**Adaptive Surface-Conforming Trajectory Planning and Control for a 5-Dof Robot Manipulator**

Sherzod Djuraev1, a), Malika Abdumajidova1, Davronbek Tukhtasinov1, Khushnid Madaliyev1, Dilshodbek Madaminov2

*1Namangan State Technical University, Namangan, Uzbekistan*

*2Urgench State University, Urgench, Uzbekistan*

*a)Corresponding author:* [*sherzoddjurayev1989@gmail.com*](mailto:sherzoddjurayev1989@gmail.com)

**Abstract**. This paper presents a kinematic and control framework for surface-conforming motion of an industrial robot manipulator. The workpiece surface is described either by an explicit height function z=f(x,y,t)z=f(x,y,t)z=f(x,y,t) or by a parametric representation, which allows both static and slowly moving shapes to be modelled in a unified way. A ring of distance sensors mounted around the end-effector provides real-time measurements of the tool–surface gap, and a simple adaptive law updates the desired vertical coordinate so that a prescribed stand-off distance is maintained. The updated Cartesian reference is mapped to joint space through inverse kinematics; for a 3-link arm closed-form solutions are derived, while a Jacobian-based iterative solver is used for a more general 5-DOF (4R+1P) manipulator. A computed-torque control law tracks the resulting joint references. Simulation results demonstrate that the proposed approach produces smooth joint trajectories and keeps the end-effector within the required distance from a non-planar surface, even when the surface geometry or pose vary in time.

**Keywords**. Robot manipulator; surface-conforming trajectory; inverse kinematics; distance sensing; adaptive control; 5-DOF robot; computed-torque control; numerical simulation.

**INTRODUCTION**

Many industrial processes such as spray painting, polishing, grinding, and non-destructive inspection require a robot to move its end-effector along a complex surface while preserving a constant stand-off distance. If the distance is too small, the tool may damage the surface or saturate the process; if it is too large, coating thickness, material removal rate, or sensor resolution may become unacceptable. Conventional trajectory planning methods usually treat the surface as a static geometric object and generate paths offline, without systematically exploiting real-time distance measurements. As a result, modelling errors, deflections, or part misalignment can lead to considerable deviations from the ideal process conditions. These limitations motivate the development of motion generation algorithms that explicitly combine surface models with online sensor feedback.

The kinematics of robot manipulators has been extensively studied, and closed-form solutions exist for many classical architectures. However, when the robot must follow a non-planar or moving surface, the inverse kinematics problem becomes coupled with the geometry of the workpiece and with the constraints on the stand-off distance. In such cases, purely geometric solutions are often insufficient, and it is natural to integrate sensing and control into the trajectory generation process. At the same time, modern manipulators are equipped with torque or current controlled joints, which makes it possible to use model-based dynamic controllers to improve tracking accuracy. Combining these elements in a coherent framework is essential to achieve robust surface-conforming motion. This paper addresses the above issues by proposing a method that links surface modelling, adaptive distance regulation, inverse kinematics, and dynamic control. The surface is described by an analytic function or by a parametric map that may depend on time, thereby covering both stationary and moving workpieces.

Distance sensors mounted around the end-effector provide on-line measurements of the tool–surface gap, and a simple proportional update law modifies the desired vertical coordinate. The resulting Cartesian reference is converted into joint coordinates using either closed-form inverse kinematics or an iterative Jacobian-based solver. A computed-torque controller then tracks the joint reference trajectories. The feasibility of the method is demonstrated in simulation on a 5-DOF manipulator following a non-planar surface.

**METHODS**

The first step in the proposed framework is to model the surface that the robot must follow. For many applications a height function of the form is sufficient, where and denote coordinates in the base frame and is the surface elevation. To incorporate motion or slow deformation of the workpiece, the function is extended to , where is time. More complex shapes can be represented by a parametric map with parameters and . This representation is flexible and can approximate a wide range of surfaces relevant to industrial processing. It also provides analytic expressions for surface normals, which are useful when orientation constraints are imposed on the end-effector [1-4].

The desired in-plane motion is described by functions and , which form the footprint of the trajectory on the surface. A set of distance sensors arranged around the tool measures the instantaneous gap between the endeffector and the surface; the average of these readings is denoted . A simple control objective is to keep this distance equal to a prescribed value , which represents the required stand-off for the technological process. The distance error is defined as . A proportional adaptation law then updates the desired vertical coordinate as

(1)

where is an adaptation gain [5-7].

