A Human - Machine Interface for Teleoperation of Arm Manipulators in a Complex Environment

I. Ivanisevic and V. Lumelsky
University of Wisconsin-Madison
Madison, Wisconsin 53706, USA

Abstract

This paper discusses the feasibility of using *configuration space* (C-space) as a means of visualization and control in operator-guided real-time motion of a robot arm manipulator. The motivation is to improve performance of the human operator in tasks involving the manipulator motion in an environment with obstacles. Unlike some other motion planning tasks, operators are known to make expensive mistakes in such tasks, even in a simpler two-dimensional case. Using an example of a two-dimensional arm manipulator, we show that translating the problem into C-space improves the operator performance remarkably, on the order of magnitude compared to the usual work space control.

1 Introduction

The goal in this project is to improve the performance of human operators in tasks that involve motion planning and control of complex objects in environments with obstacles. Human performance in such tasks is known to be patently inferior. Our focus is on developing a visual computer interface that would allow the operator to visualize and perform the work in the task *configuration space* (C-space) rather than in the *work space* (W-space) as usually done. To make it feasible, a computer intelligence is provided that works alongside with human intelligence in real time. The intent of this work is to be applicable to many existing research [1] and commercial problems [2, 3].

There is a large and rapidly developing class of technical systems that are dependent on human contribution for their operation. In various teleoperated systems (such as in space, nuclear reactors, chemical cleanup sites, underwater probes) human operators plan and guide the motion of remotely situated devices through interaction with computer displays or three-dimensional models of the device. Familiar examples include control of the NASA Shuttle arm and of the Titanic exploration probe. In such tasks operators are known to make mistakes of overlooking collisions with surrounding objects; this results in expensive repairs and limits the system effectiveness. People seem to be unable to navigate and manipulate remote equipment without colliding with objects in the environment.

Similar problems occur in other settings. Guiding the position of a robotic welding gun or spray painting device with a simultaneous translation and orientation adjustment seems to be particularly difficult for people, even when visual feedback is provided. Performance is very poor in a variety of these movement planning tasks when time is not a constraint (the Shuttle arm, for example); it becomes progressively worse in real-time operation, in three-dimensional (3D) vs 2D tasks, and when system dynamics are involved (masses, inertia etc.).

Experiments with human subjects [4, 5] suggest that the problem is in the peculiarities of human spatial reasoning: humans have difficulty handling simultaneous interaction with objects at multiple points of the device's body, or motion that involves mechanical joints (such as in arm manipulators), or dynamic tasks. Learning and practice improve the performance rather little. Furthermore, the performance pattern is the same when operating a physical rig or performing the task on a computer screen and moving the arm links with a mouse (see more on this in Section 4). On the other hand, these experiments confirm the expected fact that in a maze-searching problem, if information is provided about the whole maze (a bird's-eye view),

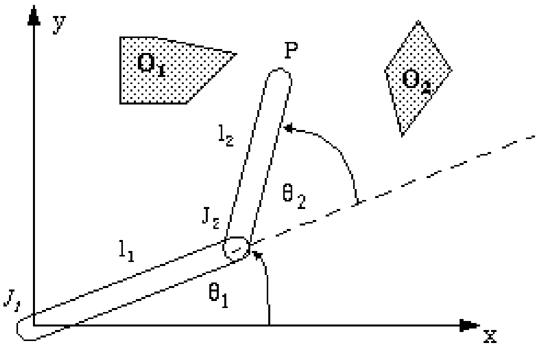

Figure 1: The 2D two-link arm manipulator in an environment with stationary obstacles.

human performance is well above the fastest computer with the best known algorithms [6].

This contrast in the subjects' performance in the two tasks above poses a question as to whether a human-machine interface, perhaps with adequate machine intelligence, can be developed to improve human performance is such applications. The current work is an attempt to answer this question. The system we chose to model the problem is a two-dimensional (2D) revolute-revolute (RR) arm manipulator operating in an environment with stationary obstacles (see Figure 1). The arm has two links moving in a plane, and two revolute joints (degrees of freedom). The idea it to present the problem to the human as one of moving a point in a maze (a task that humans are good at) rather than the actual problem of moving a jointed kinematic structure (which humans are not good at).

Below, the properties of work space control are discussed in Section 2, and those of the configuration space – in Section 3. Experimental results and discussion appear in Section 4.

