Constraints-based Motion Planning for an Automatic, Flexible Laser Scanning Robotized Platform

Th. Borangiu, A. Dogar, A. Dumitrache
University Politehnica of Bucharest, borangiu@cimr.pub.ro, dogar@cimr.pub.ro
a1ex_r0@yahoo.com

Abstract- **The paper addresses an implementation issue of a robotized reverse engineering platform for not modelled objects. The RE platform is composed of three major components: a 6 d.o.f. industrial robot, a dual-camera laser scanner and a digitally controlled turntable on which the object is placed. One important objective of this work is to develop suitable scanning strategies and robot motion patterns for automatic sensor guiding and acquisition of 3D surface data of objects to be modelled. The proposed strategy implies the synchronization of the robot movements with the rotation of the turntable, in order to avoid collisions or axis out of range and to keep, if possible, the robot in the central area of its workspace. For each unique picture taking point, the adequate turntable angle and robot configuration will be computed. A 6 d.o.f vertical articulated robot simulator was developed to generate the set of necessary robot configurations and turntable angles corresponding to the scanning strategies. The simulator is used to test whether a scanning path is feasible from the kinematics point of view. The scanning strategy consists of two stages: first, predefined scanning paths are followed, based on offline planned motion patterns; secondly, evaluation of the acquired 3D data may eventually lead to new scanning paths with different orientations of the hand-held laser scanner to obtain complete 3D information of the hidden details of the object's surface.**

I. INTRODUCTION

The described work is part of the designing and implementing of reverse engineering platform. The platform consists in a 6. d.o.f robot arm, dual camera laser scanner device, a turntable and a 5 axis CNC milling machine. The laser scanning device mounted on the robot arm probe is able to measure distances from 70 to 250 millimetres, with an accuracy of achieving 30 µm. The robotic arm moves around the work piece being scanned by computer-generated adaptive scanning paths. The scanning device is a class-2, short distance, triangulation one, and has two CMOS sensors allowing the scanning of complex object surfaces. The optimal scanning distances range from 71 mm to 242 mm. The width of the scanning line varies between 31 mm and 83 mm, and the average measuring precision at point level is 31 µm.

The acquisition rate is between 50 and 150 frames per second, the number of points which are read on a scanning line being 480. The laser range finder system is interfaced to a 3.2 GHz IBM PC-type station by means of a standard USB input port, and uses additionally a digital RS485 line for synchronization with the robot controller.

The usual 'teach and play' method used in industry to generate work data for robots is not suitable in this case. The proposed scanning strategies consist of two stages. In the first stage a predefined motion pattern is followed based on the chosen scanning strategy. There are considered three types of object classes that are to be modelled and for each scanning strategies are developed. Thus, in the first stage of the scanning process, an off-line programming method is used to generate motion data for the robot. One of the difficulties to overcome in order to generate functional OLP data, is Collision-Free Path Planning, which creates safe robot paths so that there are no collisions between robot and obstacles.

II. REVERSE ENGINEERING PLATFORM SIMULATOR

The reverse engineering platform simulator is designed to be a development tool and test bench for the developed adaptive scanning algorithms. The three main components of the platform are simulated. For the robotic arm, the simulator allows displaying the robot in any user-defined position. The robot simulator is comprised of two modules. The static simulation module allows specifying the robot pose, using one of the three input methods:

- specify the angular values for each joint and for the rotary table (direct kinematics)
- specify the position in Cartesian coordinates and the orientation in ZYZ' Euler angles, in the robot's reference frame (inverse kinematics with respect to robot)
- specify the angle of the rotary table, and the position and the orientation of the robot end point in the rotary table's reference frame (inverse kinematics with respect to rotary table)

If the specified pose with respect to rotary table cannot be reached, the table is rotated automatically until the requested pose can be reached.

The motion simulation module lets the user simulate and analyze the behaviour of the robot using a sequence of user-defined trajectories. The user interface has an editor for the motion sequence, and controls for generating the animation.

The simulator can use two modes for computing inverse kinematics:

- Internal inverse kinematics routine, using Peter Corke's Robotics Toolbox for Matlab, Release 7.1 [1]
- The inverse kinematics routine from an Adept robot controller, which is invoked using a TCP/IP connection.

