Structured Beam Projection for Semi-automatic Teleoperation

Young S. Park
Argonne National Laboratory
9700 S. Cass Ave., Argonne, IL 60439

ABSTRACT

This paper presents a robotic architecture that facilitates semi-automatic teleoperation of a dual-arm manipulator system. The architecture is built upon reactive behavior agents tightly coupling sensing and action, where emergent behaviors correlate human intervention with sensor-based autonomous operation. Reactive agents are devised for perceptual and motor behaviors, and a structured light system is adopted to provide a visual reference to both behavioral agents and the human operator. The technical feasibility of the robotic behaviors is evaluated through simulation of the collaborative motion of a dual-arm manipulator. The presented sensor based semi-automatic teleoperation may improve human perception and simplify human action, and thus improve the efficiency and precision of teleoperation.

Keywords: structured light, dual-arm manipulator, robotic architecture, reactive behavioral agent, teleoperation

1. INTRODUCTION

Telerobots are practical alternatives to autonomous robots for operations in unstructured or unpredictable environments. Teleoperation can effectively leverage the human intelligence by having human operators directly operate the slave manipulator through an input mechanism. However, teleoperation is an extremely slow process and mostly imprecise. This has become evident in a previous experience of deploying a teleoperated dual-arm manipulator system for dismantling a nuclear reactor at Argonne National Laboratory. It was observed that most of the robot operation time was spent in aligning the tools rather than actual tooling operations. Furthermore, due to the difficulty in manually commanding the two arms' simultaneous motions through teleoperation, the full task capability of the dual-arm system was never realized. What is required is an effective aid that can guide the human operator through a set of semi-automatic operations. Automating tasks in an unstructured environment, however, suffers from the difficulty of assuring reliability. In this regard, technological innovations should be made with the following considerations.

Most challenges in implementing autonomous agents in unstructured environments lies in perceiving environmental conditions using external sensors, such as cameras. Fortunately, in teleoperation, such high level reasoning is left to the human, and the task environment is often not entirely unstructured. In such a case, sensors are required mostly for low-level perception, rather than for high-level interpretation. Therefore, sensors structured for a specific purpose is preferred over those conveying general environmental information. For this purpose, a structured light system provides a simple and effective external sensing capability, because it simplifies the computational process and does not require controlled lighting. Using a grid of light and looking for the distortions in the grid pattern, it may also be possible to characterize location, orientation, and surface details. It may also provide an effective visual reference for the human operator^[1].

Robot control strategy can be designed based on either deliberative reasoning or reactive behaviors. A deliberative robotic control system is characterized by a hierarchical structure for reasoning, where higher levels in the hierarchy create sub-goals for lower levels. It assumes that all tasks can be decomposed into well-defined subtasks, and the resulting actions are predictable due to the clear and restrictive flow of control and communication between levels of the hierarchy. Hierarchical control is seemingly well suited for structured and highly predictable environments. However, it lacks responsiveness in unstructured and uncertain environments due to the requirement and limited communication pathways. Furthermore, it has certain disadvantages in deployment because it is difficult to engineer a complete system with incremental competency. That is, the entire system needs to be built before testing is feasible. On the other hand, a reactive system is ideal in the case where the drawbacks of a deliberate system are limiting^[2]. Instead of having a fixed structural hierarchy, a reactive control system is composed of a collection of behaviors that tightly link sensory inputs to motor actions. The flow of control and communication between each behavior is less restrictive, and autonomous control emerges from the interaction of multiple behaviors. A reactive system is more suitable for autonomy in unstructured and uncertain environment. Due to its incremental competency, more complex behaviors can be built incrementally from elementary behaviors.

The robot system subject to this work is a dual-arm manipulator. By having two arms work in cooperation, a dual-arm manipulator system provides a unique opportunity for performing dexterous task unachievable by a single manipulator. However, dual-arm collaborative motion is difficult to accomplish via teleoperation because it requires spatial coordination

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, make any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

beyond human intuition, and it involves motion planning in higher dimensions. Therefore, it is necessary to provide the computational means to relieve the human operator from complex kinematic reasoning.

