Newly Elaborated Hybrid Algorithm for Optimization of Robot Arm’s Trajectory in Order to Increase Efficiency and Provide Sustainability in Production

Rabab Benotsmane 1, László Dudás 1, and György Kovács 2,*

1 Department of Information Technology, University of Miskolc, 3515 Miskolc, Hungary; itraban@uni-miskolc.hu (R.B.); itidl@uni-miskolc.hu (L.D.)
2 Institute of Logistics, University of Miskolc, 3515 Miskolc, Hungary
* Correspondence: altkovac@uni-miskolc.hu

Abstract: Nowadays, resources for production (raw materials, human, energy, etc.) are limited, while population, consumption and environmental damage are continuously increasing. Consequently, the current practices of resource usage are not sustainable. Therefore, manufacturing companies have to change to environmentally friendly and innovative technologies and tools, e.g., industrial robots. Robots are necessary in the production sector and also in terms of sustainability: (1) the application of robots is needed to compensate for the lack of human resources; (2) robots can increase productivity significantly; and (3) there are several hazardous (e.g., chemical, physical) industrial tasks for which robots are more adequate than human workforce. This article introduces a newly elaborated Hybrid Algorithm for optimization of a robot arm’s trajectory by the selection of that trajectory that has the smallest cycle time. This Hybrid Algorithm is based on the Tabu Search Algorithm and also uses two added methods—Point Insertion and Grid Refinement—simultaneously to find more precisely the optimal motion path of the robot arm in order to further reduce the cycle time and utilize the joints’ torque more efficiently. This Hybrid Algorithm is even more effective than applying the Tabu Search method alone and results in even higher efficiency improvement. The Hybrid Algorithm is executed using MATLAB software by creating a dynamic model of a 5 degree-of-freedom robot arm. The main contribution of the research is the elaboration of the new Hybrid Algorithm, which results in the minimization of robot arms’ motion cycle times, causing a significant increase in productivity and thus a reduction in specific production cost; furthermore, obstacles in the workspace can be avoided. The efficiency of the Hybrid Algorithm is validated by a case study showing that application of the new algorithm resulted in 32% shorter motion cycle time.

Keywords: industrial robot arm; trajectory optimization; new Hybrid Algorithm; Tabu Search Algorithm; cycle time minimization; increase productivity; sustainability

1. Introduction

Factors such as global market competition, fast changing and more unique customer demands, and increasing environmental damage have caused new challenges and requirements in the manufacturing sector. Due to these global tendencies, the push production philosophy has been replaced by the more efficient and unique pull philosophy. Nowadays, the elements of the Industry 4.0 concept are being implemented by production companies in order to provide both profitable and sustainable operation [1,2].

Recently, the resources (raw materials, human, equipment, energy, etc.) are limited in the production sector. On the contrary, population, consumption and environmental damage are continuously increasing. Therefore, the practice of resource usage is not sustainable. Consequently, manufacturing companies need to apply environmentally friendly, energy-efficient and innovative equipment and technologies, e.g., elements of
the Industry 4.0 concept—such as industrial robots—in order to increase productivity and profitability, and at the same time provide sustainability in production [3,4].

Most of the relevant publications discuss sustainability based on three main aspects: environmental, social and economic. Environmental sustainability means the compliance with environmental regulations, a higher volume of usage of renewable resources and raw materials, more efficient utilization of resources, application of environmentally friendly technologies and equipment, reuse and recycling of wastes, etc. [5,6].

The social aspect means the operation of companies is according to compliance with labor laws, human rights and safety-at-work rules, along with the establishment of workplaces that provide healthy and safe conditions for the workers [7,8].

The economic aspect means the fulfillment of the target economic performance of enterprises, in compliance with the requirements of environmental and social sustainability simultaneously [9–11].

The significance of the research topic is that the application of industrial robots is necessary in the production sector also in terms of sustainability. There are three main ways that industrial robots contribute to sustainability which are the following:

1. The application of industrial robots is needed in production companies in order to compensate for the lack of human resources (economic and social aspects of sustainability).

2. Industrial robots—as environmentally friendly, energy-efficient and innovative technologies—have the capability to significantly increase productivity, which results in reductions in specific production costs, including labor costs, energy costs, etc. The application of energy-efficient robots causes higher utilization of resources, energy-saving and less environmental damage (environmental and economic aspects of sustainability).

3. There are several industrial tasks for which industrial robots are more suitable than human workers, because the conditions of some workplaces can be hazardous for workers (e.g., chemical, ergonomic or physical hazards) and the safety of all workers has become a priority (social and economic aspects of sustainability) [12,13].

Efficient and sustainable production means that the manufactured final products have to be competitive, cost-effective, energy-efficient and material-efficient, and produced by innovative, energy-saving and environmentally friendly technologies [14–17] in safe and healthy workplaces; furthermore, the reuse and recycling of wastes must be provided for.

In the existing literature, there are a lot of publications that discuss the Industry 4.0 concept as a new paradigm in the industrial sector. The main elements of this concept are industrial robots because the robot arms are quite possibly the most valuable tools due to their efficiency and flexibility. Different algorithms aim to optimize the robot arms’ trajectory, taking into consideration the main industrial aims, i.e., shorter lead time, higher productivity, higher quality of final products, sustainable production, safer workplaces, etc. [18–20].

The main task of the trajectory planning is how the industrial robot can find an optimal or quasi-optimal trajectory among many possibilities, because the optimal trajectory provides the highest productivity [21]. Several articles deal with reducing cycle time and increasing velocity [22,23]; other papers focus on smoothness and maintaining the stability of the system during the task execution [24,25]. It can be concluded, based on the recent literature, that the most often used objective functions applied to optimize the trajectory of industrial robots are the minimization of cycle time, minimization of energy, minimization of jerk, etc., but sometimes the optimization can be multi-objective [26].

There are a lot of different types of algorithms used for path planning of robot arms [24,27]. For example, exact and adaptive roadmap algorithms aim to define the robot search space. Cell decomposition algorithm subdivides the workspace into small cells [28–30]. Tree Search algorithm can be applied to re-order the task planning as pick and place, taking into consideration the cost of completeness and high-resolution discretization [31]. Artificial potential algorithms aim to create optimal path by avoiding obstacles [32]. Several meth-
ods are available in the existing literature for optimization of robot arms’ path planning, which are the following: e.g., Fuzzy Logic control [33], Neural Networks [34], Genetic Algorithms [35], Ant Colony Optimization [36] and Simulated Annealing [37] methods. Furthermore, there are also several searching algorithms applied to find the optimal path, i.e., Dijkstra [38], A* [39] and Tabu Search (TS) algorithms. The most commonly used searching algorithm is the Tabu Search optimization algorithm [40], which is introduced in detail in Section 2.1.

Based on the synthesis of the existing literature, it can be concluded that a lot of optimization algorithms are available for the trajectory optimization. However, there cannot be found an algorithm that is more effective than the Tabu Search Algorithm—which was applied earlier by the authors—to determine more precisely the optimal motion path of the robot arm in order to further reduce the cycle time and utilize the joints’ torque more efficiently. This kind of algorithm is very important in practice, because the minimization of a robot arm’s motion cycle time results in significant increase in productivity and thus a reduction in specific production costs.

On the other hand, based on the recent literature, it can be also concluded that the motion path planning and the trajectory optimization are discussed separately in the publications. Therefore, publications present different methods and algorithms both for determination of the optimal motion path and optimization of the robot arm trajectory [27]. Consequently, there cannot be found any algorithm in the literature that solves the two main tasks—the trajectory optimization and the path planning tasks—simultaneously.