2 Work Space Control

The revolute-revolute (RR) planar arm considered is as follows, Figure 1: Joint J_1 (the shoulder) is attached to the floor, and is the origin of a fixed reference system. Joint J_2 (the elbow) connects the two links, l_1 and l_2. The Cartesian coordinates of the endpoint (point P) are (x, y). Moving the arm involves changing the joint angles θ_1 and θ_2. There are fixed obstacles in the arm environment (O_1 and O_2, Figure 1). There are no constraints on the shape of the obstacles or the arm links. The task is to move the arm from a position S (Start) to the position T (Target), Figure 2.

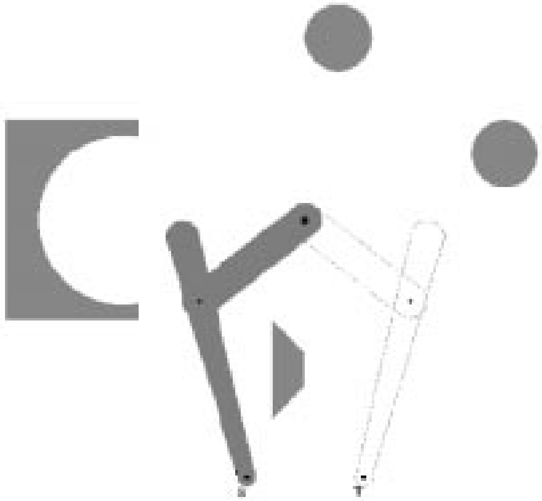

Figure 2: A sample task.

2.1 Motion Control in W-space

Arm motion is controlled with the computer mouse, in two separate modes - joint-mode and tip-mode [7]. The former allows control of individual joints while the latter controls the position of the endpoint. In the joint-mode, the algorithm computes a unit vector which describes the straight-line direction from current configuration to a specified target configuration. Assuming the target configuration does not violate step constraints (if the distance to it is larger than the selected step size, a new target is computed by multiplying the direction vector by step size), the new configuration becomes the specified configuration. In tip-mode, the direction vector describes the new position of the arm endpoint (again, subject to step constraints), and so one needs to recover the new arm configuration from the endpoint position (x, y). This is done via the usual inverse kinematics equations.

The last step in either motion mode is to determine if the new configuration would place the arm in contact with the obstacle and, if so, disallow the movement and wait for further operator input. Figure 3 shows an example of average human performance in W-space motion control; the dotted line is the trajectory of the arm endpoint along the way from S to T. The path length is the

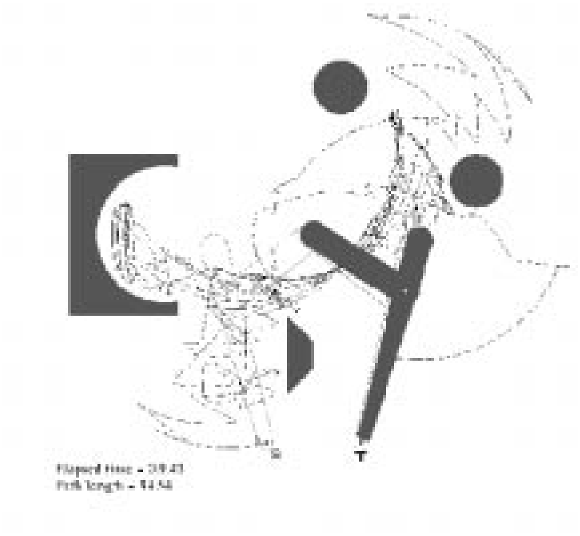

Figure 3: An example of average human performance in W-space motion control.

integral of changes in both joint angles along the way. [One may find this performance suprizingly inferior].

2.2 Characteristics of the Work Space Control

Aside from being the traditional method used, W-space control has some desirable properties:

– Interaction with the physical arm and its environment makes it easier for the operator to visualize the global navigation, such as to determine the next target configuration based on some scene property; e.g. the operator may decide to move the arm such that the left side of link l_2 will be in proximity of some object.

– If the obstacles layout is not of much constraint on the arm motion, this approach can yield very good (near optimal in terms of path length and time taken) results.