To this end, this paper presents a robotic architecture that facilitates semi-automatic teleoperation of dual-arm manipulator system taking into account the above factors. The following elements are incorporated:

- 1) Introduce a structured light system to improve environmental comprehension,
- 2) Base robotic architecture on reactive behavior agents, which facilitates a bottom-up approach to building reliable task capabilities,
- 3) Incorporate the kinematic equation of dual-arm motion into behavioral coordination of reactive behaviors.

2. KINEMATIC EQUATION OF DUAL-ARM MANIPULATOR

The dual-arm system, shown in Figure 1, consists of two hydraulically operated Titan III manipulators by Schilling Robotics Co. Each manipulator has 6 d.o.f. of mobility, not including the gripper motion. A dual-arm manipulator system's potential capabilities may effectively be exploited by treating them as a single redundant manipulator and describing the collaborative motions as relative trajectories between the two manipulators' end-effectors. The kinematic equation governing their collaborative motion can be derived as follows^[5]:

Relative location of a link j with respect to link i of a manipulator is defined by a Denavit-Hartenberg homogeneous transformation matrix of the following form:

$${}^iA_j = \begin{bmatrix} {}^i n_j & {}^i o_j & {}^i a_j & {}^i p_j \\ 0 & 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} {}^i R_j & {}^i p_j \\ 0 & 1 \end{bmatrix}$$

where ${}^i n_j$, ${}^i o_j$, ${}^i a_j$, and ${}^i p_j$ denotes the principal orientation vectors and the position vector of link j with respect to the coordinate system of link i . Since the dual-arm manipulator system consists of two 6 d.o.f. manipulators, the relative location of the second end-effector, e_2 , with respect to coordinate frame of the first end-effector, e_1 , is defined as follows:

$$\begin{aligned} {}^{e_1}A_{e_2} &= [({}^6A_{e_2})^{-1} ({}^5A_{e_2})^{-1}] ({}^4A_{e_2})^{-1} \dots ({}^0A_{e_2})^{-1} [({}^B A_{20})^{-1} {}^B A_{10}] {}^0A_{11} \dots {}^5A_{16} {}^6A_{1e_1} \\ &= {}^{12}A_{11} {}^{11}A_{10} \dots {}^2A_1 {}^1A_0 {}^6A_{1e_1} \end{aligned}$$

where iA_j and iA_2 denotes the link transform from link i to link j of manipulator 1 and 2, respectively, and B is the base coordinate system fixed to the platform base. With the introduction of a relative transformation, the dual-arm manipulator can be considered to be a 12 d.o.f. redundant manipulator. To facilitate its control scheme, it is necessary to compute the relative Jacobian. According to (Whitney [5]), such a relative Jacobian can be computed as

Figure 1. Dual-arm Manipulator System

RECEIVED
NOV 08 2000
OSTI

$$J(\theta) = \begin{bmatrix} {}^{12}a_{12} \times {}^{12}p_0 & {}^{12}a_{11} \times {}^{11}p_0 & \dots & {}^{12}a_2 \times {}^2p_0 & {}^{12}a_1 \times {}^1p_0 \\ {}^{12}a_{12} & {}^{12}a_{11} & \dots & {}^{12}a_2 & {}^{12}a_1 \end{bmatrix} = [J_{12} \ J_{11} \ \dots \ J_7 \ J_1 \ J_2 \ \dots \ J_6]$$

where ${}^{12}z_j$ is the direction of the j^{th} joint axis with respect to the 2^{nd} end-effector coordinate frame (i.e., the third column vector of the link transformation ${}^{12}A_j$). Then, the relative Jacobian, $J_r(\theta)$, is obtained by pre-multiplying $J(\theta)$ by the 3×3 rotational matrix

$$J_r(\theta) = \begin{bmatrix} {}^0R_{12} & 0 \\ 0 & {}^0R_{12} \end{bmatrix} [J_1 \ J_2 \ \dots \ J_6 \ J_7 \ \dots \ J_{11} \ J_{12}]$$

where ${}^0R_{12}$ is the 3×3 rotational submatrix of ${}^0A_{12}$. The relative Jacobian relates the composite joint velocities, $\dot{\theta}$, to the relative Cartesian velocities, $\dot{x}$, of the end-effectors of the two manipulators as