Therefore, the motivation of our research was to elaborate a new metaheuristic algorithm, which can fulfill the above mentioned requirements at the same time. Thus, the primary aim of the research was to elaborate a new Hybrid Algorithm which can determine that path with the globally minimal cycle time between the starting and ending points, avoiding given obstacles. The other aim of the research was that the elaborated Hybrid Algorithm can solve both the trajectory optimization task and the path planning task in an integrated form simultaneously in an acceptable calculation time.

The newly elaborated Hybrid Algorithm is based on the Tabu Search Algorithm and also using two added methods—Point Insertion and Grid Refinement methods—simultaneously to determine the optimal motion path of the robot arm more precisely. The Hybrid Algorithm selects that trajectory from the space of possible trajectories with the smallest cycle time, providing a quasi-optimal global solution. Thus, the essence of the new Hybrid Algorithm is to find the globally optimal path of a robot arm end effector, starting from a simple trajectory between two given points constructed using cubic spline. The path is interpolated from the starting point to the ending point, passing by some inner points. From this curve, the Hybrid Algorithm converges and finds the globally optimal curve by refining the trajectory. Thus, the algorithm searches for the best trajectory in the space of trajectories in order to identify the one with the globally minimal cycle time.

The optimization of the robot arm’s trajectory by the application of the new Hybrid Algorithm results in an even further reduction in cycle time and more efficient utilization of joint torque compared to the individual application of the Tabu Search method. Furthermore, the Hybrid Algorithm also provides that obstacles can be avoided in the workspace during the refinement of robot arm trajectories. Therefore, this new algorithm is even more effective than applying the Tabu Search Algorithm alone and results in even higher improvement of efficiency.

In Section 2 of this article, we introduce the methodology of our newly elaborated Hybrid Algorithm, describing the main features of this combined method. Then, the essence and the procedure of the new algorithm are presented in 2D in order to better understand the process. In Section 3, we present a case study on optimization of robot arm trajectory by the application of the Hybrid Algorithm, which is executed using MATLAB software by creating a dynamic model of an RV-2AJ Mitsubishi robot arm which has 5 degrees of freedom (5R). A MATLAB-Simulink function can determine the torque for every joint while the robot arm effector runs the given trajectory.
The efficiency of the new algorithm was validated by a case study in which a 32% shorter motion cycle time resulted due to the application of the Hybrid Algorithm.

The main contribution of the research is the elaboration of a new Hybrid Algorithm which can define the quasi-optimal path that has the lowest cycle time between two given points, avoiding obstacles. The new algorithm results in an even further reduction in cycle time and more efficient utilization of joint torque compared to the individual application of the Tabu Search method by minimization of robot arm motion cycle times and optimal utilization of the joints’ torque. Furthermore, the Hybrid Algorithm also solves the two main tasks—both the trajectory optimization and the path planning tasks—in an integrated form, simultaneously.

The minimization of motion cycle time causes a significant increase in productivity, which means that more final products can be manufactured in a given time interval. It results in the reduction of specific production costs and thereby the reduction of the final products’ cost. It can be summarized that the efficiency improvement results in cost saving for the enterprises. The companies can spend this cost saving on further development and invest in environmentally friendly, energy-efficient technologies and equipment which can further improve the sustainability in the production.

2. Materials and Methods

One of the main pillars of the Industry 4.0 concept is industrial robots due to their high efficiency and precision. The most important task in robotics is the trajectory planning of robot arms, because the optimal trajectory provides the highest productivity of the industrial robots. There are different terminologies discussed in the literature related to motion planning for industrial robots [41,42]. The motion planning research field can be grouped into two main concepts: path planning and optimal trajectory planning. Path planning refers to a series of defined positions that are created for execution by the robot end effector, considering the avoidance of obstacles in the workspace (e.g., machine, workstation, etc.) [22,43]. Trajectory planning refers to the result of path planning execution where further parameters of the motion must be defined, i.e., position, velocity, acceleration and sometimes the jerk variable, etc. [24,27].

The trajectory optimization determines that joint torque functions for every joint that—applying them parallel—results in the lowest motion time for executing the motion along the given path. In the trajectory optimization process the joint torque limits are the usual constraints that can be taken into account in the solution of this dynamic problem. The optimized torque control for a given path will result in minimal cycle time, but the energy minimum and the minimal jerk is not guaranteed. Generally, it is possible, that a more suitable motion path is available between the starting and ending points, avoiding given obstacles. To find this optimal path means to determine the best path among those that have the lowest cycle time for their given path due to the optimized dynamic conditions.

It is well known that the motion control of industrial robots in the industry is traditional, where the controller is a black box that uses the necessary input data and provides output data for the robot in order to execute the adequate trajectory. Typically, researchers plan a trajectory that combines multiple points for the industrial robot arm using different splines, e.g., B spline, cubic spline, trigonometric spline or Bezier spline [44]. The constructed trajectory can be executed in joint space known as free space which presents the limited space for executing motion by angular displacements of each joint of a robotic link. The following conditions should be taken into consideration related to the workspace of the robots: the singularity of the robot, the velocity and the torque of the robot arms, length of the robot arm, position and orientation of the end effector, etc. [45]. The conditions for task space are mostly based on the avoidance of obstacles and the precision required [46].

The most important task of the trajectory planning is how the industrial robot can define an optimal or quasi-optimal trajectory among many alternatives [21,27].
The newly elaborated Hybrid Algorithm simultaneously combines the Tabu Search Algorithm, the Point Insertion and Grid Refinement methods, primarily in order to determine the optimal path between two given points that has the lowest cycle time, and ensure that the obstacles can be avoided.

The conception of the new Hybrid Algorithm is to find the globally optimal path of a robot arm end effector starting from a simple trajectory between two given points constructed using cubic spline. The path is interpolated from the starting point to the ending point passing by some inner points. From this curve, the algorithm converges and finds the globally optimal curve by refining the trajectory in order to minimize the motion cycle time. For this, the algorithm generates different candidate splines as possible trajectories and determines the lowest cycle time for every mentioned trajectory with the whip-lashing algorithm [47]. During the motion path optimization, in searching for the globally optimal trajectory, the objective function is also the minimal cycle time. The Hybrid Algorithm searches for the best trajectory in the space of trajectories—where the trajectory is the lowest—using the cycle time of every candidate trajectory, trying to identify the one with the globally minimal cycle time. The algorithm also ensures that the obstacles can be avoided in the workspace during the refinement of robot trajectories.

The other advantage of the Hybrid Algorithm is that it can solve the path planning and trajectory optimization in an integrated form simultaneously, which is a very complex dual-purpose task. This duality was emphasized also by Balkan, T. [48]: “Optimal trajectory planning problem can be divided into two sub-problems. Namely planning a geometric path that accomplishes the desired task while avoiding collisions with obstacles. Then tracking this geometric path optimally by finding optimal velocity and associated torque/force profiles along this path. So an optimal control problem can be treated as an optimal trajectory planning problem, if a prescribed path is to be followed”.

Thus, the new Hybrid Algorithm solves both the global path planning geometrical task, determining the best spline path, and the dynamic task of the trajectory optimization, using maximal allowed joint torques at the same time.