1-4244-2577-8/08/$20.00 ©2008 IEEE

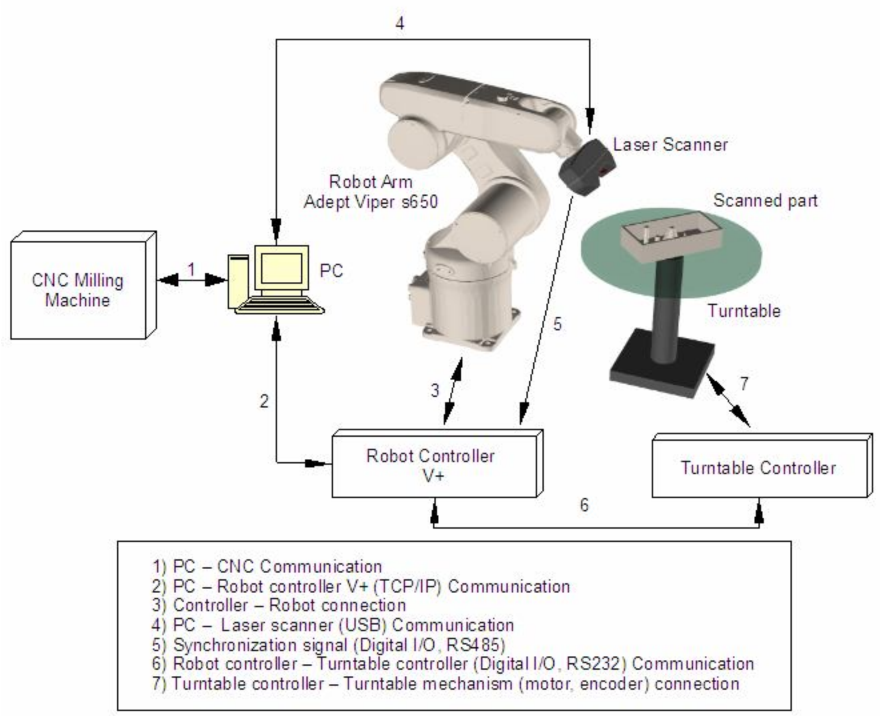


Figure 1. Hardware architecture of the laser scanner-robot-CNC machine platform for reverse engineering tasks

The first method, which uses a generic inverse kinematics based on the pseudo-inverse of the manipulator Jacobian [1], have the advantage that it can run on a standalone computer, and it executes fast for most situations where a solution for inverse kinematics does exist. However, if a solution cannot be found, the routine hangs for a few seconds before reporting the lack of convergence. Another disadvantage is that this routine is more likely to find a solution outside the joint limits of the robot, when the initial robot pose is far from the destination.

The second method is more robust with respect to initial pose of the robot, can be configured to compute solutions with desired configurations of the robot (i.e. LEFTY, RIGHTY, ELBOW UP, ELBOW DOWN, FLIP, NOFLIP), and runs fast even if there is no solution for inverse kinematics, although it has an overhead from TCP communications. When it is called directly from V^+, without the network overhead, this method is very fast, being able to perform about 500 inverse kinematics computations per second.

The motions can be defined as trajectories that interpolate between some positions, specified as transformations. A transformation contains six values: the Cartesian coordinates for the position, and the yaw, pitch and roll angles for the orientation. The points may be defined with respect to two predefined reference frames: the robot reference frame, which is fixed, and the rotary table reference frame, which changes every time the rotary table moves.

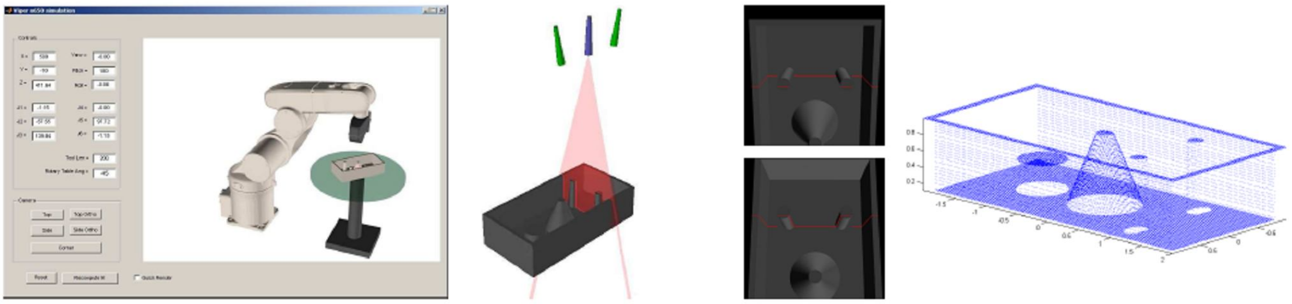
The user has two choices for interpolation between positions: either straight line, or joint interpolated motion. These modes correspond to the MOVES and MOVE instructions from the V^+ language.