$$\dot{x} = J_r(\theta) \dot{\theta}.$$

Here, θ , is a composite joint angle vector defined as $\theta = [\theta_{11} \ \theta_{12} \ \dots \ \theta_{16} \ \theta_{21} \ \theta_{22} \ \dots \ \theta_{26}]$, where θ_{ij} denotes the j^{th} joint angle of the i^{th} manipulator. For this redundant manipulator, the inverse kinematic equation is given by

$$\dot{\theta}_r = J_r^+ \dot{x}_r + \dot{\theta}_h$$

where the first term is the particular solution which produces the motion required to track a given task trajectory $\dot{x}_r$, and the second term, $\dot{\theta}_h$, is the homogeneous solution in the null space of J_r . This homogeneous solution can be utilized to induce beneficial self-motions, such as obstacle avoidance, using the null-space projection technique commonly used with redundant manipulator systems. To facilitate beneficial motion, $\dot{x}_h$, the kinematic equation is represented as

$$\dot{\theta}_r = J_r^+ \dot{x}_r + [J_c (I_6 - J_r^+ J_r)]^+ (\dot{x}_h - J_c J_r^+ \dot{x}_r)$$

where J^+ denotes the pseudo inverse of a matrix J , and defined by $J^+ = J^T (JJ^T)^{-1}$, and J_c is a composite Jacobian that relates the joint rates to the desired beneficial motions, $\dot{x}_h$. The term, $[J_c (I_6 - J_r^+ J_r)]^+$, represents the projection onto the null space of J_r . In this paper, we intend to implement reactive behavior agents to facilitate various robotic motions required for D&D tasks and other the beneficial motions.

3. ROBOTIC ARCHITECTURE

According to Stone [7], a robot architecture is defined as "the discipline devoted to the design of highly specific and individual robots from a collection of common software building blocks." According to this definition, architecture refers to software architecture, rather than the hardware side of the system. In this section, we introduce a new robotic architecture to correlate human intervention with sensor-based autonomous operation.

From our experience of deploying robotic equipment at a research nuclear reactor, common D&D processes are composed of a wide variety of strategic tasks, such as cutting pipes and bolts, tearing down graphite blocks, cutting reactor walls, unscrewing bolts, carrying waste baskets and loads, inspection, and others. By analyzing the nature of these tasks, it can be noted that various strategic tasks emerge from a few common behaviors. As shown in Figure 2, a schema-based robotic architecture^{[2][3]} is composed from assemblages of motor-behavioral agents. Each motor agent is characterized by tightly linking sensory input to motor actions. Motor behaviors are relatively large grain abstractions reusable over a wide range of circumstances, and many of the behaviors have internal parameters that provide additional flexibility in their deployment. Embedded within each motor schema is a perceptual schema that provides environmental information specific to that particular behavior. The low-level perceptual processes can also be merged into a higher-level interpretation before passing it on to the motor behavior. The output motion commands of each motor schema behaviors are then coordinated through weighted vector summations after they are classified into the respective dual-arm motion components.

Figure 2. Structure of Behavior-Based Robotic Architecture

3.1. Motor Behaviors

Taking the environmental information from the perceptual schema, the motor schema has as output a motion $\dot{\mathbf{x}}$, which is composed of linear velocity, $\mathbf{v} = [v_x, v_y, v_z]$, and rotational velocity, $\Omega = [\omega_x, \omega_y, \omega_z]$. Therefore, the relative motion of the dual-arm manipulator, $\dot{\mathbf{x}}_r$, is given as

$$\dot{\mathbf{x}}_r = \begin{bmatrix} \mathbf{v}_r \\ \Omega_r \end{bmatrix}.$$

In this work, the reactive agents embody the following motor behaviors:

Teleautonomy, $\dot{\mathbf{x}}_{tele}$, allows the human operator to provide internal bias to the control system. This behavior is encoded as

$$\mathbf{v}_r = k_v \mathbf{v}_m \text{ and } \Omega_r = k_\Omega \Omega_m$$

where $\mathbf{v}_m$ and $\mathbf{w}_m$ denotes the motion command given by the human operator, and k_v and k_Ω are the motion scale factors between master input devices and slave manipulators.

