Xiaowen Yu
Mechanical System Control Laboratory,
Department of Mechanical Engineering,
University of California,
Berkeley, CA 94720
e-mail: aliceyu@berkeley.edu

Yu Zhao
Mechanical System Control Laboratory,
Department of Mechanical Engineering,
University of California,
Berkeley, CA 94720
e-mail: yzhao334@berkeley.edu

Cong Wang
Assistant Professor
Department of Electrical and Computer Engineering,
New Jersey Institute of Technology,
Newark, NJ 07102
e-mail: cong.wang@njit.edu

Masayoshi Tomizuka
Professor
Department of Mechanical Engineering,
University of California,
Berkeley, CA 94720
e-mail: tomizuka@berkeley.edu

Trajectory Planning for Robot Manipulators Considering Kinematic Constraints Using Probabilistic Roadmap Approach

Trajectory planning is a fundamental problem for industrial robots. It is particularly challenging for robot manipulators that transfer silicon wafers in an equipment front end module (EFEM) of a semiconductor manufacturing machine where the work space is extremely limited. Existing methods cannot give satisfactory performance since they usually solve the problem partially. Motivated by this demand in industrial applications and to solve all aspects of the problem, this paper proposes to learn the work environment beforehand by probabilistic roadmap (PRM) method for collision avoidance. The cycle time preference and the robot kinematic hard constraints are considered properly. A constrained optimization problem is formulated with the shortest path searched from the roadmap and parametrized by a cubic B-spline curve, which simplifies the optimization process. [DOI: 10.1115/1.4034748]

1 Introduction

Automation of trajectory planning for industrial manipulators working in constrained environment is an important topic. One particularly difficult case is a silicon wafer handling robot that works inside the EFEM of a semiconductor manufacturing machine (Fig. 1). In Fig. 1, an EFEM is the interface of a semiconductor manufacturing machine to the factory. One side of the EFEM connects to the factory through loadports (the portal to the outside environment of the semiconductor manufacturing machine, i.e., the atmospheric environment). The other side of the EFEM connects to the wafer processing chambers through loadlocks (the portal to the vacuum environment of the semiconductor manufacturing machine). Wafers are picked up by the wafer handling robot from loadports and then transferred to loadlocks. A vacuum robot picks up wafers from loadlocks and transfers them to vacuum chamber for wafer processing.

Usually, the EFEM has a near-rectangular internal space. The space inside an EFEM is extremely limited relative to the size of the atmospheric wafer transfer robot. In addition, there are usually multiple loadports and loadlocks. The goal is to automate the trajectory planning, which means that with given loadport and loadlock positions, the trajectory planner can generate an "optimal" smooth trajectory that the robot can follow without violating any working constraints. The working constraints are defined in work space as well as in joint space. The first constraint is the obstacle constraint that a robot should not collide with obstacles (which is EFEM in this case). The second constraint is the end-effector constraint which requires that the initial and final positions and orientations of the end-effector are favorable for wafer delivery (pick up). In addition, in order to avoid the sliding of wafers, which may result in particle contamination and even wafer tip-over, the acceleration of the wafer should be limited. There are also limits on joint velocity and acceleration, which are determined by the capabilities of the motors. A trajectory that satisfies all constraints mentioned above should be further optimized to shorten the cycle time, which is crucial for semiconductor manufacturing throughput.

To deal with this complex industrial problem, many planning algorithms are available. However, they can only partially solve this problem. For example, sampling-based motion planners, whose representative examples include rapidly exploring random tree (RRT) method and PRM method [1–3], are good for fast motion planning. In fact, PRM method works well in static environments. However, PRM method suffers from the fact that it is hard to take into account the path smoothness and dynamic constraints, which will result in jerky and unnatural motion. Optimization-based methods [4,5] may have objective functions formulated as the combination of smoothness, obstacle collision, or/and other performance criteria of the trajectory. These methods usually have heavy computational loads and work only for a predetermined duration.

The wafer transferring problem in this paper can be categorized as a multi-query planning problem because of the multiple loadlock and loadport positions, compared to the single query planning problem where a robot manipulator only goes to a single desired goal position. It is desirable to extensively investigate the working environment prior to search for one particular path. PRM method is suitable in this case since it can rapidly construct a roadmap of the environment.

In order to smooth out the trajectory, one commonly used smoothing method uses shortcutting heuristic, which is to replace a subpath by a shortcut segment when it is collision free [6]. This method, however, cannot generate trajectories with higher-order smoothness and cannot guarantee any optimality. Spline curves seem to be a suitable choice to fit the path since they are able to

Contributed by the Dynamic Systems Division of ASME for publication in the Journal of Dynamic Systems, Measurement, and Control. Manuscript received February 24, 2016; final manuscript received August 26, 2016; published online October 18, 2016. Assoc. Editor: Davide Spinello.

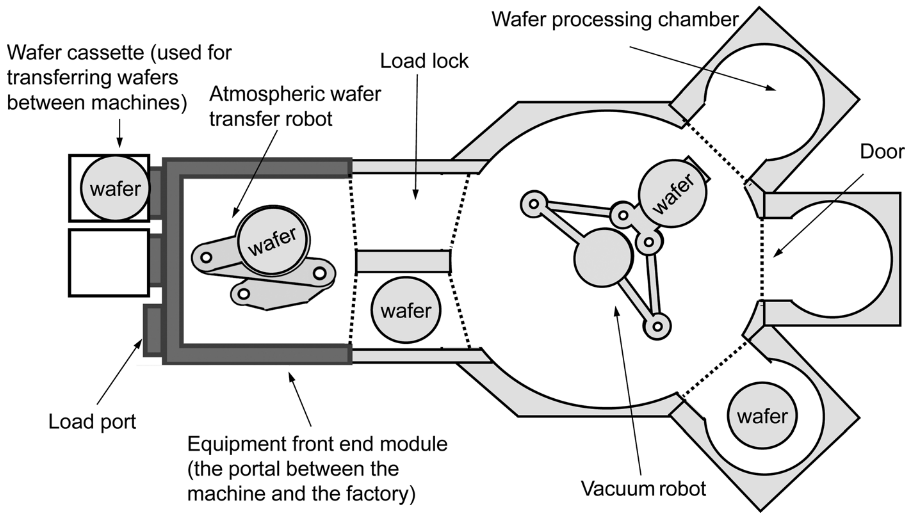

Wafer processing chamber
Wafer cassette (used for transferring wafers between machines)
Load lock
Atmospheric wafer transfer robot
Door
wafer
wafer
wafer
wafer
wafer
Load port
Equipment front end module (the portal between the machine and the factory)
Vacuum robot

Fig. 1 A robot manipulator works inside an EFEM of a semiconductor manufacturing machine

restrict the trajectory in a safe region, and in the mean time parameterize the trajectory with temporal information. Bezier curve seems to be a good choice because it can generate curves bounded by control points. However, the order of the curve grows with the increasing number of the control points. For example, in this paper, when there are 12 control points, the order of the curve will be 11, which is not realistic. Moreover, Bezier curve only works well for a single piece of curve. On the other hand, B-spline function is ideal for generating a curve which has several segments since continuity condition is automatically satisfied. Also, it is able to restrict the curve in a safe region and in the mean time keeps the order of the curve as a constant.