It can be concluded that Hybrid Algorithm has to apply to provide the global quasi-optimal solution in an acceptable calculation time. The new algorithm integrates, on the one hand, the search power of the Tabu Search Algorithm for selection of the optimal trajectory from the space of possible trajectories with the smallest cycle time. On the other hand, the algorithm utilizes the successive approximation of the Point Insertion and Grid Refinement algorithmic technologies to determine the optimal motion path of the robot arm more precisely.

2.1. Tabu Search Algorithm Overview

The Tabu Search Algorithm was introduced for the first time by Fred Glover in 1986 [49], with the aim to solve combinatorial optimization and local optimal search problems and provide global optimum. It is a meta-heuristic that directs a local heuristic search method to explore the solution space beyond local optimality [50]. The Tabu Search Algorithm is an adaptive strategy with the ability to include many other methods, thus avoiding the limitations of local optimality. The algorithm is characterized by short- and long-term memories that help to create flexible search behavior and allow the solution space to be searched economically and effectively. The basic idea is inspired by research techniques used in Artificial Intelligence and consists of keeping track of the past history of the research process in one or more memories named a tabu list and using this information to guide its future progress and avoid getting stuck in local optima (Figure 1).
3. Results—Hybrid Algorithm for Optimization of Robot Arm’s Trajectory

The Hybrid Algorithm searches for the best trajectory in the space of trajectories using the cycle time of every trajectory, trying to identify the one with the minimal cycle time. The goal of our newly elaborated Hybrid Algorithm is more precisely to find the quasi-optimal trajectory of the robot end effector motion between two given points S and E. The optimality criterion is the minimal cycle time; that is, the traveling time on this quasi-optimal trajectory. The Hybrid Algorithm generates different candidate splines as possible trajectories and determines the cycle time for every mentioned trajectory, applying a previously elaborated cycle time minimization algorithm $f = f(spline)$. This iterative algorithm has been introduced in a detailed manner in connection with the whip-lashing theory [47] and can determine the lowest cycle time for traveling a given trajectory utilizing the possible maximal torques of the robot arm joints, taking into account the different torque limits for every joint. The core of this iterative minimization method is a MATLAB-Simulink function [51], which can determine the $torque = torque(time)$ function for every joint while the robot arm effector runs the given trajectory. In this paper, this fairly complex and time consuming iterative algorithm is referred to as $f = f(spline)$ or simply $f()$, because here, only its result, the lowest cycle time of running the spline trajectory, is necessary for the Hybrid Algorithm.

The hybrid characteristics of the Hybrid Algorithm originate from the applied three methods of generation of the mentioned candidate splines: the Tabu Search (TS), the Point Insertion (PI) and the Grid Refinement (GR). The Hybrid Algorithm is based on the Tabu Search Algorithm, but for providing greater precision in limited calculation time, it simultaneously uses further two methods—Point Insertion and Grid Refinement—to find the quasi-optimal motion path of the robot arm.

To find that trajectory where the optimality criterion is the shortest cycle time is not easy among the infinity possible variants of trajectories between the given starting (S) and
ending (E) points. The Hybrid Algorithm applies the combinatorial optimization method, one sort of metaheuristics using discretization of the search space. It means that instead of the continuous 3D space, it uses a 3D grid matrix of the space points and the splines of the trajectories fit on the selected points of this grid. If the grid is fine enough, meaning the grid steps are small enough, the found quasi-optimal trajectory spline can approximate the optimal solution of continuous 3D space with a practically acceptable small difference, with a negligible larger cycle time. Using such a fine grid and the Tabu Search Algorithm for finding this quasi-optimal trajectory is theoretically possible, but in reality, because of the combinatorial nature of the task, the needed time for the solution is huge and practically unacceptable. Therefore, the application of the Tabu Search Algorithm alone is not suitable.

To resolve the aforementioned problem, a dynamically refined grid is applied, which starts with a large enough grid step and the grid is refined as the process progresses. Another parameter of the mentioned trajectory spline based on Tabu Search Algorithm is the number of inner grid points that holds the interpolating spline. Because the amount of the neighbor splines of the actual spline in the optimization process depends exponentially on the number of internal spline grid points, the limitation of this number is crucial to obtain a practically applicable heuristic algorithm that can provide a good quasi-optimal solution in acceptable running time. Considering these two very hard limitation parameters—grid resolution and number of inner spline points—a successive iteration method that tunes itself depending on the progress of the search process is formulated. As the number of inner spline points has an exponential effect on the running time of calculating the locally best neighbor trajectory, this parameter was chosen to increase less often and used as the terminating parameter. In this paper, the word “spline” will be used instead of “trajectory” or “trajectory spline” because of the geometric nature of the algorithm.

The essence of the Hybrid Algorithm and its main characteristics that it is a Tabu Search-based optimization method in a 3D grid with interpolated point (waypoint) insertion, plus grid step halving. The algorithm starts with a spline beginning from the start point S and finishing at the endpoint E, fitted on a few inner interpolated points located on grid points of a 3D grid with a relatively large grid step. Then, a double embedded cycle structure starts—the global function means a cycle with the cycle core between brackets—and it includes a cycle, where TS means Tabu Search and PI abbreviates Point Insertion, which is when a new interpolated point is inserted, halving a selected segment of the spline. GR means Grid Refinement.

The c (TS → PI) cycle finishes when the last PI is unable to find a better spline. In such case, the last point insertion is cancelled. Then, GR follows, which halves the grid step. This halving finishes the core of the outer cycle. The outer cycle repeats its core until the process stops. Several terminating conditions can be imagined, e.g., the upper limit of the number of interpolated points.

The requirement that the splines must fit on grid points was our strategy for limiting the search space and representing it in a simple manner; the use of a grid also allows us to precisely optimize the trajectory by using the easy grid refinement process.

3.1. Tabu Search Based Hybrid Algorithm

To avoid combinatorial explosion of the number of splines in the searched space, the Hybrid Algorithm uses Point Insertion (PI) and Grid Refinement (GR) processes in addition to Tabu Search (TS), as shown in Figure 2, instead of using TS alone. The figure presents the main Hybrid Algorithm flow chart for refining the trajectory. The main flow chart includes two sub-procedures; the first one relates to the Tabu Search method while the second is the Point Insertion. The GR for grid step halving is presented as a simple algorithmic envelope that halves the grid step value. The flowchart mirrors the previously mentioned nested structure of two cycles, the outer GR cycle and the inner (TS → PI) cycle: c (c (TS → PI) → GR).

3 of 23
3.1.1. Tabu Search Algorithm in Details

The control points of splines are \( P_i(X_i, Y_i, Z_i) \) general grid points given by \( X_i, Y_i, Z_i \) coordinates in the 3D volume of the robot arm working area \( V \). Some regions of this 3D space can be excluded to handle obstacles that must be avoided by the moved object. Such restrictions may also be handled by the suggested optimization method but need additional algorithmic steps not shown here.

TS uses six neighbor grid points—\( G_{i,1}(X_i - gs, Y_i, Z_i), G_{i,2}(X_i - gs, Y_i, Z_i), G_{i,3}(X_i, Y_i - gs, Z_i), G_{i,4}(X_i, Y_i + gs, Z_i), G_{i,5}(X_i, Y_i, Z_i - gs), G_{i,6}(X_i, Y_i, Z_i + gs) \)—of every inner interpolated \( P_i \) spline-point, where \( gs \) is the actual grid step, as Figure 3 shows.
Figure 3. Neighbor spline generation through the neighbor grid points of the inner grid points of the actual spline.