This produces an updated Cartesian reference that combines the nominal geometric path with real-time sensor feedback.

To actuate the manipulator, the Cartesian reference must be converted into joint coordinates. For a three-link arm with base rotation , shoulder elevation , and elbow angle , explicit inverse kinematics can be obtained from the forward-kinematics equations by geometric reasoning. The base angle is chosen as , aligning the arm with the horizontal projection of the target point. Using the horizontal distance and the vertical offset , the law of cosines yields the elbow angle, and a two-step atan2 expression gives the shoulder angle. For a more general 5-DOF manipulator with four revolute and one prismatic joint, a closed-form solution is less convenient, so an iterative Jacobian-based inverse kinematics algorithm is employed. At each time step the algorithm updates the joint vector until the Cartesian error is below a tolerance. Dynamic tracking of the joint reference is achieved with a computed-torque controller. The robot dynamics are written in standard form , where is the vector of joint torques [8]. Defining the auxiliary control input , the torque command becomes

(2)

Here and are diagonal gain matrices tuned to obtain desired stiffness and damping. The complete algorithm was implemented in a numerical simulation environment for a 5-DOF manipulator and tested on a non-planar surface. At each discrete time step, the algorithm updated the Cartesian reference from sensor data, solved the inverse kinematics, and applied the computed-torque law to track the joint trajectories, while storing joint angles and end-effector positions for later analysis [9-10].

**RESULTS AND DISCUSSION**

In many surface-following tasks the first step is to describe the workpiece mathematically.  
One common option is to use an explicit height function , where each point on the plane defines a unique height . In more general situations a parametric form is convenient:

(3)

where and are abstract surface parameters. The additional argument allows the surface itself to depend on time, which covers cases such as moving conveyors or slowly deforming objects. Expressing the surface in this way gives a compact representation that later makes it easy to impose geometric constraints on the robot motion.

The end-effector reference trajectory is represented by a time-varying position vector .

Its Cartesian components are chosen as

(4)

where and describe the footprint of the path on the surface. Physically, these functions may correspond to the desired painting line, inspection scan, or polishing pattern. The vertical component is then selected so that the tool stays at a prescribed normal distance from the surface. At this stage the reference motion is purely geometric: it does not yet react to sensor errors or modelling inaccuracies.

To keep the tool at the correct standoff distance, range sensors are mounted around the end-effector. They measure the instantaneous spacing between the tool tip and the surface, and this measured value is denoted . The control system compares this measurement with the desired distance and forms the error

(5)

If the robot comes too close to the surface, the error becomes negative; if it moves too far away, the error is positive. This simple scalar error already contains all the information needed to adapt the vertical coordinate of the planned trajectory.

A convenient adaptation rule is to shift the desired height along the surface normal direction. For a given the nominal surface height is , and the uncorrected offset would be .  
To compensate for the measured error we modify this value as

(6)

where is an adaptation gain. A larger gain gives stronger and faster corrections but may amplify sensor noise or cause oscillations. Collecting the coordinates, the updated reference position is

(7)

which now depends both on the desired geometric path and on real-time sensor readings.

Robots are actuated in joint space, so the Cartesian reference must be converted into joint variables. This conversion is done by the inverse kinematics map, written symbolically as

(8)

where denotes the desired orientation of the end-effector. For redundant or complex robots the inverse kinematics generally has multiple solutions, so an additional criterion such as joint-limit avoidance or minimal motion is usually applied. In practice the inverse kinematics may be solved analytically for simple arms, or numerically using iterative Jacobian methods for more complicated geometries. The resulting vector plays the role of a moving set-point for the low-level joint controllers. When torque or current control is available, a common choice is a computed-torque control law. The robot dynamics are described by the standard rigid-body model with inertia matrix , Coriolis and centrifugal term , and gravity vector . The control torque is then defined as

(9)

Here and are positive-definite gain matrices shaping stiffness and damping in joint space. Under standard assumptions this structure makes the tracking error behave like a set of decoupled secondorder systems, so the end-effector follows the updated surface-conforming trajectory with good transient performance.

To illustrate the ideas, consider a simple planar surface. Let the plane be described by

(10)

where and represent the surface slopes and defines its vertical offset. Even though the surface is simple, this model already covers many practical objects such as flat panels mounted at an angle. If the robot must keep the tool at a constant normal distance from this plane, the vertical coordinate must be chosen accordingly. The requirement is that when the tool moves along the path, its tip never deviates from the prescribed offset.