However, direct B-spline fitting always violates the original path and leads to collision. Gasparetto and Zanotto [7] use B-spline function to generate and optimize a trajectory on a set of via-points. While all the kinematics constraints are satisfied, it cannot guarantee that the interpolation is a safe one when obstacles are present.

In this paper, a cubic B-spline function is used to fit the path. By carefully locating the control points, it is guaranteed that the fitting is collision free. A constrained optimization problem is formulated with the objective function as the total execution time and constraints as kinematic constraints. The B-spline curve parametrization makes the optimization problem extremely simplified and a good initial point can be determined by a standard routine. Online implementation is possible with this approach which is desirable in industrial applications.

The rest of the paper is organized as follows: In Sec. 2, the kinematic model of the robot and a model for the obstacle are described. The path finding problem is defined. In Sec. 3, an adaptive PRM planner with three types of samplers is introduced to generate the roadmap with the given obstacles. With a query as an input to the PRM planner, it returns a piecewise linear safe path, which is in fact a shortest path in the roadmap. Section 4 gives the whole algorithm of transferring the piecewise linear path into a smooth trajectory that the robot can follow satisfying all the constraints. Section 5 shows by simulation the effectiveness of the algorithm in Sec. 4.

2 Configuration of the Atmospheric Wafer Transfer Robot and the Obstacle

The atmospheric wafer transfer robot is a three-axis robot with three revolute joints. The robot is a planner robot whose motion is restricted in the *x*–*y* plane, as shown in Fig. 2.

The robot works in an EFEM, which gives an extremely limited work space. Because the robot is a planer one, the EFEM can be modeled as a closed boundary around the robot. The robot needs to pick up wafers from loadports and unload wafers to loadlocks. In this paper, however, only the motion inside the EFEM is considered. The beginning of the motion is right after the robot picks up the wafer from the loadport and retracts back to EFEM. The end of the motion is before the robot extends to unload wafer to the loadlock, as shown in Fig. 3. There are multiple positions for loadports and loadlocks. The top view of the boundary, loadports and loadlocks, as well as the robot is shown in Fig. 4. The goal of trajectory planning is that given the positions of a loadport and a loadlock (and immediately the corresponding configurations inside the EFEM), generate a safe trajectory which satisfies all constraints.

The constraints include:

1. The trajectory should be defined in robot joint space.
2. Robot should not collide with obstacles (which is the boundary).
3. Constraints on the maximum velocity and acceleration of the joints should be satisfied.
4. The wafer-center (end-effector) acceleration amplitude should be limited within a threshold so as to prevent wafer sliding.

In addition to the constraints, the cycle time should be as short as possible because productivity is a very sensitive issue in industrial applications. The total time duration of the trajectory should be minimized without violating any constraints.

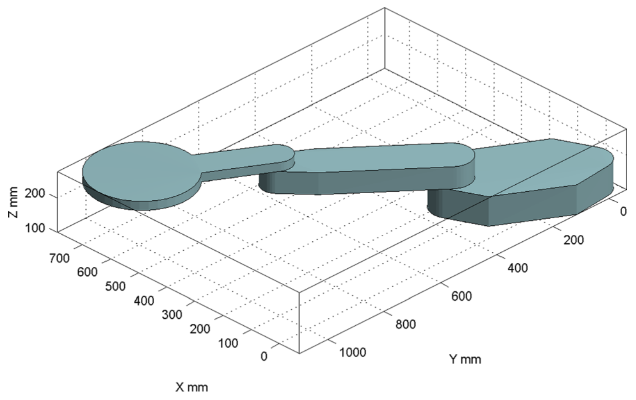


Fig. 2 Model of the three-axis manipulator works in an EFEM: (*a*) an initial robot configuration inside the EFEM and (*b*) a goal robot configuration inside the EFEM

