Preliminary results in force-guided assembly for teams of heterogeneous robots.

Juan Rojas[a] and R. A. Peters II[a,b]

[a]Center for Intelligent Systems, Vanderbilt University School of Engineering, Nashville, TN 37235, USA.

[b]Universal Robotics, P.O. Box 171062, Nashville, TN 37217 USA

ABSTRACT

The missions to the Moon and to Mars currently being planned by NASA require the advanced deployment of robots to prepare sites for human life support prior to the arrival of astronauts. Part of the robot's work will be the assembly of modular structures such as solar arrays, radiators, antennas, propellant tanks, and habitation modules. The construction will require teams of robots to work cooperatively and with a certain degree of independence. Such systems are complex and require of human intervention in the form of teleoperation attending unexpected contingencies. Latency in communications, however, will require that robots perform autonomous tasks during this time window. This paper proposes an approach to maximize the likelihood of success for teams of heterogeneous robots as they autonomously perform assembly tasks using force feedback to guide the process. An evaluation of the challenges related to the cooperation of two heterogeneous robots to join two parts into a stable, rigid configuration in a loosely structured environment is conducted. A control basis is such approach: it recasts a control problem by concurrently running a series of controllers to encode complex robot behavior. Each controller represents a control law that parses the underlying continuous control space and provides asymptotic stability, even under local perturbations. The control basis approach allows several controllers to be active concurrently through the null space control technique. Preliminary experimental results are presented that demonstrate the effectiveness of the control basis to address the challenges of assembly tasks by teams of heterogeneous robots.

Keywords: Cooperative heterogenous robots, autonomous assembly, force sensing.

1. INTRODUCTION

The missions to the Moon and to Mars currently planned by NASA require the deployment of robots to prepare sites for human life support prior to the arrival of astronauts. Part of the robot's work will be the assembly of modular structures such as solar arrays, antennas, and habitation modules. The construction requires teams of robots to cooperate with a certain degree of independence. Although they will be tasked and supervised by human operators on Earth it will be necessary for the robots to be autonomous over limited time intervals. Latencies in space communications preclude instantaneous teleoperation of the kind that is now used to control the NASA Robonaut. Currently a teleoperator dons a VR helmet, data gloves, and other sensors that connect to the robot so that the operator's motions are reflected by the robot concurrently with effectively no time delay.[1] Even without a delay, such minute teleoperation is difficult, tedious, tiring, and prone to errors that often lead to several attempts at a task before it is completed. A team of robots individually controlled that way by a number of operators tasked to jointly handle and mate parts for assembly would greatly magnify those problems, further challenged by an intermittent connection to the robots.

Traditional robotic control requires model-based control techniques on manipulators and planning of executed trajectories and forces. New goals in space robotics require the deployment of teams of heterogeneous robots that can perform assembly tasks autonomously in open environments. Such implementation requires that robot *coordination and cooperation* use force sensing to drive flexible and robust control strategies. Many current control

Further author information:
J. Rojas: E-mail: Juan.Rojas@Vanderbilt.Edu, Telephone: 1 615-322-0266
R. A. Peters II: E-mail: rap2@vuse.vanderbilt.edu, Telephone: 1 615-322-0266

Unmanned Systems Technology XI, edited by Grant R. Gerhart, Douglas W. Gage, Charles M. Shoemaker,
Proc. of SPIE Vol. 7332, 73320J · © 2009 SPIE · CCC code: 0277-786X/09/$18 · doi: 10.1117/12.819038

solutions require monolithic controllers and/or pre-planning to execute tasks. Our approach modularizes and encapsulates the control problem and recasts it in terms of a sequence of locally robust and reactive controllers. To date, there are no teams of multi-manipulator systems that use force sensing to perform joint assembly tasks in open environments. This work focuses in advancing the capabilities of autonomous heterogeneous robots in the area of low-level automated tasks such as assembly. It presents a control strategy that will allow independent robots in loosely structured environments to carry out beginning-to-end insertion tasks. This work is the first of its type and promises to solve very useful and necessary problems as part of the future developments in space exploration. The paper proposes a solution for effective coordination using a shared control basis composed of hybrid position, force, and moment controllers. In Section 2, we survey the state of the art in related fields and highlight unresolved problems. Section 3, reviews the theory of the Control Basis. Section 4, presents controllers used in cooperative assembly tasks. Section 5, presents experiments and results. Section 6 remarks on the effectiveness of the control basis.