Approach-goal, $\dot{\mathbf{x}}_{goal}$, permits the robot to move toward a specified goal location, deaccelerating at close proximity to the goal location. This is encoded as

$$\mathbf{v}_r = V_{mag} \cdot \frac{\mathbf{p}_g - \mathbf{p}}{|\mathbf{p}_g - \mathbf{p}|} \quad \text{where} \quad V_{mag} = \begin{cases} V_{max} & \text{for } d > S \\ V_{max} \cdot \frac{d}{S} & \text{for } d \leq S \end{cases}$$

and d is the distance to the goal location, and S denotes a sphere of influence.

Align-tool, $\dot{\mathbf{x}}_{align}$, places and orients the tool in the direction normal to the surface of a part to be dismantled. This is encoded as

$$\Omega_r = W_{mag} \cdot (\mathbf{n} \times \mathbf{n}_d + \mathbf{o} \times \mathbf{o}_d + \mathbf{a} \times \mathbf{a}_d) \quad \text{where} \quad W_{mag} = \begin{cases} 0 & \text{for } d > S \\ W_{max} \cdot \frac{d}{S} & \text{for } d \leq S \end{cases}$$

Grasp-tool, $\dot{\mathbf{x}}_{grasp}$, closes the gripper when an object is detected between the jaws.

Dock, $\dot{\mathbf{x}}_{dock}$, slowly moves the tool in the direction normal to the part surface until it is pressed against the part.

Move-along-path, $\dot{\mathbf{x}}_{path}$, moves the tool along a path generated from the interpretive perceptual schema, **define-tool-path**. Therefore, $\mathbf{v}_r = \mathbf{v}_{path}$ and $\Omega_r = \Omega_{path}$, where $\mathbf{v}_{path}$ and Ω_{path} are the planned tool trajectory.

Maintain-attitude, $\dot{\mathbf{x}}_{attitude}$, provide bias to the path following motion to maintain offset distance away from or into the surface.

$$\mathbf{v}_r = V_{mag} \cdot \mathbf{n}_{path} \quad \text{where} \quad V_{mag} = \begin{cases} K & \text{for } d > D + S \\ W_{max} \cdot \frac{d - D}{S} & \text{for } d \leq D + S \end{cases}$$

where D is the desired offset distance.

Escape, $\dot{\mathbf{x}}_{escape}$, moves the tool in backward biased random directions to escape from a jammed condition. This is encoded

$$\mathbf{v}_r = V_1 \cdot \mathbf{e}_{random} - V_2 \cdot \frac{\mathbf{v}_{path}}{|\mathbf{v}_{path}|}$$

$$\omega_r = W \cdot \mathbf{e}_{random}$$

where V_1 and V_2 and W are velocity gains, and $\mathbf{e}_{random}$ is a unit vector generated in random directions.

Absolute position constraint, $\dot{\mathbf{x}}_{abs}$, deliberately provides absolute constraint on the position of either one or both manipulators. This behavior is useful for tasks such as fixturing a part with one arm, while the other arm applies a tool to the part. This behavior is encoded as

$$\dot{\mathbf{x}}_h = \begin{bmatrix} \mathbf{x}_d - \mathbf{x} \\ 1/2(\mathbf{n} \times \mathbf{n}_d + \mathbf{o} \times \mathbf{o}_d + \mathbf{a} \times \mathbf{a}_d) \end{bmatrix}$$

where $\mathbf{x}_d$ and $\mathbf{x}$ are the desired and the present location of the end-effector, respectively. Also, $\mathbf{n}$, $\mathbf{o}$, $\mathbf{a}$ denotes the orientation vectors.

Avoid-obstacle, $\dot{\mathbf{x}}_{obs}$, provide beneficiary motion to move the manipulator links away from the approaching obstacles. This behavior is encoded by the following self-motion:

$$\dot{\mathbf{x}}_h = \begin{cases} 0 & \text{for } d > S \\ -\frac{\mathbf{n}}{d} & \text{for } R \leq d \leq S \end{cases}$$

where $\mathbf{n}$ is the vector normal to the obstacle, and d is distance from the link to the obstacle.