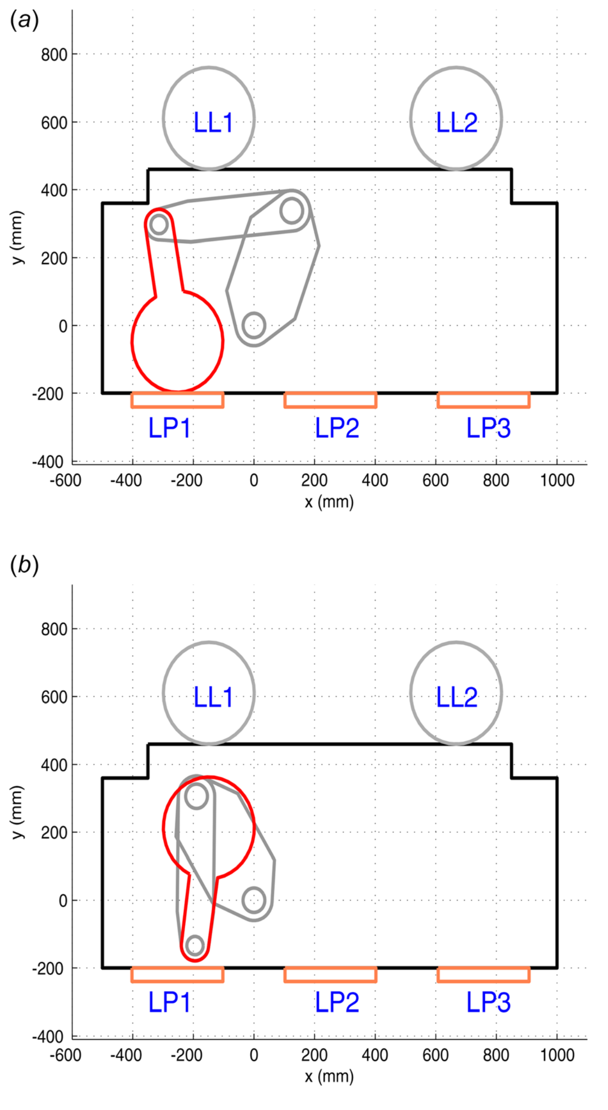


Fig. 3 Motion is considered inside the EFEM

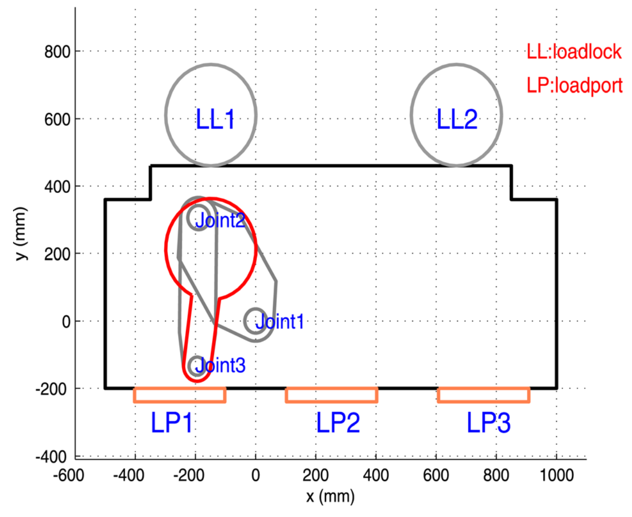


Fig. 4 The EFEM is modeled as a closed boundary obstacle: (*a*) obstacle-free space and (*b*) samples generated by PRM (1500 nodes and 39,900 edges, edges are not displayed)

3 Probabilistic Roadmap Planning

A PRM planner is formulated for the multi-query problem in this paper. "The basic idea behind PRM is to take random samples from the configuration space of the robot, testing them for whether they are in the free space, and use a local planner to attempt to connect these configurations to other nearby configurations. The starting and goal configurations are added in, and a graph search algorithm is applied to the resulting graph to determine a path between the starting and goal configurations."[1] In this paper, PRM tries to construct a roadmap that covers the obstacle-free space as much as possible. Because of the unknown property of the obstacle-free space, an adaptive sampling method is adopted to adjust samplers. The detailed formulation of this method can be found in Ref. [1].

There are three samplers in the sampler pool each with a distinctive strength: uniform sampler, Gaussian sampler, and bridge test sampler. The uniform sampler provides good coverage of C-space while the Gaussian and bridge test samplers deal with the narrow passages if existed in the C-space. The sampling strategy is to give reward to the sampler that has generated useful point and punish sampler that has generated useless point. The result of the roadmap constructed in this way is given in Figs. 5 and 6. The red region is the obstacle-free space, all the blue dots are sampled points (robot configurations) generated by samplers (Fig. 5(*b*)), and all the thin lines show the connectivity between two points in the obstacle-free space (Fig. 6). The roadmap shows that the proposed method can provide a good coverage of the whole obstacle-free space.

Once a roadmap is constructed, the PRM planner enters another phase: the query phase. With a given query, which is a pair of initial and goal configurations (q_I and q_G), an A-star search method is adopted to search through the roadmap for a shortest path, as shown in Fig. 6 (red line).

4 Trajectory Generation Using Cubic B-Spline

With the path searched from the roadmap, it has to be smoothed out and parametrized by time "t" to become a trajectory. The trajectory must be smooth and safe (no collision happens). Besides, the execution time should be as short as possible. In this section, the trajectory is parametrized by time intervals based on cubic B-spline functions and then it is optimized with respect to the total execution time while not violating any operational constraints.

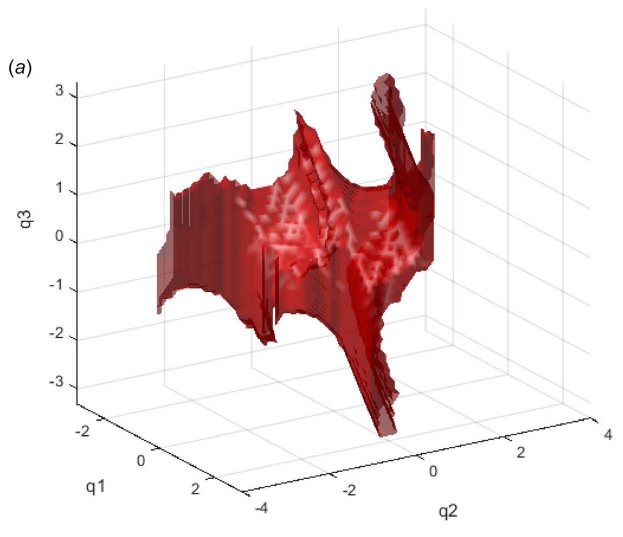
4.1 Blending Within a Safe Region.

The basic idea of trajectory smoothing is to blend the connecting points of the piecewise linear path searched from the roadmap (Fig. 6) by spline curves. Because only the piecewise linear path is feasible, each connecting point of the path is first bounded by a safe region, which is a box that the trajectory is safe as long as it remains inside the box, as shown in Fig. 7. Once the boxes are generated, safe blending can be performed to generate smooth trajectory within the box.

The bounding boxes are generated based on the shortest distance of each connecting point to the obstacle. The goal is to find a box for a given connecting configuration q, i.e., to find a Δq ($\Delta q = [\Delta q_1, \Delta q_2, \Delta q_3]^{\mathrm{T}}$, and $\Delta q_i > 0$, $i = 1,\ 2,\ 3$ for three joints) such that a new configuration q' is a safe one as long as $|q'_i - q_i| < \Delta q_i, i = 1, 2, 3$. With the given shortest distance d from robot to obstacle at the given configuration q, Δq can be determined if all points on the robot links travel less than distance d when the robot moves from configuration q to q'.

This problem can be further decomposed for three joints, respectively. First, for each joint (joint i), a Lipschitz condition is derived as

[1]"Probabilistic Roadmap," accessed August 26, 2016, https://en.wikipedia.org/wiki/Probabilistic_roadmap.