The search is executed in the space of non-tabu splines that are fitted on the $7^{N-2}$ variants of the neighboring grid points and the $(N-2)$ inner grid points of the previous spline. The TS cycle continues as long as the stopping criterion allows. When the actual TS finishes, it has found a trajectory between S and E points having quasi-optimal $ct$ cycle time. However, this trajectory is far from the global optimum, very likely because of the large $gs$ grid step and small number $N$ of holder points. To provide a better approximation of the optimal trajectory, incrementing the $N$ number of holder points or halving the $gs$ grid step is necessary before restarting TS from the last quasi-optimal trajectory. The repetition of TS results in better approximation of the global optima in most cases.

The flowchart of TS subroutine is shown in Figure 4. The flowchart visualizes the working steps. The actual state (trajectory) ($n$), the best state among non-tabu neighbor states ($n_{\text{LocalBest}}$), and the best state found so far ($n_{\text{GlobalBest}}$) are records, storing the spline and the cycle time of the corresponding trajectory.

1. Let the initial state be $n$ (storing a starting spline, whose $ct$ is a large value, e.g., $1E20$); let the best state found so far be $n_{\text{GlobalBest}}$, initialized by the value of $n$; let the $\text{TabuList}$ be initialized with an empty list, so $\text{TabuList}$ length $tll$ is equal to 0; let the max value of $tll$ be equal $tll_{\text{max}} = 80$ (this max value can be modified based on running experience); let the counter of false comparisons be $cfc = 0$, max value $cfc_{\text{max}} = 80$ (this max value can also be modified based on running experience); let the cycle variable $k$ be initialized to 1; and let the maximum value of cycle variable $k_{\text{max}}$ be 10,000.

2. If the stop criterion is met, return to the $n_{\text{GlobalBest}}$ state, and stop.
3. Otherwise, let $n_{\text{LocalBest}}$ be equal to the state having the minimal cycle time $ct_{\text{min}}$ among the $f() = ct$ values of neighbor states of $n$, excluding the states of the TabuList.

4. Update the TabuList with $n$ state and let $n$ new value be equal to $n_{\text{LocalBest}}$.

5. If $(n_{\text{LocalBest}}.ct < n_{\text{GlobalBest}}.ct)$, then $n_{\text{GlobalBest}} = n_{\text{LocalBest}}$ and $cfc = 0$; otherwise, let the counter of false comparisons be $cfc = cfc + 1$.

6. Increment the cycle variable $k$ and repeat from step 2.

Some remarks on the algorithm:

- A state is a record (structure) containing:
  - the spline from $S$ to $E$ through the actual internal control grid points $P_1, \ldots, P_i, \ldots, P_N$ represented by $N$ interpolated grid points (control points) in vector $Vspline$; and
  - the cycle time $ct = f(spline)$ of traveling a trajectory—represented by the spline—is determined using the same $f()$ iterative algorithm that was used to determine the minimal cycle time in whip-lashing theory [47].

- The TabuList is a vector of those tabu states that have been traversed in the search process in the last $tll$ searching steps, where $tll$ is the TabuList length. The state $n_{\text{LocalBest}}$ among the non-tabu local neighbor states whose cycle time was the minimal $ct_{\text{min}}$ is inserted in the beginning of the TabuList in the TabuList actualization step. The TabuList length is checked after insertion and if it is longer then $tll_{\text{max}}$, it is cut back to $tll_{\text{max}}$.

- The stop criterion includes three conditions:
  - $k > k_{\text{max}}$; OR
  - all neighbor states are included in the TabuList (no state to step in); OR
  - counter of the number of $(n_{\text{GlobalBest}}.ct > n_{\text{LocalBest}}.ct) = \text{false comparisons number } cfc$ reaches $cfc_{\text{max}}$. (The local bests did not give a new global best in many checks.)

- The checking of the tabu condition can be simplified by handling the spline content of the used states only; the $ct$ cycle time can be ignored.

- For the real robot, not every grid point will be used (e.g., at the border of the motion area or near obstacles), so not every neighbor state can be generated. (In the first version of the program, we supposed that the progress of the algorithm would not touch such non-existing splines.)
3.1.2. Point Insertion Method in Details

The second part of the inner ($TS \rightarrow PI$) cycle inserts a new interpolated point. The new point is generated as one of the halving points of the $(N - 1)$ spline segments between spline interpolated points. Instead of such a halving spline point, the closest grid point is
taken. That grid point is selected from the \((N-1)\) candidates, which results in the best spline, i.e., the spline having the smallest cycle time. Figure 5 shows the PI subroutine.

![Flow chart of the Point Insertion subroutine.](image)

**Figure 5.** Flow chart of the Point Insertion subroutine.

### 3.1.3. Grid Refinement Method in Details

The last step of the Hybrid Algorithm outer cycle core in Figure 2 halves the grid step \(gs\). As \(gs\) is one of the parameters of TS and PI, the refined value will take effect in the continuation of the search process.
The Hybrid Algorithm tries to solve the task with the minimum interpolated points and with the largest grid step. The termination condition is $N$ maximum value, because $N$ has an exponential effect on running time.

The goal of using Point Insertion is to produce more fitting splines, but the use of only a few points in the beginning speeds up the search and prevents combinatorial explosion of the calculation time. The goal of using Grid Refinement is similar, and the use of a grid makes calculations easy.

3.2. Demonstration of Tabu Search Based Hybrid Optimization Algorithm in 2D

This section will demonstrate the working of the algorithm in two dimensions to make it easier to follow the working steps and their effect. The working of the $c (TS \rightarrow PI) \rightarrow GR$ algorithm shown in Figure 6 supposes that the searched optimal trajectory’s spline is the one marked with wide red line. The starting spline, marked 1, has $N = 3$ interpolated points, and thus one inner point.

Figure 6. Tabu Search-based Hybrid Optimization Algorithm demonstration steps in 2D.

Then, the Hybrid Algorithm starts. The blue splines numbered 2, 3 and 4 show the progress of the first running of TS algorithm (simplified).

As there is no better spline fitting on neighboring grid points, the PI point insertion follows. The 5-5 spline marked with brown is the result of the PI point insertion. Then, the TS begins again starting from the 5-5 spline and progressing through the 6-6 and 7-7 brown splines.

Then, the TS terminates because there is no better spline with two inner points. The PI point insertion tries to insert a third inner point on the grid, but these trials do not result in a better spline. Thus, the $c (TS \rightarrow PI)$ cycle finishes and the GR grid refinement halves the grid step. Then, the $c (TS \rightarrow PI)$ cycle starts again using the 7-7 spline as the starting spline and the TS runs. It finds the 8-8 green spline and terminates because there is no better spline fitting on the neighboring points. The PI point insertion tries to insert a point on the refined grid, but the terminating limit of $N < 5$ does not allow $N$ to be incremented to 5.

Based on the synthesis of the existing literature, the authors discovered two gaps during the literature review of the research topic.
1. A lot of optimization methods and algorithms are available in the literature for trajectory optimization [27–40,42,49,50], but there cannot be found any algorithm which combines the Tabu Search Algorithm, the Point Insertion and the Grid Refinement methods simultaneously. The newly elaborated Hybrid Algorithm integrates, on the one hand, the search power of Tabu Search Algorithm [40,49,50] for the selection of the optimal trajectory from the space of possible trajectories that has the smallest cycle time. On the other hand, the Hybrid Algorithm utilizes the successive approximation of the Point Insertion and Grid Refinement algorithmic technologies to determine the optimal motion path of the robot arm more precisely [42].