When the motions are executed in the rotary table reference frame, the interpolation is also performed in the rotary table reference frame. This means that, for a straight line motion, for any possible trajectory for the rotary table, the motion of the end effector of the robot will be always be a line with the respect to the table, but with respect to the robot base, the motion may have a complex shape. This situation is handled by the motion planner for the rotary table

As for the rotary table, its rotation is simulated, and the workpiece that sits on the table should rotate synchronously with the table. Behaviours such as inertial movement due to fast table movements do not have to be simulated; the workpiece should be considered attached to the table.

For the laser probe, the software should simulate the interaction of the laser beam with any user-defined workpiece, having various surface properties. The two cameras which are integrated into the laser probe should also be simulated, and the image which would be captured by them should be displayed to the user. Furthermore, the laser beam should be detected in the images from the two cameras, and the triangulation equations should be applied to them in order to compute a point cloud. The point clouds obtained from simulating the scanning process from different viewing angles should be transformed into a fixed coordinate system that will be attached to the workpiece being scanned.

In the second stage of the scanning process the acquired data is used to generate new scanning paths, whether to obtain a high resolution detail or to manage occlusions.



a) Screen shot of the simulator b) Schematic view of laser probe c) Images from the two cameras d) Computed point cloud

Figure 2. Laser scanner - robot arm system simulator

III. DEFINED CONSTRAINTS IN THE ROBOT CONFIGURATION SPACE

The defined constraints in the robot system are of two types: hard constraints and soft constraints. The hard constraints must be kept at each step of the motion planning in order to ensure a valid path. The hard constraints consist in: known obstacles in the robot workspace, articulated robot singularities and articulated robot joint angle limits. The defined soft constraints are surface avoiding (keeping the minimum allowed scanning distance towards the modelled object), flexible reach (avoiding “un-comfortable” positions of the robot arm) and following the computed path.

A very important aspect of the implementation consists in avoiding the existent obstacles in the robot working area. These predefined obstacles are: the turntable, the object to be modelled, the conveyor and the CNC machine.

Using the utility SPEC of the V+ operating system, one can define up to 4 Cartesian obstacles and clearance distances. There must be defined parameters used to avoid collisions with the existent static obstacles in the workspace. The path of the robot tool tip is automatically tested to ensure that it does not collide with defined obstacles under the following circumstances: when the robot is being moved in WORLD or TOOL manual control mode; when the destination of each motion is being planned; and while straight-line motions are being performed.

Each obstacle is defined by its shape, location, and size. The shape of an obstacle can be a box, a cylinder, or a sphere. The location of each obstacle is defined with respect to the base reference frame of the robot when its BASE transformation is null.

The turntable is defined as a cylinder with the height equal with the turntable height. Since the dimensions and shape of the modelled object are not a priori known, the proposed strategy defines as a robot obstacle a semi sphere with the radius equal to the minimum scanning distance allowed by the scanning device plus minimum height of the scanned object. Because the collision detection references the tool point of the end-effector, which is typically at the centre of the tooling, there have been added 1/2 of the diameter of the robot's end-of-arm tooling to each surface of the existing obstacle being modelled.

A configuration singularity can be defined as a location in the robot workspace where two or more joints no longer independently control the position and orientation of the tool. As a robot executes a straight-line motion that moves close to a configuration singularity, the robot joint speeds necessary to achieve that motion become excessive. The types of configuration singularities that can be experienced by a robot depend on the physical relationships between the robot joints. The configuration singularities are as follows:

- Wrist singularities: occurs when the axes of Joints 4 and 6 are aligned;
- Alignment singularities: occurs when Joint 6 (wrist) and Joint 1 axes are aligned;
- Elbow singularities: occurs when the arm is fully extended. In this case, as the elbow joint becomes further extended, higher joint speeds are required to maintain constant Cartesian speed. The robot cannot extend beyond its reach.