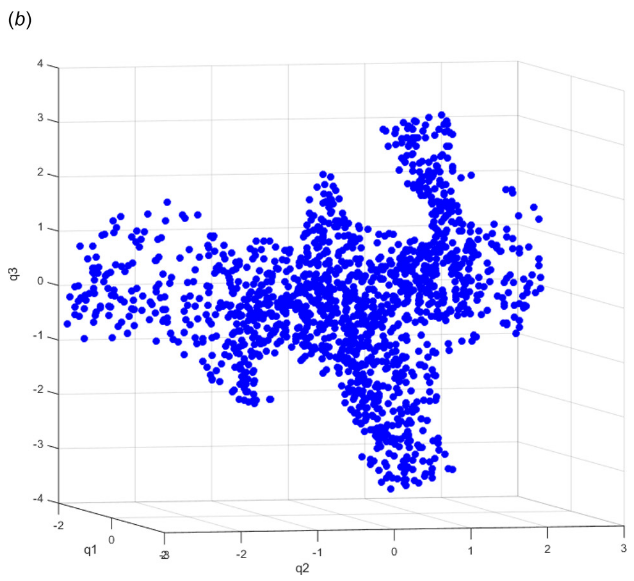


Fig. 5 Obstacle-free space and samples generated by PRM

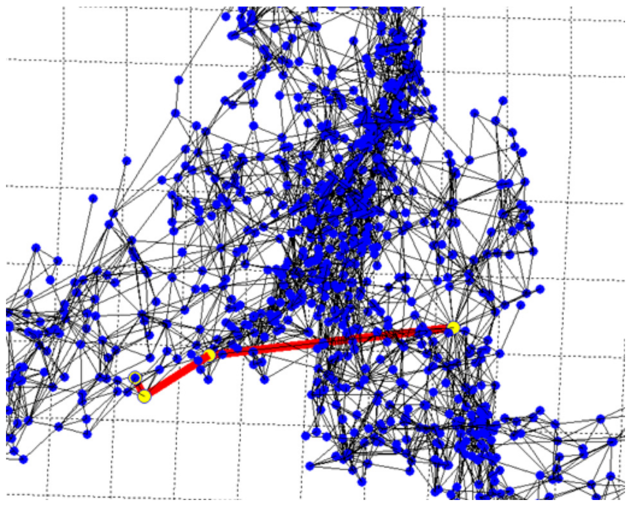
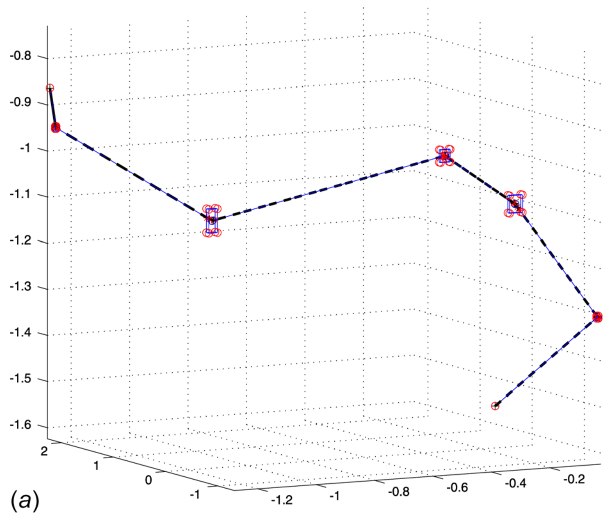


Fig. 6 The shortest path connects q_I and q_G searched in roadmap by A^*: (*a*) bounding boxes for all connecting points of a path and (*b*) bounding box for a single connecting point



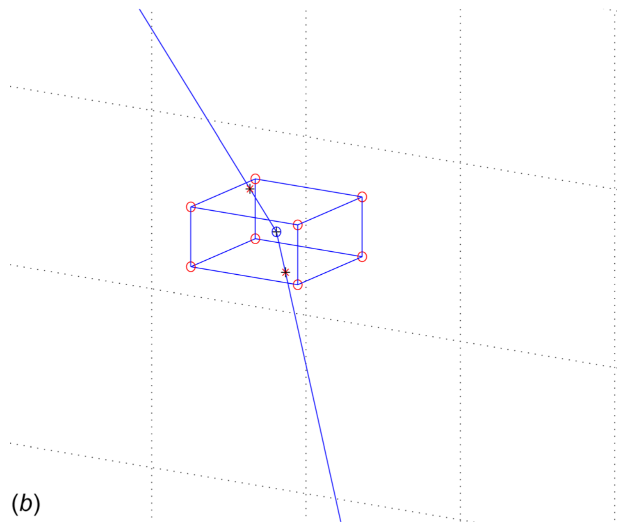


Fig. 7 Bounded region of a given piecewise linear path

$$\left\| \begin{matrix} a(q_1, \ldots, q_{i-1}, q_i, q_{i+1}, \ldots, q_n) - \\ a(q_1, \ldots, q_{i-1}, q_i', q_{i+1}, \ldots, q_n) \end{matrix} \right\| < c_i |q_i - q_i'| \quad (1)$$

in which a is any point on the robot, c_i is a Lipschitz constant, n is the number of joints (in this case, $n = 3$), and the Lipschitz condition is written in a general form. The goal is to make the Lipschitz constant c_i as small as possible to get a bigger variation in q_i. One example of determining c_i is in Fig. 8. In this example, the robot has only one link, and c_i is simply the length of the robot r. The bound of the robot displacement is

$$\|a(q) - a(q')\| < \sum_{i=1}^{n} c_i |q_i - q_i'| \quad (2)$$

Let $\sum_{i=1}^{n} c_i |q_i - q_i'| < d$, Δq is derived as

$$\Delta q_i = \frac{d}{n c_i}, \quad i = 1, 2, 3 \quad (3)$$

The bounding box for configuration q is determined by Δq, also the intersection of straight line segments and bounding boxes are determined (Fig. 7(*b*)).

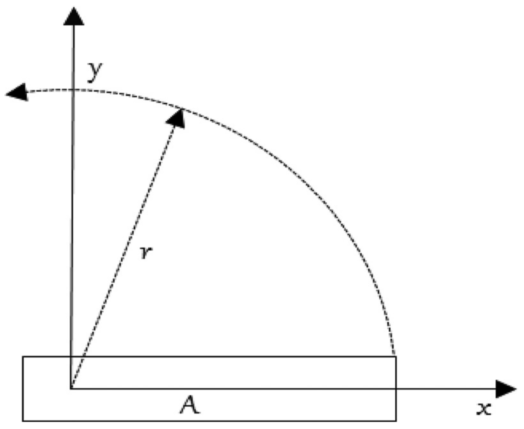


Fig. 8 One link robot example for determining c_i

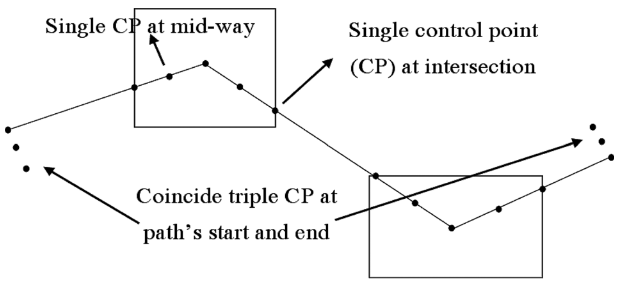

Single CP at mid-way
Single control point (CP) at intersection
Coincide triple CP at path's start and end

Fig. 9 Control points' distribution example: (*a*) trajectory with time sequence substituted and (*b*) zoom in of a segment of the trajectory