Avoid-joint-limit, $\dot{\mathbf{x}}_{j_limit}$, provide beneficiary motion to prevent approaching the joint limits of the manipulators.

3.2. Behavioral coordination

Our experience in robotic applications in a nuclear facility revealed that most D&D tasks can be represented as a sequence of a few characteristic tasks. Such high-level behaviors can be assembled from the primitive reactive behaviors. In this work, behavioral assemblages are composed of the following abstract tasks, as shown in Figure 3:

Transport a tool: Human operator carries the tool, held by one or two arms, from one location to a goal location marked by a structured beam projection. Reactive agents assist human effort by providing bias force to avoid collision with obstacles, and to direct and align the tool to the goal location.

Acquire a tool: Human operator provides gross motion of moving the end-effector to the vicinity of the tool. Then reactive agents act to automatically approach, grasp, and retract the tool, guided by the grid beam projection. In this mode, tasks are shared sequentially with human and autonomous agents.

Apply a tool: Tooling generally requires precise motion in a rather structured environment, and thus can benefit from automating the process. Based on a priori knowledge of the task geometry and sensory information, the task trajectory is generated and updated, and reactive agents provide motions to track the trajectory, with task-specific motion constraints.

The behavioral assemblage is defined as a coordinated collection of primitive behaviors or assemblages. As explained previously, the relative motion of a dual-arm manipulator can be constituted from primary motion, $\dot{x}_r$, and secondary motion, $\dot{x}_h$, such that

$$\dot{\theta}_r = J_r^+ \dot{x}_r + P_{J_r} (\dot{x}_h - J_c J_r^+ \dot{x}_r)$$

where P_{J_r} is the projection onto the null space of J_r . The output motion commands of each motor behavior are coordinated through weighted vector summations after they are classified into the respective dual-arm motion components. For the primary motion, the behavioral coordination is realized by the following vector sum:

Figure 3. Behavioral Assemblages

$$\dot{\mathbf{x}}_r = \dot{\mathbf{x}}_{tele} + \dot{\mathbf{x}}_{goal} + \dot{\mathbf{x}}_{align} + \dot{\mathbf{x}}_{grasp} + \dot{\mathbf{x}}_{dock} + \dot{\mathbf{x}}_{path} + \dot{\mathbf{x}}_{attitude} + \dot{\mathbf{x}}_{escape}.$$

To incorporate the secondary self-motion, it is necessary to define the compound motion vector, $\dot{\mathbf{x}}_h$, composed of the individual motion components and the compound Jacobian, $\mathbf{J}_c$, to relate the beneficial motions into joint velocities. Therefore, the secondary motion is defined as

$$\dot{\mathbf{x}}_h = \begin{bmatrix} \dot{\mathbf{x}}_{abs} \\ \dot{\mathbf{x}}_{obs} \\ \dot{\mathbf{x}}_{j_limit} \end{bmatrix} \quad \text{and} \quad \mathbf{J}_c = \begin{bmatrix} \mathbf{J}_{abs} \\ \mathbf{J}_{obs} \\ \mathbf{J}_{j_lim} \end{bmatrix}.$$

Here $\mathbf{J}_{abs} = [\mathbf{J}_{abs,1} \ \mathbf{J}_{abs,2}]$ is the absolute motion Jacobian comprised of two 6×12 matrices, $\mathbf{J}_{abs,1} = [[\mathbf{J}_{m1}][0_6]]$ and $\mathbf{J}_{abs,2} = [[0_6][\mathbf{J}_{m2}]]$, where $\mathbf{J}_{m1}$ and $\mathbf{J}_{m2}$ are the Jacobians for the first and second manipulators. The obstacle avoidance Jacobian, $\mathbf{J}_{obs} = [\mathbf{J}_{obs,11} \ \dots \ \mathbf{J}_{abs,16} \ \mathbf{J}_{abs,21} \ \dots \ \mathbf{J}_{abs,16}]$ is comprised of 3×12 Jacobians, $\mathbf{J}_{obs,ij}$, relating the motion of j^{th} link of i^{th} manipulator to joint velocity. For instance, $\mathbf{J}_{obs,23}$ is represented as $\mathbf{J}_{obj,23} = [0 \ 0 \ \dots \ 0 \ | \ v_{21} \ | \ v_{22} \ | \ v_{23} \ | \ 0 \ 0 \ 0]$, where v_{21} is the linear velocity component of the j^{th} column of the i^{th} manipulator Jacobian. The joint limit Jacobian, $\mathbf{J}_{j_lim} = [\mathbf{J}_{j1} \ \dots \ \mathbf{J}_{j12}]$ is composed of 1×12 Jacobian matrices, $\mathbf{J}_i$, defined for joint i such that all its elements are zeros except that the i^{th} element is 1.

