithm which enables the actual motion to track the desired motion. On the one hand, the multi-objective optimization procedure considers not only the kinematics and the dynamics of assembly line robots, but also the constraint of energy consumed, while avoiding collisions. Furthermore, the trade-offs among the economic parameters of an assembly line and the optimal time trajectory of the desired motion are analyzed using Pareto fronts.

On the other hand, the AFSMC minimizes the discrepancy between actual motions and the desired motion, which are due to the highly nonlinear dynamic parameters of the problem. By considering real-time feedback of the AFSMC, the driving motors are provided with the appropriate input forces and torques to track the desired motion. Bear in mind that the nominal input forces and torques are different from the real ones and the resulting uncertainties should be compensated by AFSMC.

The methodology has been applied to different examples. The results show how this methodology finds the optimal time trajectory that enhances the performance and reliability of assembly lines and, thus, helps companies to stay competitive in a changing economic context. It helps to devised flexible manufacturing systems, better production planning and inventory management to achieve greater profitability. This is achieved by adjusting the shifts needed in relation with the demand of products and avoiding robot downtimes and expensive unplanned stoppages for repair and maintenance works. Therefore, the proposed algorithm can help to decide which products should be manufactured, to plan an efficient scheduling to produce them, to achieve more annual revenue and to better adapt to changes in product costs, prices and demands in the worldwide market. Moreover, the Pareto-optimal sets allows one to assess the trade-offs among the different competing parameters of the multi-objective optimization problem (i.e., working times and profits for each product manufactured at the production line).

The limitations of the presented methodology come from the definition of the path and the dynamic behavior of the robot arm, since some simplifications are used to reduce computing times. These simplifications have led to conservative results for the trajectories obtained, so that lower times to perform the trajectories could be achieved. For instance, the path is decomposed into intervals in which cubic interpolation functions are used. In addition, during the search of collisions, patterned obstacles are used to represent the work environment. A better representation of such simplifications could be used.

In future research, we will delve into the reduction of the energy consumed by the robot system and how it affects to the greenhouse gas emissions, since environmental concerns and strict legislations have been receiving increasing attention in recent years.

Author Contributions: Conceptualization, C.L.-A. and F.R.; methodology, F.V.; software, F.V.; validation, C.L.-A., F.R. and F.V.; formal analysis, C.L.-A. and F.R.; investigation, C.L.-A.; resources, F.V.; data curation, F.V.; writing—original draft preparation, C.L.-A. and F.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Aoyama, T.; Nishi, T.; Zhang, G. Production planning problem with market impact under demand uncertainty. *J. Adv. Mech. Des. Syst. Manuf.* **2017**, *11*, 1–17. [CrossRef]
2. Llopis-Albert, C.; Rubio, F.; Valero, F. Improving productivity using a multi-objective optimization of robotic trajectory planning. *J. Bus. Res.* **2015**, *68*, 1429–1431. [CrossRef]
3. Rubio, F.; Valero, F.; Suñer, J.L.; Cuadrado, J. Optimal time trajectories for industrial robots with torque, power, jerk and energy consumed constraints. *Ind. Robot Int. J.* **2012**, *39*, 92–100. [CrossRef]
4. Llopis-Albert, C.; Rubio, F.; Valero, F. Optimization approaches for robot trajectory planning. *Mul. J. Educ. Soc. Tech. Sci.* **2018**, *5*, 1–16. [CrossRef]
5. Yang, Y.; Pan, J.; Wan, W. Survey of optimal motion planning. *IET Cyber-Syst. Robot.* **2019**, *1*, 13–19. [CrossRef]
6. Gasparetto, A.; Zanotto, V. A technique for time-jerk optimal planning of robot trajectories. *Robot. Comput. Integ. Manuf.* **2008**, *24*, 415–426. [CrossRef]
7. Mohammed, A.; Schmidt, B.; Wang, L.; Gao, L. Minimizing Energy Consumption for Robot Arm Movement. *Procedia CIRP* **2014**, *25*, 400–405. [CrossRef]
8. Van Den Berg, J.; Abbeel, P.; Goldberg, K. LQG-MP: Optimized path planning for robots with motion uncertainty and imperfect state information. *Inter. J. Robot. Res.* **2011**, *30*, 895–913. [CrossRef]
9. Liu, S.; Sun, D.; Zhu, C. Coordinated motion planning for multiple mobile robots along designed paths with formation requirement. *Mechatron. IEEE/ASME Trans.* **2011**, *16*, 1021–1031. [CrossRef]
10. Plaku, E.; Kavraki, E.E.; Vardi, M.Y. Motion planning with dynamics by a synergistic combination of layers of planning. *Robot. IEEE Trans.* **2010**, *26*, 469–482. [CrossRef]