Table 1 Symbols used when defining the B-spline

Symbol	Definition
$k-1$	Degree of the B-spline
k	Order of the B-spline
$p(t)$	B-spline curve
$N_{i,k-1}(t)$	Base function of degree $k-1$
CP_i	Control points of the B-spline
$n+1$	Number of control points
t_i	Nodes of the B-spline (time sequence)
$k+n+1$	Number of nodes

4.2 Parametrizing Trajectory by Cubic B-Splines. First, note that a B-spline curve is defined by its degree $k-1$, order k, control points, and nodes. A B-spline curve $p(t)$ is expressed as a linear combination of base functions weighted by control points

$$p(t) = \sum_{i=0}^{n} \mathrm{CP}_i N_{i,k}(t), \quad t_{k-1} \le t \le t_{n+1} \tag{4}$$

where $\mathrm{CP}_0, \ldots, \mathrm{CP}_n$ are $n+1$ control points, $t_{k-1}, \ldots, t_{n+1}$ are nodes (which is a time sequence, and it is clamped, i.e., the values have multiplicity k at extremities so that $p(t_{k-1})$ and $p(t_{n+1})$ coincide with the first and last control points, respectively), and $N_{i,k}(t)$'s are base functions recursively determined by de Boor–Cox formula

$$\begin{aligned} N_{i,0}(t) &= \begin{cases} 1, & t_i \le t \le t_{i+1} \\ 0, & \text{else} \end{cases} \\ N_{i,k}(t) &= \frac{t-t_i}{t_{i+k}-t_i} N_{i,k-1}(t) + \frac{t_{i+k+1}-t}{t_{i+k+1}-t_{i+1}} N_{i+1,k-1}(t), \\ &\text{and let } \frac{0}{0} = 0 \end{aligned} \tag{5}$$

Symbols used when defining B-spline curve is given in Table 1. Because a cubic B-spline curve is at least twice differentiable, which is a favorable property for trajectory smoothness, the cubic B-spline is selected for path fitting.

The trajectory should be restricted within the path and bounded boxes obtained in the previous sections, i.e., the blending curves connecting adjacent straight lines, must be inside bounded boxes. To meet these requirements, with the strong convex hull property of the B-spline (which is given below), triple coincident control points are put at the start and end of the path, which ensures that the start and the end are complete stops (zero speed and acceleration). Also, single control points are put at all connecting points as well as at the midway of intersection points and connecting points. An example of the distribution of the control points is in Fig. 9. With this distribution, the B-spline curve is connected of several segments, and all of them are safe.

Strong convex hull property: A B-spline curve is contained in the convex hull of its control poly line. To be specific, if t is in knot span (t_i, t_{i+1}), $p(t)$ is in the convex hull of control points $\mathrm{CP}_{i-k+1}, \mathrm{CP}_{i-k+2}, \ldots, \mathrm{CP}_i$. This is because $N_{i,k-1}(t) \ge 0$, $\sum N_{i,k-1}(t) = 1$.

With this property, for a B-spline curve of degree $k-1=3$, suppose that control points are $\mathrm{CP}_0, \ldots, \mathrm{CP}_n$, then $k+n+1$ nodes are needed for a complete B-spline trajectory. Let nodes be $t_0, \ldots, t_{k+n}$, which is in fact a time sequence. The trajectory $p(t)$ is parametrized by time sequence while under the safe operation condition. One trajectory with a time sequence substituted is given in Fig. 10, and the trajectory is a safe one.

4.3 The Optimization Problem With Kinematic Constrains. The trajectory parametrized by a time sequence $t_0, \ldots, t_{n-k+2}$ features sufficiently smooth properties, i.e., it is guaranteed that the trajectory, velocity, and acceleration are continuous functions. In industrial applications, however, the execution time is crucial for manufacturing throughput. Thus, the trajectory should be further optimized while taking into account the kinematic constraints.

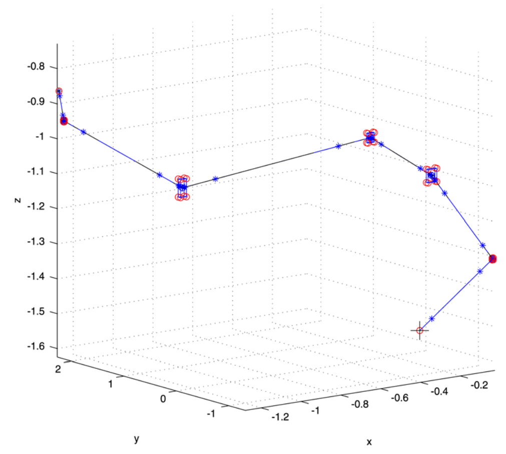
Kinematic constraints are usually upper bounds on joint's velocity and acceleration. However, in this application, the kinematic constraints are defined not only in the joint space but also in the robot work space. The acceleration of the wafer center in the work space should be limited because the wafer is in contact with the blade only by means of friction. It has been proved empirically that wafer sliding is avoided when wafer-center acceleration is bounded.

To explicitly express the constraints in the optimization problem, some remarks about the derivatives of the trajectory are necessary. The derivative of a B-spline curve is

$$\begin{aligned} v(t) = p'(t) &= \left(\sum_{i=0}^{n} \mathrm{CP}_i N_{i,k}(t)\right)' = \sum_{i=0}^{n} \mathrm{CP}_i N'_{i,k}(t) \\ &= (k-1)\sum_{i=1}^{n}\left(\frac{\mathrm{CP}_i - \mathrm{CP}_{i-1}}{t_{i+k}-t_i}\right) N_{i,k-1}(t) \\ &= \sum_{i=1}^{n} \mathrm{CPV}_i \cdot N_{i,k-1}(t) \\ \mathrm{CPV}_i &= (k-1)\left(\frac{\mathrm{CP}_i - \mathrm{CP}_{i-1}}{t_{i+k}-t_i}\right) N_{i,k-1}(t) \end{aligned} \tag{6}$$

in which CPV_i are control points of velocity. Similarly, the acceleration is

(*a*)



(*b*)