Therefore, the primary aim of our research was to elaborate the Hybrid Algorithm to determine more precisely the optimal motion path of the robot arm in order to further reduce the cycle time and utilize the joints’ torque more efficiently avoiding given obstacles. The significance of the new algorithm is the minimization of robot arms’ motion cycle time, resulting in a significant increase in productivity. The newly elaborated Hybrid Algorithm is even more effective than applying Tabu Search method alone and results in even higher efficiency improvement. The efficiency of the Hybrid Algorithm was validated by a case study.

2. Based on the synthesis of the existing literature, it can also be concluded that in the literature, the motion path planning and the trajectory optimization are discussed separately [27,48,52]. Different methods and algorithms are introduced in publications for the determination of the optimal motion path and the optimization of the robot arm trajectory. Although a lot of optimization algorithms are available [28–40], there cannot be found any algorithm which solves the two main tasks—both the trajectory optimization and the path planning tasks—in an integrated form simultaneously.

Therefore, the other aim of our research was for the elaborated new Hybrid Algorithm to solve both the trajectory optimization task and the path planning task in an integrated form at the same time.

Consequently, the newly elaborated Hybrid Algorithm is an absolutely novel optimization algorithm and fills the two above-mentioned gaps in the literature.

4. Discussion—Application of the Hybrid Algorithm for Optimization of Robot Arms’ Trajectory, Case Study

In order to test the elaborated Hybrid Algorithm, geometric and dynamic parameters of a Mitsubishi RV-2AJ robot arm are applied. MATLAB Simulink is used for the program and the necessary dynamic simulation. We imported the mechanical structure—sizes of links, number of joints, rotation axes, etc.—of an RV-2AJ robot arm, and then we configured its inverse kinematic by writing a short program in MATLAB script. The Hybrid Algorithm uses the f(spline) cycle time determination algorithm that can provide the least time for travelling a given trajectory, utilizing as much joint torques as possible. That algorithm starts from a large cycle time value and decreases it in each iteration as long as the maximum torque for robot joints does not exceed the allowable torque value. The algorithm stops when the torque of one of the joints exceeds the allowable torque according to the given parameters, and the value of cycle time for the preceding step is stored as the least cycle time $ct$ [47].

4.1. Parameters of the Optimization Task

The achievement of the algorithm is based on different inputs as presented in the following:

- The RV-2AJ robot arm’s kinematic and dynamic properties.
  - The robot arm working space $V$, so the limiting surfaces of those points that can be reached by the gripper of the robot arm;
  - Vector of maximal allowed torques $aT$ for every joint $J$ of the robot arm.
- The object $O$ to be conveyed.
  - The mass $m$ of the $O$ conveyed object;
- The vector $v$ that points from the gripper coordinate-system origin to the mass center of the conveyed object.

- Starting parameters.
  - The searched trajectory at the initial moment:
    1. The number $N$ of the trajectory interpolated points including $S$ and $E$ ($N > 2$);
    2. $(X_i, Y_i, Z_i)$ coordinates of the trajectory interpolated points in a $P(X_i, Y_i, Z_i)$ matrix ($i = 1 \ldots N$), where the coordinates of the starting point $S = P(X_S, Y_S, Z_S)$, ending point $E = P(X_E, Y_E, Z_E)$ and trajectory inner points ($i = 2 \ldots N - 1$).

- Starting grid step $g_s$.

- Stopping parameters.
  - Stopping criteria of Tabu Search;
  - Stopping criteria of the points number incrementation cycle.

4.2. Hybrid Algorithm Realization

The algorithm realization and execution were carried out in MATLAB software by entering the necessary input arguments for each block from the organigram and writing an optimal code script. Figure 7 describes the results of the Hybrid Algorithm execution for all iterations, starting from the first trajectory until the final one. To observe clearly the trajectories obtained, the result is presented in different views, where the first trajectory is presented in purple and the final trajectory in red, while the iterations between them are presented in blue, and points are presented in black.

Figure 7. Hybrid Algorithm execution for all iterations.
To see the iteration of the algorithm more precisely and study the nature of trajectories obtained, we plotted only the best iterations where the \textit{n\textsubscript{GlobalBest}} was actualized in the complex search process, starting from the initial trajectory until the final trajectory, with their minimum cycle times, as presented in Figure 8. The figure presents the series of best candidate trajectories explored by the Hybrid Algorithm. We can see that the algorithm progresses well, and it could converge and find the quasi-optimal trajectory among many possibilities, beginning with a starting trajectory which includes three points (Starting point, First inner point, Ending point). This trajectory was initialized in the input arguments of the code script. From this trajectory, the algorithm starts to converge to the quasi-optimal one using TS optimization based on the evolving of the neighboring non-tabu splines of previous spline. A neighbor spline passes over one of the neighbor points of the first inner point, thus creating a trajectory. Neighbor splines are created first using the set of neighboring points, then the algorithm calculates the cycle time for each of them. By comparing the values of cycle times for the neighboring non-tabu candidate splines, the local optimum spline for the next step is selected. The neighborhood represents the locality of the TS algorithm.

![Figure 8](image)

Figure 8. The best candidate trajectories executed by the Hybrid Algorithm. The point P2 presents the convergence of the first inner point, the points P3 and P4 are created by Point Insertion method during the execution of the Hybrid Algorithm.

As was mentioned in the introductory part of this section, each cycle time for a given spline is a minimal cycle time for that spline determined with a borrowed, not detailed algorithm, utilizing the limits of the joint torques.

Tabu Search converges to the global best trajectory, as presented in Figure 8. The algorithm goes from starting trajectory to best trajectory, overwriting the \textit{n\textsubscript{GlobalBest}} for the first time, since the minimum cycle time for this trajectory \textit{ctmin} = 2.59 (s) is smaller than the starting one which has \textit{ctmin} = 5.5 (s), so from our first glimpse at its operation, we remark that the algorithm optimizes the cycle time by a great percentage.
By completing the same process, the algorithm converges to the second best trajectory for the second time, then third best trajectory, where the number of points included in these trajectories $N = 3$; as we can see, the minimum cycle time in each iteration becomes shorter: for the third round, the value of the cycle time for the best trajectory is $ct_{min} = 1.81$ (s).

From the best trajectory in the fourth round, the algorithm completes this Tabu Search period—finding the best trajectory that can be created with three points—and then jumps to the second method (Point Insertion) by inserting the second inner point in the trajectory, and the process of Tabu Search starts working again to find a fifth best trajectory, which includes four points and its minimum cycle time is $ct_{min} = 1.78$ (s). By iterating another time, the TS algorithm finishes with a sixth trajectory which results in an unacceptable cycle time compared to the previous values; therefore, the second method is applied by inserting the third inner point in the trajectory, but stops here due to the condition of the number point limit $N = 5$. The algorithm did not go to the third method, which is grid refinement in this case, because we reached the limit of $N$. Thus, the algorithm executed the final $n_{GlobalBest}$ state’s trajectory with $ct_{min} = 1.75$ (s).

From this process, we can confirm that the Hybrid Algorithm can find a quasi-optimal trajectory starting with a random one by converging to the best trajectory with minimum cycle time value. Unfortunately, the static figures do not display the dynamic behavior of the robot motion, so we have to describe in words that the best trajectory track begins with a steeply sloping section where the gravitational force helps to accelerate the links of the robot arm and the second finishing part of the track utilizes this motion energy to accomplish the track very speedily. This interesting motion, which reminds us of the robot arm motion analyzed in whip-lashing theory [47], seems to be effective in providing high speed and small cycle time when travelling a route between two given points. (It is also very similar to the motion of the carriage on the first part of a rollercoaster track in an amusement park.)