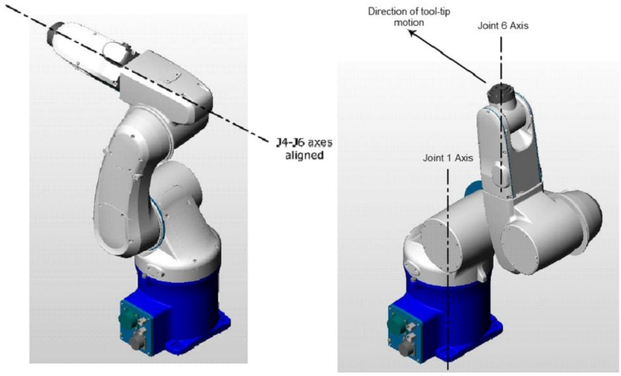


Figure 3 Configuration singularities (wrist and alignment singularities)

Due to the configuration of the 3D scanning platform and the characteristics of the robot motions, the alignment and elbow singularities are very improbable to occur. The developed algorithm solves the wrist singularity problem.

For a more general solution, the singular configurations will be avoided by using the below defined measure of the manipulator's *dexterity*

$$\mathrm{dex}(\mathbf{q}) = [\det(\mathbf{J}^T(\mathbf{q})\mathbf{J}(\mathbf{q}))]^{1/2}, \; n \le 6. \qquad (1)$$

Hence a manipulator is at a joint-space singularity $\mathbf{q}_s$ if $\mathrm{dex}(\mathbf{q}_s) = 0$ or, more generally, near it when $\lim_{\mathbf{q}\to\mathbf{q}_s} \mathrm{dex}(\mathbf{q}_s) = 0$.

IV. CONSTRAINT-BASED MOTION PLANNING

The path planning problem is a NP-complete problem. The existing algorithms for this problem are of two major categories: classic and heuristic approaches. The current developed classic methods are variations of a few general approaches: *Roadmap*, *Cell Decomposition*, *Potential fields* and *mathematical programming*. The majority of motion planning problems can be solved using these approaches, which are not mutually exclusive and a combination of them is often used. In the **Roadmap** approach, the free C-space, i.e., the set of feasible motions, is retracted, reduced to, or mapped onto a network of one-dimensional lines. This approach is also called the *Retraction*, *Skeleton*, or *Highway* approach. The search for a solution is limited to the network, and MP becomes a graph-searching problem. The well-known roadmaps are Visibility graph, Voronoi diagram, Silhouette, and the Subgoal Network. In **Cell Decomposition (CD)** Algorithm, the free C-space is decomposed into a set of simple cells, and the adjacency relationships among the cells are computed. A collision-free path between the start and the goal configuration of the robot is found by first identifying the two cells containing the start and the goal and then connecting them with a sequence of connected cells. The **Potential Fields (PF)** concept was first introduced by Oussama Khatib. A robot in Potential Fields method is treated as a point represented in configuration space as a particle under the influence of an artificial potential field *U* whose local variations reflect the 'structure' of the free space. The potential function can be defined over free space as the sum of an attractive potential, pulling the robot toward the goal configuration, and a repulsive potential pushing the robot away from the obstacles. The **Mathematical programming approach** represents the requirement of obstacle avoidance with a set of inequalities on the configuration parameters. MP is formulated then as a mathematical optimization problem that finds a curve between the start and goal configurations minimizing a certain scalar quantity.

The classic approaches have the some drawback such as high complexity in high dimensions and trapping in local minima.In order to improve the efficiency of Classic methods, probabilistic algorithms have been developed; including **Probabilistic Roadmaps (PRM)** and **Rapidly- exploring Random Trees (RRT)**, with major advantages is high-speed implementation. Also other approaches exist in RMP such as **Level set** and **Linguistic Geometry** (LG). To fix the local minima problem, many **Heuristic** and Mete-heuristic algorithms are used in RMP. For example, a combination of the **Simulated Annealing (SA)** technique and PF remedies this problem. Other approaches include Artificial Neural Network (ANN), Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Ant Colony (ACO), Stigmergy, Wavelet Theory, Fuzzy Logic (FL) and Tabu Search (TS). Heuristic algorithms do not guarantee to find a solution, but if they do, are likely to do so much faster than deterministic methods [5].

For the constraint-based motion planning of the scanning robot a novel algorithm was developed. The motion planner must generate valid scanning paths in both stages of the 3D data acquisition. In the first stage of the scanning process, based on the developed motion patterns dedicated to each class of objects and the hard and soft constraints, the collision-free flexible motion path is generated. The hard and soft constraints are included in a general cost function of the motion planner [3].