Assume the desired in-plane motion is given by functions and , for example a straight line or a closed contour. Then the nominal plane height along this path equals . Maintaining the constant gap gives the Cartesian reference

(11)

This expression explicitly links the surface geometry and the robot motion: any change in the plane parameters or in the footprint trajectory immediately affects the desired pose. Such an analytic expression is useful both for simulation studies and for designing real-time control algorithms.

Now consider a spatial manipulator whose reachable workspace can be approximated by three main joint variables. Let be a base rotation (yaw), a shoulder elevation, and an elbow angle, with corresponding link lengths and . The base is elevated by a constant height above the ground frame. A typical forward-kinematics model for such an arm is

(12)

These equations express the end-effector position as a function of joint angles and link dimensions, and they form the basis for the inverse-kinematics derivation.

To obtain the joint variables for a given desired Cartesian point , we proceed step by step. The base angle is chosen to align the arm horizontally with the target,

(13)

Next we define the planar reach

(14)

and the desired horizontal distance

(15)

Setting and considering the triangle formed by , and the distance from the shoulder joint to the target in the sagittal plane leads to

(15)

Using the law of cosines we compute the elbow angle

(16)

and finally the shoulder angle

(17)

These analytic expressions are valuable because they avoid iterative numerical solvers for this class of manipulators.

In many industrial applications the work surface itself is not static.For example, parts may move on a conveyor, or the robot may follow a slowly oscillating platform. To capture such behaviour we extend the plane model to be time dependent,

(18)

Here the functions , and can encode time-varying slopes or vertical drifts. This generalisation allows the same kinematic framework to be applied to both stationary and moving objects without changing the structure of the control laws.

When the surface varies with time, the distance-keeping condition must be satisfied at every instant. Given the in-plane reference , the updated vertical coordinate becomes

(19)

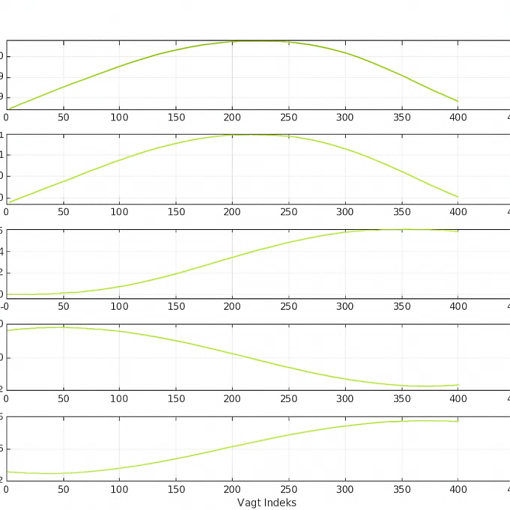
Thus the Cartesian reference trajectory is

(20)

The inverse kinematics is evaluated for this time-dependent pose, leading to joint trajectories that automatically incorporate the motion of the surface. This formulation naturally fits within a real-time control loop, where surface parameters are updated from sensors or from a higher-level planner.

To validate the proposed approach, a five-degree-of-freedom robot manipulator was simulated.

The arm consists of four revolute joints and one prismatic joint, which together provide sufficient dexterity to follow a curved surface. A desired path lying on a non-flat surface was specified in Cartesian space, and the constant standoff distance was imposed. At every time step the trajectory update rule produced a corrected reference position using the current sensor readings. The inverse kinematics map then generated the corresponding joint vector, denoted , by means of an iterative Jacobian-based solver.



**FIGURE 1**. Motion trajectory of each joint of a 5-link robot manipulator

The simulation produced several informative plots and visualisations. The three-dimensional curve traced by the end-effector over the surface was rendered from different viewing angles, clearly showing how the tool conformed to the uneven geometry. A separate set of graphs displayed the time histories of the five joint variables, illustrating the coordinated motion required to maintain the correct distance. The comparison between the nominal geometric path and the sensor-corrected path confirmed that the adaptation law successfully compensated for deviations in the measured surface. Furthermore, the resulting joint trajectories remained smooth, indicating that the chosen gains did not excite unwanted oscillations.

These results demonstrate that the method is suitable for practical implementation in tasks such as spray painting, grinding, or inspection over complex surfaces.