The primary aim of the research was to elaborate a new Hybrid Algorithm that is more effective than the Tabu Search Algorithm alone, because the Hybrid Algorithm determines more precisely the optimal motion path of the robot arm in order to further reduce the cycle time and utilize more efficiently the joints’ torque.

During the trajectory optimization, the new Hybrid Algorithm searches for the globally optimal path of a robot arm end effector starting from a simple trajectory generated between the given starting point (S) and the ending point (E). The first inner point is arbitrary generated between the given S and E points. The starting trajectory is defined by the cubic spline function of the MATLAB software. From this curve, the Hybrid Algorithm converges and finds the globally optimal curve by refining the trajectory. Thus, the Hybrid Algorithm tries to find a better inner point that replaces the first inner point. It is the best trajectory. After this, the Hybrid Algorithm continues the searching for the best trajectory in the space of trajectories in order to provide the globally minimal cycle time; in the case study, this is from the first best trajectory to the final best, which is the seventh best trajectory.

If we would like to compare the efficiency of the Hybrid Algorithm to the application of the Tabu Search Algorithm, we only have to compare the simulation results. The best trajectory ($ct_{min} = 2.59$ (s) cycle time) is the result of the individual application of the Tabu Search Algorithm which is the part of the Hybrid Algorithm. On the contrary, the final best trajectory ($ct_{min} = 1.75$ (s) cycle time) is the final result of the application of the Hybrid Algorithm which is the integration of the Tabu Search Algorithm, the Point Insertion method and the Grid Refinement method. The comparison between the application of the Tabu Search Algorithm and the Hybrid Algorithm is presented by the comparison between the cycle times of these two trajectories generated by the two algorithms. The efficiency of the Hybrid Algorithm was validated by the case study, showing that application of the new Hybrid Algorithm resulted in 32% shorter motion cycle time.
4.3. Case Study for Joints Torque Function Regarding Starting and Final Best Trajectories

This section deals with another subject, namely the working of the \( f(spline) \) function borrowed from the whip-lashing theory that determines the lowest cycle time for a given trajectory. As was mentioned earlier, the \( f(spline) \) function is a part of the core of Hybrid Algorithm but here, in this section introducing its dynamic simulation aspect, we refer to the concept of whip-lashing theory at a minimum, because we would like the reader to understand the dynamic simulation which results in the cycle time \( ct \) for a given trajectory. It is an iteration method based on the torque of the joints, maximally utilizing the allowable torque. This iteration method in the \( f(spline) \) function can repeatedly simulate a trajectory running with successively decreased cycle times and finally, as a result, it gives the least cycle time \((ct)\) for that trajectory. Additionally, it is capable of showing the change in the joints’ torque along the trajectory as a \( torque = torque(time) \) function. The simulation model built in MATLAB Simulink environment is presented in Figure 9 with its block diagram, where different sub-blocks are described as follows:

- **Trajectory** block includes the inverse kinematic solution for the different position points connected using the spline function;
- **RV-2AJ arm** block includes the mechanical structure of the RV-2AJ robot arm with its given parameters and configuration;
- **Torque cost function** block produces the dot product of the torque matrix with its transpose, and this result can be used in the rare case that Tabu Search finds two or more trajectories with the same least cycle time, so it will choose the best one by comparing them by the minimum torque cost value;
- **aT block** describes the vector of allowable torques for each joint of the RV-2AJ robot arm;
- **mT block** describes the maximum torque in the iteration;
- **Cycle time algorithm** presents the core iterative algorithm of the whip-lashing theory responsible for determining the minimum cycle time \((ct)\);
- **TM (i)(j) block** presents the vector of joints torque for each moment from the simulation;
- **Joint torque visualization** presents a scoop block that visualizes the joint torque diagrams.

![Figure 9. The block diagram for cycle time determination.](image-url)
The main result of running the program based on this block diagram was the cycle time and the visualization of the torque functions, where these functions help us to understand the behavior of the simulation and the behavior of the robot along the trajectory.

Running the simulation of the block diagram in a Simulink environment, RV-2AJ will behave according to the algorithm, and the appearing Mechanics Explorers window shows the dynamic simulation, as presented in Figure 10.

![Figure 10](image_url)

Figure 10. Dynamic simulation of different trajectories in the Mechanics Explorer from different views.

Figure 10 presents more given trajectories. The simulations of trajectory execution introduce the robot motion dynamics and picture their different cycle times.

In the last part of this section, we present the last shoot of two iterations when the whip-lashing algorithm produced the least cycle time for the starting trajectory and final best trajectory (see Figure 8 also). The \( \text{torque} = \text{torque}(\text{time}) \) functions for these two trajectories and for four joints are shown in Figure 11. The very quick changes of the curves at the start are additionally enlarged in the diagrams.

Figure 11 illustrates the joint torque variation for starting and final paths with their minimum cycle time values \( \text{ct} = 5.5 \) (s) and \( \text{ct} = 1.75 \) (s). The figure includes four sub-figures; each sub-figure presents a joint (Waist-Shoulder-Elbow-Wrist) and every sub-figure describes the torque function for two different splines and the allowable torque value \( aT \) for such a joint. The purpose of this figure is to show that reaching the allowable torque happens for different joints for the different splines, whereas for the first trajectory, the reaching of allowable torque is in the Waist joint with \( \text{ct} = 5.5 \) (s) and \( aT_1 = 15 \) (N·m) colored in brown; for the final trajectory, allowable torque is reached in the Shoulder joint with \( \text{ct} = 1.75 \) (s) and \( aT_2 = 35 \) (N·m) with the blue broken line.
The Hybrid Algorithm based on Tabu Search is a freshly elaborated optimization technique which aims to refine the motion trajectory executed by a robot arm. We can conclude from the analysis of the Hybrid Algorithm that it selects the trajectory from the space of possible trajectories with the smallest cycle time, and the cycle times are determined by applying dynamic simulations such as those introduced by the torque diagrams shown in Figure 11.

5. Conclusions and Future Research

(1) The findings and the results of the research and their applicability are the following:

- The authors elaborated a new Hybrid Algorithm for the optimization of a robot arm’s trajectory. This Hybrid Algorithm is based on the Tabu Search Algorithm and also uses two added methods—Point Insertion and Grid Refinement methods—simultaneously to determine that trajectory of the robot arm from the space of possible trajectories with the lowest cycle time between two given points, avoiding obstacles.
- The optimization of the robot arm’s trajectory by the application of the newly elaborated Hybrid Algorithm results in an even further reduction in cycle time and more efficient utilization of joints’ torque compared to the individual application of the Tabu Search Algorithm.
- Due to further reduction in the cycle time, the application of the new Hybrid Algorithm results in even higher efficiency improvement and provides a greater contribution to sustainable production compared to the individual application of the Tabu Search Algorithm.
- The efficiency of the Hybrid Algorithm was validated by a case study in which 32% shorter motion cycle time was obtained by the application of the new algorithm. The application of the Hybrid Algorithm was executed using MATLAB software by creating a dynamic model of a RV-2AJ Mitsubishi robot arm with 5 degrees of freedom.
• The minimization of motion cycle time causes significant increasing of the productivity in practice, which means that more final products can be produced in a given time interval. It results in the reduction of specific production costs, thereby reducing the final product’s cost. It can be concluded that the efficiency improvement results in cost saving for the production companies.
• The enterprises can use this cost saving for further development and can invest in environmentally friendly, energy-efficient technologies and equipment which can further improve sustainability in the production. In addition, smaller cycle times mean less demand against all infrastructure usage, heating and lighting of the manufacturing shop, standby time of error protection systems or readiness of special human resources such as troubleshooting teams. The gained time, energy and workforce can be used in other areas of production.