11. Rubio, F.; Llopis-Albert, C.; Valero, F.; Suñer, J.L. Assembly Line Productivity Assessment by Comparing Optimization-Simulation Algorithms of Trajectory Planning for Industrial Robots. *Math. Prob. Eng.* **2015**, *2015*, 931048. [CrossRef]
12. Rubio, F.; Llopis-Albert, C.; Valero, F.; Suñer, J.L. Industrial robot efficient trajectory generation without collision through the evolution of the optimal trajectory. *Rob. Auton. Syst.* **2016**, *86*, 106–112. [CrossRef]
13. Llopis-Albert, C.; Valero, F.; Mata, V.; Pulloquinga, J.L.; Zamora-Ortiz, P.; Escarabajal, R.J. Optimal Reconfiguration of a Parallel Robot for Forward Singularities Avoidance in Rehabilitation Therapies. A Comparison via Different Optimization Methods. *Sustainability* **2020**, *12*, 5803. [CrossRef]
14. Llopis-Albert, C.; Valero, F.; Mata, V.; Escarabajal, R.J.; Zamora-Ortiz, P.; Pulloquinga, J.L. Optimal Reconfiguration of a Limited Parallel Robot for Forward Singularities Avoidance. *Mul. J. Educ. Soc. Tech. Sci.* **2020**, *7*, 113–127. [CrossRef]
15. Yang, J.; Su, H.; Li, Z.; Ao, D.; Song, R. Adaptive Control with a Fuzzy Tuner for Cable-based Rehabilitation Robot. *Int. J. Cont. Autom. Syst.* **2016**, *14*, 865–875. [CrossRef]
16. Zhang, G.; Zhang, X. Concise Adaptive Fuzzy Control of Nonlinearly Parameterized and Periodically Time-varying Systems via Small Gain Theory. *Int. J. Cont. Autom. Syst.* **2016**, *14*, 1–13. [CrossRef]
17. Sutono, S.B.; Abdul-Rashid, S.H.; Aoyama, H.; Taha, Z. Fuzzy-based Taguchi method for multi-response optimization of product form design in Kansei engineering: A case study on car form design. *J. Adv. Mech. Des. Syst. Manuf.* **2016**, *10*, 1–16. [CrossRef]
18. Dubey, A.K. Performance Optimization Control of ECH using Fuzzy Inference Application. *J. Adv. Mech. Des. Syst. Manuf.* **2009**, *3*, 22–34. [CrossRef]
19. Zhang, H.; Fang, H.; Zhang, D.; Luo, X.; Zou, Q. Adaptive Fuzzy Sliding Mode Control for a 3-DOF Parallel Manipulator with Parameters Uncertainties. *Complexity* **2020**, *2020*, 2565316. [CrossRef]
20. Markazi, A.H.D.; Maadani, M.; Zabihifar, S.H.; Doost-Mohammadi, N. Adaptive Fuzzy Sliding Mode Control of Under-actuated Nonlinear Systems. *Int. J. Autom. Comput.* **2018**, *15*, 364–376. [CrossRef]
21. Truong, H.V.A.; Tran, D.T.; To, X.D.; Ahn, K.K.; Jin, M. Adaptive Fuzzy Backstepping Sliding Mode Control for a 3-DOF Hydraulic Manipulator with Nonlinear Disturbance Observer for Large Payload Variation. *Appl. Sci.* **2019**, *9*, 3290. [CrossRef]
22. Piltan, F.; Rashidian, M.R.; Shamsodini, M.; Allahdadi, S. Effect of Rule Base on the Fuzzy-Based Tuning Fuzzy Sliding Mode Controller: Applied to 2nd Order Nonlinear System. *Int. J. Adv. Sci. Tech.* **2012**, *46*, 39–70.
23. El Fadil, H.; Giri, F. Reducing chattering phenomenon in sliding mode control of Buck-Boost power converters. In Proceedings of the 2008 IEEE International Symposium on Industrial Electronics, Cambridge, UK, 30 June–2 July 2008; pp. 287–292.
24. Li, T.-H.S.; Huang, Y.C. MIMO adaptive fuzzy terminal sliding-mode controller for robotic manipulators. *Inf. Sci.* **2010**, *180*, 4641–4660. [CrossRef]
25. Wang, L.X. *A Course in Fuzzy Systems and Control, Upper Saddle River*; Prentice Hall: Hoboken, NJ, USA, 1997.