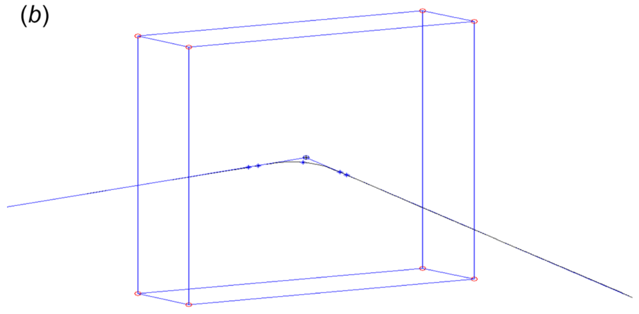


Fig. 10 Trajectory generated by B-spline curve

$$a(t) = v'(t) = \left(\sum_{i=1}^{n} \mathrm{CPV}_i N_{i,k-1}(t)\right)' = \sum_{i=1}^{n-1} \mathrm{CPA}_i \cdot N_{i,k-2}(t) \qquad (7)$$
$$\mathrm{CPA}_i = \frac{k-2}{t_{i+k-1} - t_{i+1}}(\mathrm{CPV}_i - \mathrm{CPV}_{i-1})$$

in which CPA_i are control points of acceleration. Thus, the trajectory, velocity, and acceleration are all expressed in B-spline form. Recall the convex hull property of B-spline curve, which states that a B-spline curve is contained by all of its control points. Thus, to bound a B-spline curve, it is sufficient (not necessary) to bound its control points instead. The kinematic constraints on velocity and acceleration in joint space are then written as

$$\begin{aligned} |\mathrm{CPV}_{j,i}| &\le \mathrm{LV}_j, \quad i = 1, \ldots, n, \quad j = 1,2,3 \\ |\mathrm{CPA}_{j,i}| &\le \mathrm{LA}_j, \quad i = 1, \ldots, n-1, \quad j = 1,2,3 \end{aligned} \qquad (8)$$

in which j is the joint number, and LV_j and LA_j are limits of velocity and acceleration, respectively. It should be noted that all control points of velocity and acceleration are functions of time sequence.

The wafer-center acceleration in work space is first calculated by forward kinematics and is expressed as a piecewise function parametrized by time t and time sequence $t_0, \ldots, t_{n-k+2}$. The acceleration constraints imposed on acceleration function should be held for any time instant, which is an infinite constraints problem that is hard to address. Moreover, because of the forward kinematics involves complex nonlinearity of the robot kinematics, the acceleration constraints in the work space require high-computational load. A pseudo semi-infinite constraints approach is proposed in this paper to express the kinematic constrains in work space efficiently. The acceleration function at all time sequence $t_0, \ldots, t_{n-k+2}$ are sampled and set within the constraints

$$\begin{aligned} &|\mathrm{Acc_{ws}}j(t = t_i; \{t_0, \ldots, t_{n-k+2}\})| \le \mathrm{LA_{ws}}j, \\ &i = 0, \ldots, n-k+2, \quad j = x, y \end{aligned} \qquad (9)$$

in which Acc_ws is the acceleration function, LA_ws is the acceleration limit in work space, and j stands for two axes x and y.

It should also be noted that there is constraint set on the time intervals, which is due to the fact that they are lower bounded because of the limits of the joint velocity. It is hard to get this lower bound because the joint positions are not fixed at the time spot in the time sequence $t_0, \ldots, t_{n-k+2}$. However, it is possible to get a rough lower bound as a substitute by considering going straight line from initial to goal point at maximum speed. Besides, t_0 should be zero

$$\begin{aligned} &t_i - t_{i-1} \ge \min_j \frac{|q_G - q_I|}{\mathrm{LV}}, \\ &i = 1, \ldots, n-k+2, \quad j = 1,2,3, t_0 = 0 \end{aligned} \qquad (10)$$

After all kinematic constraints are defined, the objective function of the optimization problem is defined as the total execution time

$$\min_{\{t_0,\ldots,t_{n-k+2}\}} t_{n-k+2} \qquad (11)$$
s.t. kinematic constraints (Eqs. (7), (9), and (10))

5 Simulation Results of the Optimized Trajectory

The optimization problem formulated in Eq. (11) is a constrained nonlinear optimization problem which is usually solved based on iteration. Thus, the optimization problem is sensitive to the initial point which affects the running time or even the result.

First, the kinematic limits are given in Table 2. For the robot moving from loadport1 to loadlock1, the initial and goal configurations corresponding to the given loadport and loadlock position are shown in Fig. 3. The solution of the optimization problem is obtained by "interior-point" method in MATLAB. The interior-point method requires that the initial searching point should be feasible. By iteratively increasing the gain of the barrier function, it is proved that the interior-point method converge to a solution [8–21].

The feasible initial point can be determined by the following steps: First, an initial time sequence $\mathbf{h}$ is obtained by considering the lower bound of time intervals with Table 2, which is a uniformly distributed time sequence with the final time as $\min_j(|q_G - q_I|/\mathrm{LV}), j = 1,2,3$. Then, by substituting $\mathbf{h}$ into Eq. (4), the B-spline curve $p(t)$ is obtained, and velocity control

Table 2 Values of kinematic limits

	Kinematic limits in joint space	
Joints	Velocity (rad/s)	Acceleration (rad/s^2)
1	4.2	8.4
2	8.4	16.8
3	6.3	12.6
	Kinematic limits in work space	
Axis	Acceleration (mg)	
x	100	
y	100	

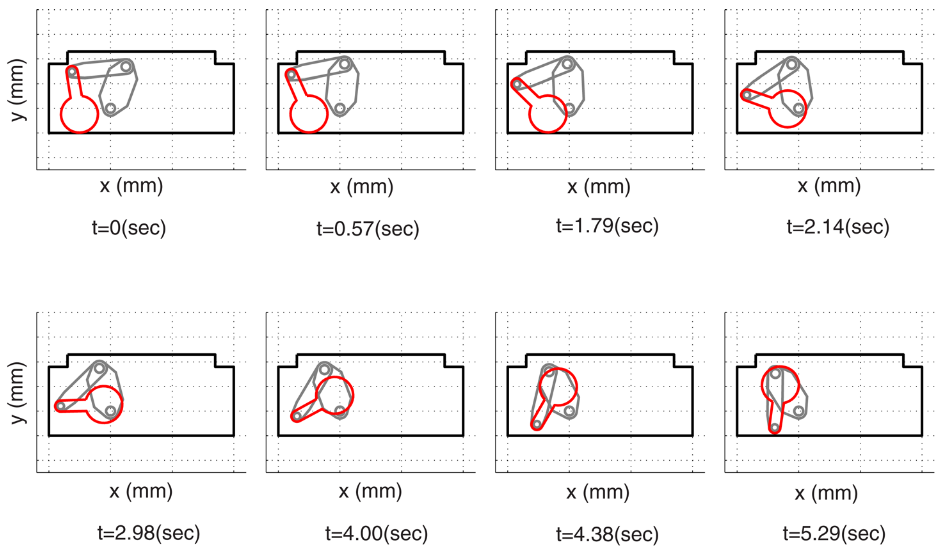


Fig. 11 The motion sequence of the trajectory: (*a*) position of the three joints for the optimized trajectory and (*b*) velocity of the three joints for the optimized trajectory