(2) The main added values of the research are the following:
• The main contribution of the research is the elaboration of a new Hybrid Algorithm which can define the quasi-optimal path that has the lowest cycle time between the starting and ending points, avoiding given obstacles. This new algorithm results in an even further reduction in cycle time and more efficient utilization of joint torque compared to the individual application of the Tabu Search Algorithm.
• Further added value of the research is that the newly elaborated Hybrid Algorithm can solve the two main tasks—both the path planning task determining the best motion path and the dynamic task of the trajectory optimization using maximal allowed joint torques—in an integrated form simultaneously.
• The newly elaborated Hybrid Algorithm is an absolutely novel optimization algorithm in the recent literature and fills the previously mentioned gaps in this research field.

(3) The application of the newly elaborated Hybrid Algorithm for the optimization of the robot arm’s trajectory has three main ways of contributing to sustainability in terms of environmental, social and economic aspects:
• The industrial robots—as environmentally friendly, energy-efficient and innovative technologies—have the capability to significantly increase productivity. In addition, the optimization of the robot arm trajectory by the application of the new Hybrid Algorithm results in an even further reduction in cycle time compared to the individual application of the Tabu Search Algorithm. Therefore, the shorter cycle time causes higher productivity. This efficiency improvement results in reduction of specific production costs (including labor costs, energy costs, etc.), thereby the reduction of the final product’s cost. The application of energy-efficient robots causes higher utilization of resources, energy-saving and less environmental damage.
• There are several industrial tasks for which industrial robots are more suitable than human workers, because the conditions of some workplaces can be hazardous for workers (e.g., chemical, ergonomic or physical hazards) and the safety of all workers has become a priority.
• Generally, the application of industrial robots is required at manufacturing companies in order to eliminate the lack of decreasing human resources.

In our future research, the Hybrid Algorithm can be improved using local coordinate systems of the interpolated spline points instead of a global grid, which is recently a limitation. This means that neighboring spline points having equal distances from the actual spline can be formulated and this distance can be halved in the iteration instead of the grid step. Additionally, the manually set parameters of the Hybrid Algorithm can be fine-tuned using Genetic Algorithm in case of frequent use of the algorithm for similar robot arms. The algorithm can be installed into new generation robot controllers that can apply it for optimizing any trajectory on the sparsely defined traveling route of the end
effector. The handling of sets of obstacles is also easy. These research directions will form the foundation of our future investigations.

**Author Contributions:** Conceptualization, R.B., L.D. and G.K.; methodology, R.B. and L.D.; software, R.B.; writing—original draft preparation, R.B.; writing—review and editing, R.B., L.D. and G.K.; visualization, R.B. and L.D.; supervision, L.D. and G.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** The research was supported by the Stipendium Hungaricum Scholarship Programme launched by the Hungarian Government.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

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

**References**

1. Koberg, E.; Longoni, A. A systematic review of sustainable supply chain management in global supply chains. *J. Clean. Prod.* 2019, 207, 1084–1098. [CrossRef]
2. Stock, T.; Seliger, G. Opportunities of Sustainable Manufacturing in Industry 4.0. *Procedia CIRP* 2016, 40, 536–541. [CrossRef]
3. Grondys, K.; Androniceanu, A.; Dacko-Pikiewicz, Z. Energy management in the operation of enterprises in the light of the applicable provisions of the energy efficiency directive (2012/27/EU). *Energies* 2020, 13, 4338. [CrossRef]
4. Benotsmane, R.; Kovács, G.; Dudás, L. Economic, Social Impacts and Operation of Smart Factories in Industry 4.0 Focusing on Simulation and Artificial Intelligence of Collaborating Robots. *Soc. Sci.* 2019, 8, 143. [CrossRef]
5. Ghadimi, P.; Wang, C.; Lim, M.K. Sustainable supply chain modeling and analysis: Past debate, present problems and future challenges. *Resour. Conserv. Recycl.* 2019, 140, 72–84. [CrossRef]
6. Walker, H.; Seuring, S.; Sarkis, J.; Klassen, R. Sustainable operations management: Recent trends and future directions. *Int. J. Oper. Prod. Manag.* 2014, 34, 1–12. [CrossRef]
7. Yawar, S.A.; Seuring, S. Management of Social Issues in Supply Chains: A Literature Review Exploring Social Issues, Actions and Performance Outcomes. *J. Bus. Ethics* 2017, 141, 621–643. [CrossRef]
8. Kovács, G.; Illés, B. Development of an optimization method and software for optimizing global supply chains for increased efficiency, competitiveness, and sustainability. *Sustainability* 2019, 11, 1610. [CrossRef]
9. Muangmee, C.; Dacko-Pikiewicz, Z.; Meekaewkunchorn, N.; Kassakorn, N.; Khalid, B. Green Entrepreneurial Orientation and Green Innovation in Small and Medium-Sized Enterprises (SMEs). *Soc. Sci.* 2021, 10, 136. [CrossRef]
10. Szczepańska-Woszczyna, K.; Kurowska-Pysz, J. Sustainable Business Development Through Leadership in SMEs. *Ekon. I Zarz.* 2016, 8, 57–69. [CrossRef]
11. Oncioiu, I.; Petrescu, A.G.; Bilcan, F.R.; Petrescu, M.; Fülöp, M.T.; Topor, D.I. The influence of corporate governance systems on a company’s market value. *Sustainability* 2020, 12, 3114. [CrossRef]
12. Benotsmane, R.; Dudás, L.; Kovács, G. Collaborating robots in industry 4.0 conception. *IOP Conf. Ser. Mater. Sci. Eng.* 2018, 448, 012023. [CrossRef]
13. Kot, S. Sustainable Supply Chain Management in Small and Medium Enterprises. *Sustainability* 2018, 10, 1143. [CrossRef]
14. Kun-Bodnár, K.; Kundrač, J.; Maros, Z. Machining of rotationally symmetric parts with abrasive waterjet cutting. *IOP. Conf. Ser. Mater. Sci. Eng.* 2018, 448, 012053. [CrossRef]
15. Rosyidi, C.N.; Hapsari, S.N.; Jauhari, W.A. An integrated optimization model of production plan in a large steel manufacturing company. *J. Ind. Prod. Eng.* 2021, 38, 186–196. [CrossRef]
16. Jaegler, Y.; Jaegler, A.; Mhada, F.Z.; Trentesaux, D.; Burlat, P. A new methodological support for control and optimization of manufacturing systems in the context of product customization. *J. Ind. Prod. Eng.* 2021, 38, 341–355. [CrossRef]
17. Kundrač, J.; Karpuschewski, B.; Pálmai, Z.; Felhő, C.; Makkai, T.; Borysenko, D. The energetic characteristics of milling with changing cross-section in the definition of specific cutting force by FEM method. *CIRP J. Manuf. Sci. Technol.* 2021, 32, 61–69. [CrossRef]
18. Lee, J.; Davari, H.; Singh, J.; Pandhare, V. Industrial Artificial Intelligence for industry 4.0-based manufacturing systems. *Manuf. Lett.* 2018, 18, 20–23. [CrossRef]
19. Benyoucef, L.; Grabot, B. *Artificial Intelligence Techniques for Networked Manufacturing Enterprises Management*; Springer: London, UK, 2010.
20. Karabegović, I. The role of industrial robots in the development of automotive industry in China. *Int. J. Eng. Work.* 2016, 3, 92–97.
21. Kim, J.; Kim, S.R.; Kim, S.J.; Kim, D.H. A practical approach for minimum-time trajectory planning for industrial robots. *Ind. Robot. Int. J.* 2010, 37, 51–61. [CrossRef]
22. Cservenák, A. Simulation of a mobile robot’s motion. *Acad. J. Manuf. Eng.* 2021, 9, 80–88.
23. Martinez, J.R.G.; Reséndiz, J.R.; Prado, M.A.M.; Miguel, E.E.C. Assessment of jerk performance s-curve and trapezoidal velocity profiles. In Proceedings of the XIII International Engineering Congress, Santiago de Queretaro, Mexico, 15–19 May 2017; pp. 1–7.