4. PERCEPTUAL SYSTEM

In our implementation, an external sensor consisting of a structured light system and manipulator internal-position sensors are adopted to provide raw sensory data to the perceptual schema.

4.1. Structured light system

In this work, vision is adopted as the primary perception. To facilitate conducting quantitative scene analysis, a structured laser beam projector is added to the camera system. The projected beam pattern over environmental objects can provide a visual reference to control agents as well as the human operator. As shown in Figure 4, the camera and laser light source are assumed mounted on the robot's end-effector. The structured light beam is projected parallel to the wrist plane of the manipulator. The camera is mounted above the light source and tilted down. If the projected light strikes an object in the field of view of the camera, a pattern of line segments are formed on the object. The shape, size, and orientation of the

projected grid pattern are analyzed to identify 'what' and 'where' the parts are, as implemented in the following perceptual agents.

4.2. Perceptual agent

Perception is conducted on a need-to-know basis: individual perceptual algorithms provide the information immediately related to the behavior which decides an action. The low level perceptual processes may also be merged into a higher-level interpretation before it is passed on to the motor behavior. The following perceptual agents are defined in our implementation:

Find distance: As the robot gripper moves closer to the object, the grid center point moves down in the camera's field. Therefore, the vertical distance from the bottom of the image frame to the grid center point represents the distance to the part, as illustrated in Figure 5. A calibration chart is constructed experimentally to quantify this relationship.

Find size: The size of an object is determined by measuring the pixel distance between the line end points and measuring it against the horizontal axis of the calibration chart.

Find-normal-direction: The orientation of a part's surface is measured from the tilt angles of the horizontal and vertical grid lines, as illustrated in Figure 5. Then the surface normal direction is given as

$$a_s = [-S\phi C\alpha \quad S\alpha \quad -C\phi C\alpha]^T.$$

The pitch, ϕ , and yaw, α , angles completely characterize the surface normal vector, a_s , and a calibration chart is constructed to relate them to the slopes of the beam lines.

Find-orientation: This routine determines the orientation, ψ , of the part with respect to the approach direction of the robot hand. This angle is determined from the angle between the part's edge line and the beam line. This also completes the relative rotation matrix, R_s , of the part represented in the hand coordinate system,

$$R_s = [n_s \quad o_s \quad a_s] = \begin{bmatrix} C\phi C\psi + S\phi S\alpha S\psi & -C\phi S\psi + S\psi S\alpha C\psi & S\phi C\alpha \\ C\alpha S\psi & C\alpha C\psi & -S\alpha \\ -S\phi C\psi + C\phi S\alpha S\psi & S\phi S\psi + C\phi S\alpha C\psi & C\phi C\alpha \end{bmatrix}$$

where n_s , o_s , and a_s are the orthonormal directional vectors.

Find shape: The shape of the part, such as a cylindrical or rectangular block, is determined from the shape of the beam lines. As shown in Figure 5, cylindrical parts result in curved beam lines, while a rectangular region results in straight lines.

Define-tool-path: From the shape and size information, the motion bias is generated along the optical line designating the tool path in such a way as to help complete the necessary tooling. This requires compound perceptual strategies.

Figure 5. Measurement from Beam Projection

Check-tool-error: Abnormal tool operating conditions are detected using internal sensors, such as current sensors or force/torque sensors.