However, this type of control also has some serious drawbacks which may outweigh its positive sides:

– In tip-mode, calculating the inverse kinematics becomes progressively more complex and time-consuming as the number of joints increases.

– In a complex environment, the operator may have hard time determining which direction of local motion is better, or whether a given direction leads to a "deadend". This is a serious drawback: for example, in Figure 3 one can pass the topmost obstacle with the elbow to the left or to the right of the obstacle; one of those turns out to be wrong as it leades to a deadend, and this would become clear only significantly later.

– From the standpoint of motion planning, a complex environment is not necessarily one with many or with large obstacles; this is much clearer in C-space (see Section 3) than in W-space.

Consequently, W-space control is likely to produce redundant motion: as illustrated in Figure 3, the operator will often try, backtrack, try again, backtrack again, and so on until the passage is found, not rarely through blind luck. This also endangers the arm, as all such motion multiplies potential collisions with surrounding objects. While most people do benefit from a training period in such systems, the improvement is marginal [5].

3 Configuration Space Control

The arm from Section 2 can be defined in terms of the shoulder angle θ_1 and the elbow angle θ_2. The set of all configurations (θ_1, θ_2), Figure 1, define the arm's *configuration space* (C-space), which can be represented as the surface of a common two-torus. An arm configuration in W-space corresponds to a point in C-space. This mapping preserves continuity; small change in W-space position corresponds to a small change in the C-space position. A geodesic line between two points on the torus (a straight line in the plane (θ_1, θ_2)) is the "shortest path" between the points: four such paths can actually appear [6].

3.1 Motion Control in C-space

We will now attempt to control the arm motion indirectly, via its point image in C-space (C-point). Each time the operator moves the C-point slightly, the algorithm recovers a new set of configuration variables (θ_1, θ_2) from the C-point coordinates and automatically translates it into the actual motion in W-space. That is, after the direction vector is calculated and step size is taken into account, similar to the joint-mode in W-space control, angles θ_1 and θ_2 become available, and they are used to control the arm's next step. Though not necessary for control purposes, for convenience a W-space window with the arm real-time motion is shown next to the C-space window used by the operator. Certain applications, e.g. grasping, may require knowledge of the Cartesian position (x, y) of the arm endpoint. If necessary, (x, y) values can be recovered from C-space information via the direct kinematics equations.

We are now one step away from converting the complex problem of W-space control to a simpler problem of navigating a point in the maze (C-space). What is missing is the maze itself. This is done by computing the *C-space obstacles*, also called *virtual obstacles*. Each point of a virtual obstacle corresponds to an arm configuration that is not attainable because of interference with the corresponding physical obstacle. The related (x, y) positions in W-space may or may not be occupied by an obstacle - in the latter case such pieces of an obstacle are called its *shadows*. A finite number of obstacles in W-space produce a finite set of virtual obstacles in C-space. The boundaries of virtual obstacles are known to consist of simple closed curves [6]. Since virtual obstacles are defined in terms of arm variables (θ_1, θ_2), their shape is visually unrelated to the shape of the W-space obstacles [7, 8].

3.2 Construction of C-space Obstacles

The greatest improvement in the operator performance comes when full information (the bird's-eye view) about C-space is available (on the issue of operating with uncertainty, see the discussion in Section 4). We thus need to compute and display all the virtual obstacles. One such algorithm was proposed in [8]. A simpler approach (used by our simulation as described in [7]) is to employ a variation of the Bug1 [6] algorithm.

Figure 4 gives the C-space representation of W-space of Figure 2. Angle θ_1 is along the horizontal axis, θ_2 is along the vertical axis. The range of change of each angle is 2π, making C-space a square. The dark areas represent the virtual obstacles. The C-space correspondence to a two-torus means that all four corners of the square are identified (i.e. correspond to the same point). Similarly, the top and bottom edges of the square are identified, and so are the left and right edges. Point T is chosen as the corner point of the C-space square; in principle, therefore, one's moving from point S to any corner will produce a legitimate (if not necessarily the shortest) path for the arm in W-space. The significance of the dotted line (path) is explained in Section 4.

3.3 Characteristics of the Configuration Space Control

This mode of control has several distinct advantages (see also Results, Section 4):

– From the operator standpoint, the task is simplified greatly: instead of dealing with a complex jointed kinematic structure, the operator has to solve a simple maze-searching problem with complete information, which humans are very good at.