24. Liu, X.; Qiu, C.; Zeng, Q.; Li, A. Kinematics Analysis and Trajectory Planning of collaborative welding robot with multiple manipulators. *Procedia CIRP* **2019**, *81*, 1034–1039. [CrossRef]

25. Fang, Y.; Hu, J.; Liu, W.; Shao, Q.; Qj, J.; Peng, Y. Smooth and time-optimal S-curve trajectory planning for automated robots and machines. *Mech. Mach. Theory* **2019**, *137*, 127–153. [CrossRef]

26. Bobrow, J.E.; Martin, B.; Sohl, G.; Wang, E.C.; Park, F.C.; Kim, J. Optimal robot motions for physical criteria. *J. Robot. Syst.* **2001**, *18*, 785–795. [CrossRef]

27. Gasparetto, A.; Boscardi, P.; Lanzutti, A.; Vidoni, R. Path planning and trajectory planning algorithms: A general overview. In *Motion and Operation Planning of Robotic Systems*; Carbone, G., Gomez-Bravo, F., Eds.; Springer: Berlin/Heidelberg, Germany, 2015; Volume 29.

28. Lozano-Pérez, T.; Wesley, M.A. An algorithm for planning collision-free paths among polyhedral obstacles. *Commun. ACM* **1979**, *22*, 560–570. [CrossRef]

29. Elbanhawi, M.; Simic, M.; Jazar, R. Autonomous Robots Path Planning: An Adaptive Roadmap Approach. *Appl. Mech. Mater.* **2013**, *373*, 246–254. [CrossRef]

30. Glavaški, D.; Volč, M.; Bonkovic, M. Mobile robot path planning using exact cell decomposition and potential field methods. *WSEAS Trans. Circuits Syst.* **2009**, *8*, 789–800.

31. Torres, F.; Puente, S.; Díaz, C. Automatic cooperative disassembly robotic system: Task planner to distribute tasks among robots. *Control Eng. Pract.* **2009**, *17*, 112–121. [CrossRef]

32. Zhu, Q.; Yan, Y.; Xing, Z. Robot Path Planning Based on Artificial Potential Field Approach with Simulated Annealing. In *Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications*, Jinan, China, 16–18 October 2006; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NY, USA, 2006; Volume 2, pp. 622–627.

33. Beom, H.R.; Cho, H.S. A sensor-based navigation for a mobile robot using fuzzy logic and reinforcement learning. *IEEE Trans. Syst. Man Cybern.* **1995**, *25*, 464–477. [CrossRef]

34. Janglová, D. Neural Networks in Mobile Robot Motion. *Int. J. Adv. Robot. Syst.* **2004**, *1*, 15–22. [CrossRef]

35. Gerke, M. Genetic path planning for mobile robots. *Proc. Amer. Control Conf.* **1999**, *4*, 2424–2429.

36. Garcia, M.P.; Montiel, O.; Castillo, O.; Sepúlveda, R.; Melin, P. Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation. *Appl. Soft Comput.* **2009**, *9*, 1102–1110. [CrossRef]

37. Elbanhawi, M.; Simic, M. Sampling-Based Robot Motion Planning: A Review. *IEEE Access* **2014**, *2*, 56–77. [CrossRef]

38. Dijkstra, E.W. A note on two problems in connexion with graphs. *Numer. Math.* **1959**, *1*, 269–271. [CrossRef]

39. Hart, P.E.; Nilsson, N.J.; Raphael, B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Trans. Syst. Sci. Cybern.* **1968**, *4*, 100–107. [CrossRef]

40. Glover, F. Tabu Search—Part I. *ORSA J. Comp.* **2019**, *1*, 190–206. [CrossRef]

41. Brady, M.; Hollerbach, J.; Timothy, L.; Lozano-Pérez, T.; Mason, M.T. Robot Motion: Planning and Control; MIT Press: Cambridge, MA, USA, 1983.

42. Lewis, F.L.; Dawson, D.M.; Abdallah, C.T. *Robot Manipulator Control: Theory and Practice*, 2nd ed.; Marcel Dekker Inc.: New York, NY, USA, 2003.

43. Sobrino, D.R.; Koštál, P.; Cagáňová, D.; Cambíl, M. On the Possibilities of Intelligence Implementation in Manufacturing: The Role of Simulation. *Appl. Mech. Mater.* **2013**, *309*, 96–104. [CrossRef]

44. Lin, C.; Chang, P.; Luh, J. Formulation and optimization of cubic polynomial joint trajectories for industrial robots. *IEEE Trans. Autom. Control.* **1983**, *28*, 1066–1074. [CrossRef]

45. Benotsmane, R.; Kacemi, S.; Benachenhou, M.R. Calculation methodology for trajectory planning of a 6 axis manipulator arm. *Ann. Fac. Eng. Hanoi* Int. J. Eng. **2018**, *3*, 27–32.

46. Gupta, S.M.; Del Pobil, A.P. *Practical Motion Planning in Robotics: Current Approaches and Future Directions*; John Wiley & Sons: Hoboken, NJ, USA, 1998.

47. Benotsmane, R.; Dudás, L.; Kovács, G. Trajectory Optimization of Industrial Robot Arms Using a Newly Elaborated “Whip-Lashing” Method. *Appl. Sci.* **2020**, *10*, 8666. [CrossRef]

48. Balkan, T. A dynamic programming approach to optimal control of robotic manipulators. *Mech. Res. Commun.* **1998**, *25*, 225–230. [CrossRef]

49. Glover, F.; Laguna, M. Tabu search principles. In *Tabu Search*; Springer: Boston, MA, USA, 1997; pp. 125–151.

50. Alshibli, M.; El Sayed, A.; Kongar, E.; Sobh, T.M.; Gupta, S.M. Disassembly Sequencing Using Tabu Search. *J. Intel1. Robot. Syst.* **2016**, *82*, 69–79. [CrossRef]

51. Perutka, K. *MATLAB for Engineers—Applications in Control, Electrical Engineering, IT and Robotics*; Intech: Rijeka, Croatia, 2011.

52. Cao, B.; Dodds, G.I. Time-optimal and smooth joint path generation for robot manipulators. In *Proceedings of the International Conference on Control—Control ‘94*, Coventry, UK, 21–24 March 1994; pp. 1122–1127.