Detect-obstacle: Internal joint-position sensors monitor the relative locations of the links of the other arm, and transfer the distance data if it within proximity.

Check-joint-limit: Internal joint-position sensors report the joint angles if they approach the joint limits.

5. SIMULATION AND RESULTS

A simulation study was conducted to test the feasibility of the proposed robotic architecture for use as an aid in the teleoperation of a dual-arm manipulator. The chosen task was to use dual-arm collaboration to perform the process of cutting a large pipe. The workpiece is held by one arm and the other arm applies the circular saw 180° around the circumference of the workpiece. Since the workpiece moves with the arm motion, it is difficult to accomplish such dual-arm motion through teleoperation, hence the proposed behavior-based robotic architecture is used to aid the operator. Thus, instead of specifying the individual arm motions, the operator designates the tool trajectory as relative motion. For the circular duct cutting process, this operator command input, incorporated as the motor behavior *teleautonomy*, is given as a straight-line trajectory (a) in Figure 6. Then, the trajectory is modified by the actions of the behavioral motor agents based on the relative location and part geometry, inputs that the structured-light camera system adequately provides the perceptual agents. Trajectory (b) in Figure 6 shows the tool motion trajectory modified after the action of the motor behavior *move_along_path*, which takes in to account the circular geometry of the workpiece. The task trajectory is further modified to allow and maintain the offset of the saw blade onto the workpiece. This is accomplished by the action of the motor behavior *maintain_attitude*. The resulting trajectory is shown as (c) in Figure 6. Figure 7 shows the motions of the two manipulators making the relative motion of the circular trajectory. As noted from the results, the agent-based robotic architecture can effectively aid human operator by introducing partial autonomy in various processes. Furthermore, it supports incrementally building complex behaviors from the bottom-up, producing a working system at each level of complexity.

Figure 6. Relative Tool Trajectories

6. SUMMARY AND CONCLUSION

For aiding in teleoperation of a dual-arm system, a structured light system is introduced as a perceptual device for the behavior-based robotic architecture. By mounting it on the robot hand, this simple and robust vision system directly provides the relative location and shape of external objects, without requiring the level of computations needed for image processing or reasoning.

To accomplish efficient and precise teleoperation, a robotic software architecture is designed from assemblages of reactive behavioral agents. Semi-automatic teleoperation is achieved by correlating operator motion inputs to the reactive behaviors. Such an agent-based robotic architecture has the advantage that it supports incremental build-up of new behaviors.

To exploit the dual-arm manipulator's potential task capabilities, kinematic equations for the dual-arm collaborative motions are incorporated in the coordination function of the robotic architecture. The introduction of a structured light and camera system provides a mechanism for guiding teleoperation.

Figure 7. Absolute Robot Motion

REFERENCE

1. G. J. VanderBurg, J. S. Albus, and E. Barkmeyer, "A Vision System for Real Time Control of Robots," Proceedings of the 9th International Symposium on Industrial Robots, pp. 213-231, Washington, D.C., March 13-15, 1979
2. R. C. Arkin, "Navigational Path Planning for a Vision Based Mobile Robot," *Robotica*, Vol. 7, pp. 49-63, 1989
3. R. C. Arkin, "Reactive Control as a Substrate for Telerobotic Systems," *IEEE Aerospace and Electronics System Magazine*, Vol. 6, No. 6, pp. 24-31, June, 1991
4. A. A. Maciejewski and C. A. Klein, "Obstacle Avoidance for Kinematically Redundant Manipulators in Dynamically Varying Environments," *Int. J. Robotics Res.*, vol. 4, No. 3, pp. 109-117, 1985
5. C. L. Lewis, "Trajectory Generation for Two Robots Cooperating to Perform a Task," Proceedings of the 1996 IEEE International Conference on Robotics and Automation, pp.1626-1631, Minneapolis, Minnesota, April, 1996
6. D. E. Whitney, "Resolved Motion Rate Control of Manipulators and Human Prostheses," *IEEE Tr. Man-Machine Systems*, vol. MMS-10, No. 2, pp47-53, 1969
7. H. Stone, *Introduction to Computer Architecture*, 2nd ed., SRA, Chicago, 1980