points (CPV_i), CPA_i as well as acceleration in the workspace ($\mathrm{Acc\_ws}_j$) are all obtained. A scaled time variable τ is defined as $\tau = \lambda t$, and λ can be computed by the control points and workspace acceleration, as given in Eq. (12). Then the initial point is $\mathbf{h}^{(0)} = \lambda \mathbf{h}$

$$\begin{aligned}
\lambda_1 &= \max_{j,i} \frac{\mathrm{CPV}_{j,i}}{\mathrm{LV}_j}, \quad i = 1, \ldots, n, \; j = 1, 2, 3 \\
\lambda_2 &= \max_{j,i} \frac{\mathrm{CPA}_{j,i}}{\mathrm{LA}_j}, \quad i = 1, \ldots, n-1, \; j = 1, 2, 3 \\
\lambda_3 &= \max_{j,i} \frac{\mathrm{Acc\_ws}_{j,i}}{\mathrm{LA\_ws}_j}, \quad i = 1, \ldots, n, \; j = x, y \\
\lambda &= \max\left(1; \lambda_1; \sqrt{\lambda_2}; \sqrt{\lambda_3}\right)
\end{aligned} \tag{12}$$

Figure 11 gives the motion sequence of the generated trajectory. The simulation results in Figs. 12 and 13 show that all constraints are satisfied. Noting that the trajectory is far from the constraints in joint velocity and acceleration mainly because the workspace acceleration constraint is more limited. The proposed method also works when the kinematic constraints are small. The change of the kinematic limits will directly change the initial point (always feasible), and the optimization process will optimize over the initial point and remain feasible.

6 Discussion

In Sec. 3, a roadmap with PRM method is constructed. One important part in generating roadmap is to check the collision for points in robot C-space. The collision checking in this paper is based on geometric collision model which considers all the collision scenarios that can happen between robot and obstacle. This approach is complicated and hard to implement. However, when the computer-aided design models of the robot as well as the EFEM are available, it is possible to utilize these computer-aided design models for fast collision checking. For example, some bounding volume hierarchy-based software [22] are available and can be implemented easily. The result of the optimized trajectory is not globally optimal. It is a suboptimal trajectory. First, the kinematic constraints imposed on the velocity and acceleration (Eq. (7)) are only sufficient but not necessary, which means that there might exist a better trajectory though violating the constraints. Also, because the optimization is a nonconvex one, it is

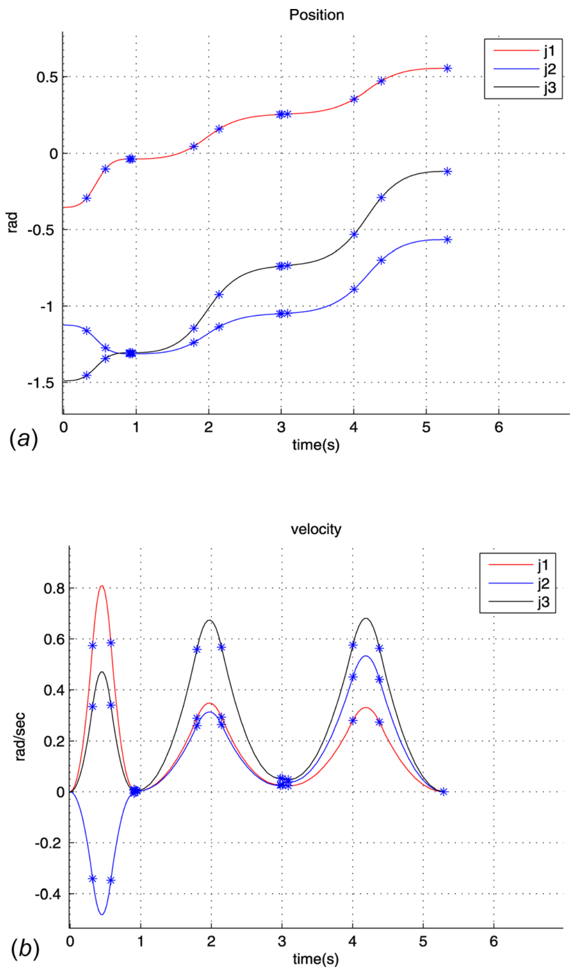


Fig. 12 Position and velocity of the three joints for the optimized trajectory (under constraints in Table 2): (*a*) acceleration of the three joints for the optimized trajectory and (*b*) acceleration in the workspace for the optimized trajectory

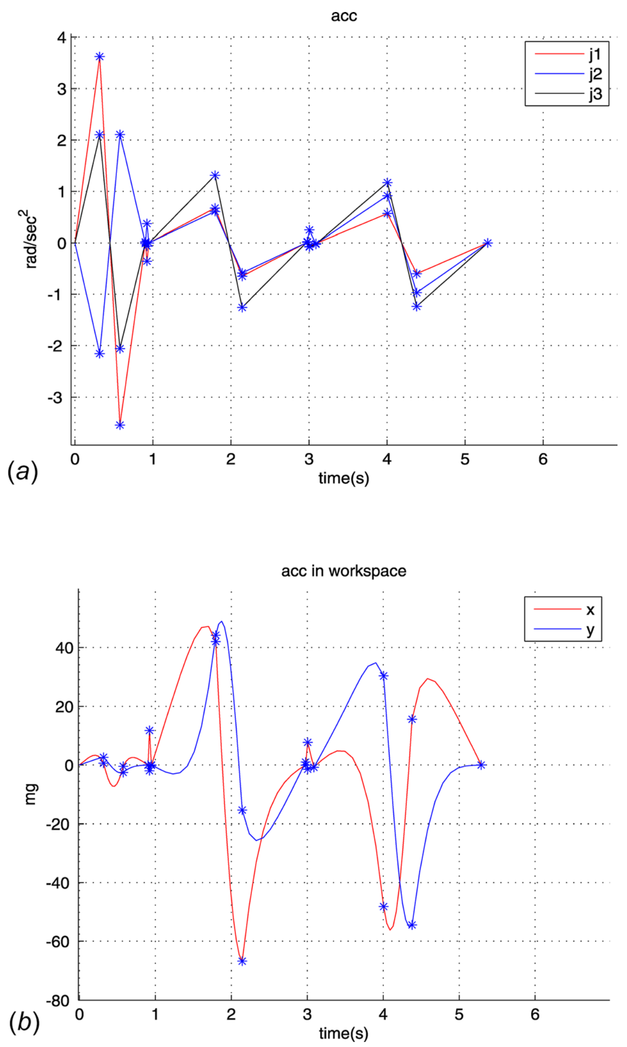


Fig. 13 Acceleration for the optimized trajectory (under constraints in Table 2)

hard to guarantee that the solution is globally optimal. However, given the optimization with inequality constraints and feasible initial point, the interior-point method is guaranteed to output an improved feasible solution.