The turntable is seen as a seventh degree of freedom added to the 6-DOF robot arm. The turntable is necessary since the robot should be able to analyze the workpiece from various directions (e.g. front, sides, top, back), and not all these orientations can be reached without moving the workpiece. The motion planning problem for the turntable is continuously finding an angle for the seventh joint (the table) such as the end effector of the robot should be able to reach the desired pose for scanning and that pose to be comfortable. The turntable motion planner will therefore perform the computations after the scanning trajectories have been defined. (Fig. 4)

Input data for this problem consists of the scanning toolpaths, which are a series of locations (positions and orientations), in the workpiece's reference frame. The scanning system has to move continuously and synchronously the rotary table and the robot arm, such as the laser probe achieves the programmed locations and be able to take the measurements.

Output data is a sequence of joint values of the robotic arm and the rotary table angle, which give the desired location of the laser probe with respect to the workpiece. In other words, the problem is the inverse kinematics for a 7-DOF mechanism.

In addition, the computed solution has to satisfy the following requirements:

- minimize the accelerations and limit the speed of the rotary table;
- avoid collisions with any obstacles which may be within the manipulator's range, or between the manipulator and the rotary table.

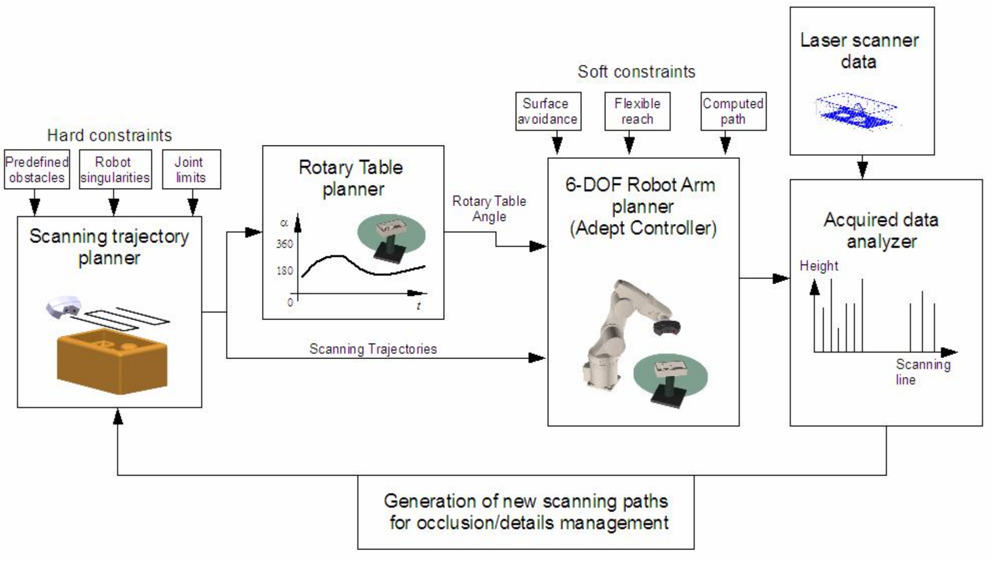


Fig. 4 Constraint-based motion planning in the two stages of the scanning process.

ACKNOWLEDGEMENTS

This work is funded by the National Council for Scientific University Research, in the framework of the National Plan for Research, Development and Innovation, grant 69/2007.

REFERENCES

[1] P. I. Corke, *A Robotics Toolbox for MATLAB*, IEEE Robotics and Automation Magazine, vol. 3, no. 1, mar. 1996, pp.24-32

[2] Latombe, J.C., *Robot Motion Planning*, Kluwer Academic Publishers, 1993.

[3] Kim. J.H. and et al., Collision Avoidance Planning of Jointed Robots using Configuration Space, *Korean Society of Automotive Engineering*, Vol. 2, No. 6, 1994.

[4] Makita, Y.; Hagiwara, M. and Nakagawa, M.; “A simple path planning system using fuzzy rules and a potential field”, Proc. IEEE World Cong. Comput. Intel.,(1994) pp. 994-999.

[5] Masehian E., Sedighizadeh D.,Classic and Heuristic Approaches in Robot Motion Planning – A Chronological Review, Proceedings of World Academy of Science, Engineering and Technology Volume 23 August 2007 ISSN 1307-6884