These results demonstrate that the method is suitable for practical implementation in tasks such as spray painting, grinding, or inspection over complex surfaces.

Overall, the study highlights the benefits of combining surface models, sensor feedback, inverse kinematics, and dynamic control in a unified framework. The mathematical description of the surface provides a compact way to encode geometric requirements, while the distance error bridges the gap between theory and real measurements. Inverse kinematics translates the Cartesian requirements into physically meaningful joint commands, and the computed-torque law ensures that these commands are followed accurately. The numerical experiments with the 5-DOF robot confirm that a constant-gap trajectory can be maintained even when the surface is non-planar or slowly varying. Consequently, the proposed method offers both a solid theoretical basis and a practical tool for designing surface-conforming trajectories in robot manipulators.

**CONCLUSION**

This paper has presented a unified approach to generating and tracking surface-conforming trajectories for robot manipulators using a combination of analytic surface models, distance sensing, inverse kinematics, and model-based control. The surface is represented in a compact mathematical form that can accommodate both stationary and slowly moving workpieces. A simple proportional adaptation law uses real-time distance measurements to maintain a constant stand-off distance between the end-effector and the surface. The updated Cartesian reference is mapped into joint space with either closed-form inverse kinematics or a Jacobian-based iterative solver, and a computed-torque controller ensures accurate tracking of the resulting joint trajectories. Simulation of a 5-DOF manipulator following a non-planar surface shows that the method yields smooth joint motion and keeps the tool within the required distance range.

The proposed framework can serve as a basis for a wide range of surface-processing applications, including painting, polishing, and inspection. Because of its modular structure, the method can be extended by incorporating more advanced distance-sensor fusion, adaptive gains, or optimization-based planners for the in-plane trajectory. Future work may also include experimental validation on physical robot platforms, robustness analysis under model uncertainty and sensor noise, and integration with real-time quality monitoring of the technological process. Overall, the results suggest that combining geometric modelling with online sensing and dynamic control is an effective strategy for precise and robust surface-conforming robot motion.

**REFERENCES**

1. Siciliano, B., Sciavicco, L., Villani, L., & Oriolo, G. (2010). *Robotics: Modelling, Planning and Control*. London: Springer.
2. Spong, M. W., Hutchinson, S., & Vidyasagar, M. (2006). *Robot Modeling and Control*. Hoboken, NJ: Wiley.
3. Gulyamov, G., Dadamirzaev, M.G., Boidedaev, S.R. (2000). Hot carrier electromotive force caused by surface potential modulation in a strong microwave field. *Semiconductors, 34, 555-557*. <https://doi.org/10.1134/1.1188027>
4. Craig, J. J. (2005). *Introduction to Robotics: Mechanics and Control* (3rd ed.). Upper Saddle River, NJ: Prentice Hall.
5. De Schutter, J., & Van Brussel, H. (1988). Compliant robot motion I: A formalism for specifying compliant motion tasks. *The International Journal of Robotics Research, 7(4)*, 3–17.
6. Ahmetoglu (Afrailov), M., Kaynak, G., Shamirzaev, S., Gulyamov, G., Gulyamov, A., Dadamirzaev, M.G., Boydedayev, S.R., Aprailov, N. (2009). To the theory of electromotive force generated in potential barrier at ultrahigh frequency field. *International Journal of Modern Physics B*, *23(15)*, 3279-3285. <https://doi.org/10.1142/S0217979209053084>
7. Chiaverini, S., Siciliano, B., & Egeland, O. (1994). Review of the damped least-squares inverse kinematics with experiments on an industrial robot manipulator. *IEEE Transactions on Control Systems Technology, 2(2)*, 123–134.
8. Djuraev, S., & Tursunov, A. (2025). Effect of particle size and concentration on multicyclone device efficiency. AIP Conference Proceedings, 3304, 030049. <https://doi.org/10.1063/5.0269110>
9. Mirzabayev, B., Djuraev, S., & Korabayev, S. (2025). Creating an adaptive control system for increasing the efficiency of the linter machine. *AIP Conference Proceedings, 3304(1)*, 030063. <https://doi.org/10.1063/5.0269114>
10. Gulyamov, G., Dadamirzaev, M.G., Boidedaev, S.R. (2000). Kinetics of attaining the steady state of hot carrier thermoelectric power in a p-n junction with consideration for lattice heating. *Semiconductors, 34,* 260-263. <https://doi.org/10.1134/1.1187967>