7 Conclusion

In this paper, a trajectory planning problem is solved for a wafer handling robot that works in the EFEM of a semiconductor manufacturing machine. The generated trajectory is guaranteed to be a safe one without colliding with any obstacles and is optimized with respect to the total execution time for working efficiency. Unlike most existing methods for trajectory planning either partially solved the problem or with heavy computational load, the proposed trajectory planning in this paper solves the problem with consideration of almost all aspects, which involves obstacle avoidance and cycle time optimization under kinematic constraints.

For the obstacle avoidance, first a roadmap describes that the connectivity of the free space is constructed by the PRM method with multiple samplers adaptively updated. Then, a shortest path is searched for the given query, i.e., the initial and goal configuration of the robot. With the path searched from the roadmap, the trajectory is composed by means of a cubic B-spline parametrization, and all kinematic constraints are formulated in this manner.

The problem is formulated as an optimization problem and solved by an interior-point method and the results show the effectiveness of the proposed method.

Future work will be devoted to consider trajectory outside the EFEM, as well as the optimal control problem which involves the robot dynamics. Also, for a real-time application, it is desirable to reduce the computational time with more efficient computing structure.

Acknowledgment

This project has been supported by the Applied Materials.

References

[1] Hsu, D., Sanchez-Ante, G., and Sun, Z., 2005, "Hybrid PRM Sampling With a Cost-Sensitive Adaptive Strategy," Proceedings of the IEEE International Conference on Robotics and Automation, pp. 3874–3880.
[2] Kavralu, L. E., Svestka, P., Latombe, J.-C., and Overmars, M. H., 1996, "Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces," IEEE Trans. Rob. Autom., **12**(4), pp. 566–580.
[3] Kavraki, L. E., Kolountzakis, M. N., and Latombe, J.-C., 1998, "Analysis of Probabilistic Roadmaps for Path Planning," IEEE Trans. Rob. Autom., **14**(1), pp. 166–171.
[4] Ratliff, N., Zucker, M., Bagnell, J. A., and Srinivasa, S., 2009, "CHOMP: Gradient Optimization Techniques for Efficient Motion Planning," IEEE International Conference on Robotics and Automation, pp. 489–494.
[5] Park, C., Pan, J., and Manocha, D., 2012, "ITOMP: Incremental Trajectory Optimization for Real-Time Replanning in Dynamic Environments," International Conference on Automated Planning and Scheduling, Sao Paulo, Brazil, June 25–29.
[6] Hauser, K., and Ng-Thow-Hing, V., 2010, "Fast Smoothing of Manipulator Trajectories Using Optimal Bounded-Acceleration Shortcuts," IEEE International Conference on Robotics and Automation, pp. 2493–2498.
[7] Gasparetto, A., and Zanotto, V., 2007, "A New Method for Smooth Trajectory Planning of Robot Manipulators," Mech. Mach. Theory, **42**(4), pp. 455–471.
[8] Wang, C.-H., and Horng, J.-G., 1990, "Constrained Minimum-Time Path Planning for Robot Manipulators Via Virtual Knots of the Cubic B-Spline Functions," IEEE Trans. Autom. Control, **35**(5), pp. 573–577.
[9] Lepetič, M., Klančar, G., Škrjanc, I., Matko, D., and Potočnik, B., 2003, "Time Optimal Path Planning Considering Acceleration Limits," Rob. Auton. Syst., **45**(3–4), pp. 199–210.
[10] Judd, K. B., and McLain, T. W., 2001, "Spline Based Path Planning for Unmanned Air Vehicles," AIAA Guidance, Navigation, and Control Conference and Exhibit, Guidance, Navigation, and Control and Co-located Conferences, Montreal, QB, Canada, Aug. 6–9.
[11] Cao, B., Doods, G. I., and Irwin, G. W., 1994, "Time-Optimal and Smooth Constrained Path Planning for Robot Manipulators," IEEE International Conference on Robotics and Automation, Vol. 3, pp. 1853–1858.
[12] LaValle, S. M., 2006, *Planning Algorithms*, J. O'Kane, ed., Cambridge University Press, New York.
[13] Bobrow, J. E., Dubowsky, S., and Gibson, J. S., 1985, "Time-Optimal Control of Robotic Manipulators Along Specified Paths," Int. J. Rob. Res., **4**(3), pp. 3–17.
[14] Chen, Y.-C., 1991, "Solving Robot Trajectory Planning Problems With Uniform Cubic B-Splines," Optim. Control Appl. Methods, **12**(4), pp. 247–262.
[15] Zanotto, V., Gasparetto, A., Lanzutti, A., Boscariol, P., and Vidoni, R., 2011, "Experimental Validation of Minimum Time-Jerk Algorithms for Industrial Robots," J. Intell. Rob. Syst., **64**(2), pp. 197–219.
[16] Choset, H., Lynch, K., Hutchinson, S., Kantor, G., Burgard, W., Kavraki, L., and Thrun, S., 2005, *Principles of Robot Motion: Theory, Algorithms, and Implementations (Intelligent Robotics and Autonomous Agents series)*, A Bradford Book, The MIT Press, Cambridge, MA.
[17] Kim, J.-T., and Kim, D., 2014, "Manipulator Motion Planning With Unconstrained End effector Under Obstacle Environment," Adv. Rob., **28**(8), pp. 533–544.
[18] Hwang, Y. K., Watterberg, P. A., Chen, P. C., and Lewis, C. L., 1991, *General Techniques for Constrained Motion Planning*, Educational Testing Service, Princeton, NJ.
[19] Capisani, L. M., Facchinetti, T.,Ferrara, A., and Martinelli, A., 2013, "Obstacle Modelling Oriented to Safe Motion Planning and Control for Planar Rigid Robot Manipulators," J. Intell. Rob. Syst., **71**(2), pp. 159–178.
[20] Gasparetto, A., and Zanotto, V., 2008, "A Technique for Time-Jerk Optimal Planning of Robot Trajectories," Rob. Comput.-Integr. Manuf., **24**(3), pp. 415–426.
[21] Luenberger, D. G., and Ye, Y., 2008, *Linear and Nonlinear Programming*, 3rd ed., Springer-Verlag, US, New York.
[22] Larsen, E., Gottschalk, S., Lin, M. C., and Manocha, D., 1999, "Fast Proximity Queries With Swept Sphere Volumes," Department of Computer Science, UNC Chapel Hill, Technical Report No. TR99-018.