– The arm's actual motion is quickly and easily calculable from user input, guaranteeing good real-time performance.

– Unlike in W-space, performance here does not seem to depend much on the obstacle layout. Indeed, this mode has consistently yielded near optimal performances by the human operator in a variety of settings. This is consistent with the fact that humans can easily "see" the path in a bird-eye view of a fairly complex maze, while they have difficulty visualizing a path in a simple scene with an arm manipulator (see Figure 2). The operator easily discards many "deadend" directions in the maze representation, but find it difficult to identify them in Figure 2.

– The mode requires very little training, mostly to get used to the peculiarities of flat presentation of two-torus - e.g. to the fact that once the point reaches the top edge of the C-space square, it appears at the bottom edge.

– Unlike the W-space control, the subject can often easily see if a solution (a path) exists. In fact, it is this kind of decision-making that the operator uses extensively along the way to discard potential dead-ends.

A few drawbacks deserve to be noted of this mode, although their impact is not nearly as great as those in W-space control:

– The fact of dealing with an abstract (C-) rather than physical (W-) space may make it difficult for the operator to address some global navigation tasks, such as choosing targets for the arm to reach. This problem is easily avoided if the corresponding W-space view is drawn in parallel with the C-space used by the operator.

– Computation of C-space is an expensive operation which must be performed to satisfy the complete information model (see Section 4 for details on the proposed uncertainty model).

4 Results and Discussion

Results. Overall, the proposed C-space control mode performed admirably when compared to the traditional W-space control. Current results show improvement in performance on the order of magnitude when switching from W-space control to the proposed C-space control. The path produced approaches the optimal (shortest) path and time to complete the task. This remarkable fact puts the human operator ahead of the existing computer algorithms, contrary to the W-space control where human performance has been much worse.

Table 1 summarizes information from a series of controlled experiments performed in 1996-97 at the UW Robotics Lab, to test human performance in motion planning tasks. One of the tasks given to the human subjects was to move a two-link arm, very similar to the one considered in this paper, from the start to target configurations. Only W-space control was available (Section 2). In the table, the path length is the integral of both joint angle changes during the motion; also given is the time (in seconds) taken by subjects to complete the task. The data given represents the performance of 12 subjects on the second day of tests, after training and practice on the previous day. (The results on 48 untrained subjects, in tests with a simulated as well as physical arm manipulator, were quite similar). A full analysis of this work can be found in [5].

Table 1: Descriptive Statistics

	Mean	Min.	Max.	St.Dev
path len.	129.04	15.13	393.90	107.99
time, sec	504.83	90.00	900.00	365.89

No similar study was carried out for the C-space control mode. However, based on the observations and tests by these authors, the study is not necessary: the performance improvement is very clear and consistent. Further, it is clear that in the task of Figure 2 different subjects are likely to produce almost the same (nearly optimal) path, with the mean path length of about 12, the standard deviation of about zero, and the mean time below 60 sec. The path length and time values in Table 2 show an order of magnitude improvement compared to the data on W-space control in Table 1. Sample results from 5 consecutive runs of C-space control are given in Table 2. One (typical) run is shown in Figure 4.

Table 2: Sample Runs

	Run1	Run2	Run3	Run4	Run5
path l.	12.67	12.39	12.24	12.27	12.28
time, s	56	54	53	53	54

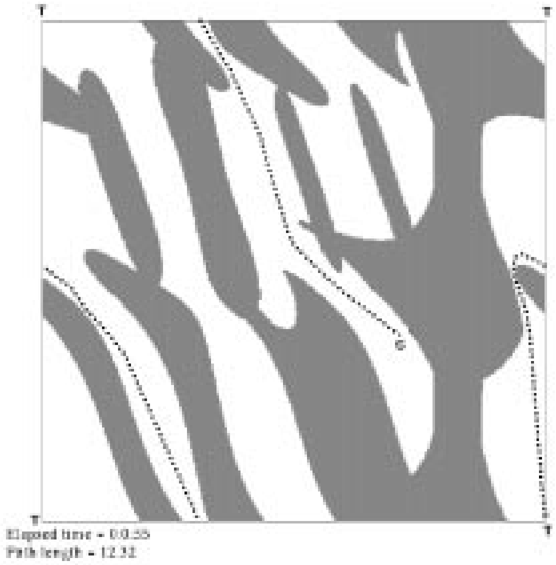

Figure 4: The sample task of Fig. 2: C-space motion control. The corresponding W-space in Fig. 5.

The consistency between these runs - both in path length and completion time - is very similar to the subjects performance in a common maze-searching problem. It also stands in contrast to the wide range of results produced in the W-space model. This suggests that the proposed transformation to C-space control does indeed make the task at hand similar to the maze-searching task.

Discussion. This paper proposes an approach to human-guided teleoperation of a robot arm manipulator based on the configuration space (C-space) rather than on the common work space (W-space) control. Instead of directly confronting the problem of collision analysis, which is known to be extremely challenging for the human spatial reasoning, the task is offered to the operator in C-space where one can concentrate on global navigation, leaving collision analysis to the computer. Thus reduced task becomes a maze-searching problem in which humans are known to be very good. Designing such a system takes, first, calculation of the C-space, and second, an adequate user interface.

While this approach can be immediately useful

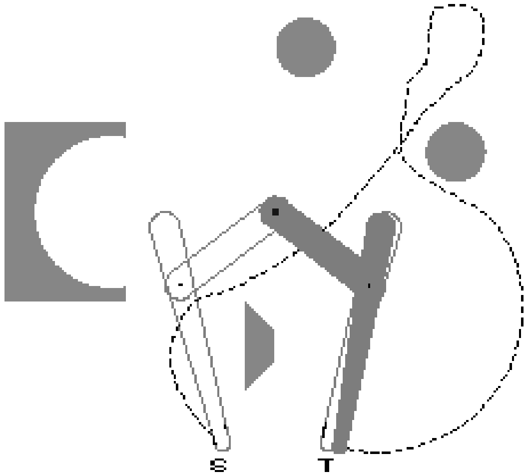

Figure 5: The W-space view of the task in Fig. 4. The path produced does not contain unnecessary "detours" common to W-space control (see Fig 3), and approaches optimal path for this task.

even in its two-dimensional version described, in order to become a truly universal tool it needs to be extended to the three-dimensional case and to more degrees of freedom. The advantage for the operator of dealing with a point rather than a complex jointed kinematic structure is obvious. The challenge is to produce an adequate user interface (specifically, develop ways of visualizing and guiding a point in a higher-dimensional space) and to do C-space calculation and collision analysis fast enough to keep the operator active at the control station. One possibility here is to help the operator handle the environment with incomplete, rather than complete, information; this would mean a significant reduction in the C-space computation costs. Success in this area will also mean applicability of the approach to a dynamic environment with moving obstacles. Computer algorithms for motion planning with incomplete information are available (e.g. [6]). Experiments with human subjects operating in an unknown maze [4, 5] suggest that humans might be able to handle this case as well.

References

[1] Q. Lin, C. Kuo, Virtual Teleoperation of Underwater Robots, *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, April 1997.

[2] P. Schenker et al., Development of a telemanipulator for dexterity enhanced microsurgery, *Proceedings of the 2nd International Symposium on Medical Robotics and Computer Assisted Surgery*, pp. 81-88, 1995.

[3] I. Hunter et al., A Teleoperated Microsurgical Robot and Associated Virtual Environment for Eye Surgery, *Presence*, Vol. 2, pp. 265-280, Fall 1993.

[4] V. Lumelsky, S. Rogers, J. Watson, F. Liu, An Experimental Study of Human Performance in Planning Object Motion. Final Report. University of Wisconsin Robotics Lab. July 1997.

[5] F. Liu, Multivariate Analysis of Human Performance in Motion Planning. M.S. Thesis, University of Wisconsin-Madison, Mechanical Engineering, May 1997.

[6] V. Lumelsky, Algorithmic and Complexity Issues of Robot Motion in an Uncertain Environment, *Journal of Complexity*, No.3, 1987.

[7] T. Skewis, V. Lumelsky, Simulation of Sensor-Based Robot and Human Motion Planning, *International Journal of Robotics and Automation*, Vol.8, No. 4.

[8] T. Lozano-Perez, A Simple Motion-Planning Algorithm for General Robot Manipulators, IEEE Journal of Robotics and Automation, Vol. RA-3, No. 3, pp. 224-238, 1987.