2. RELATED WORK

Multi-manipulator coordination for assembly first began with homogeneous robots in the assembly line. We present the development of a number of adaptive techniques that have been developed to adjust to different kinds of uncertainties.[2] Due to challenges in flexibility and autonomy, researchers have sought to develop schemes for flexible and fault tolerant cooperation. Heger *et al*, for example, have worked on a framework for sliding autonomy in human-robot interaction for a number of years. The level of control that a human supervisor has over a robot can vary from strict teleoperation to full robot autonomy within the course of a single interaction.[3] Similar efforts to solve dexterous challenges through the use of cooperative heterogeneous robots is presented as well.

2.1 Homogeneous Multi-Robot Manipulators in Assembly Tasks

Development of control systems for homogeneous multi-robot manipulators for assembly tasks began in the late 1980's.[4] The assembly line was the setting for static multi-armed robots where strict schedules had to be maintained and the environment strictly controlled. Two classes of coordination schemes were pursued, *centralized systems* and *decentralized systems*.[5] The former combines all the robots into a single system and exerts a tight coordination between them. In a highly controlled environment such a system can exhibit a high throughput. One drawback of the approach is brittleness – a single, localized failure can subvert the entire system.[6] In the latter class, robots work independently, without higher-level coordination; they interact physically at the task level, rendering them more fault tolerant and flexible, however, tight coordination is challenging.

Neither of the schemes provides the degree of cooperation or coordination necessary to overcome the challenges posed by autonomy in open environments. Multi-robot coordination for assembly is thwarted by uncertainties in the system and the environment. Adaptive techniques have been developed to cope with uncertainties in scheduling, the system, and the environment.[2,7–9] Even though modern robot manipulation systems are able to adapt to a small degree of uncertainty, they remain largely inflexible and restricted primarily to manufacturing and the assembly line.[10]

2.2 Heterogeneous Robots and Assembly

The use of heterogeneous robots for autonomous assembly tasks is a novel field.[3,6,11] At the Carnegie Mellon Robotics Institute, researchers are working on a team of heterogeneous robots that coordinate to perform simple assembly tasks without force-feedback . They use a sliding autonomy architecture with a fine degree of granularity. Experimental results showed that marked improvement in the efficiency and performance of the task is attained when used under sliding autonomy. Researchers at NASA and the Jet Propulsion Laboratory have work with a group of two mobile robots that grasp, transport, and place beams on fixed structures using a behavior-based architecture.[12] Their experiments had high precision requirements and were performed in a natural environment with variable lighting.

Our work differs in that we use individual force-control actively across robots for assembly operations, while we enable them to have leader-subordinate roles or leader-leader roles. The JPL group likewise employs force feedback but they use it to coordinate the formation for their robots during material transport. We use force feedback not only for inter-robot communication but also within the control loops for insertion and mating of parts. At this time, there is no team of autonomous robots that coordinate their work to perform an assembly task via mutual force-torque feedback guidance.

3. THE CONTROL BASIS APPROACH

The control basis is a primitive organizational structure composed of parametric closed-loop controllers. The approach decomposes a complex control system into a series of modular control elements that when connected appropriately synthesize a variety of behaviors. The approach is rooted in closed-loop control for the following reasons: (i) nearly all action in robotics is encompassed under closed-loop control, (ii) it is a basis for error suppression, and (iii) the interaction between the robot and the world is modeled as a dynamical system.[13,14] This approach was originally suggested by Coelho and Grupen at the University Massachusetts, Amherst.[15] A basis comprises a number of closed loop controllers, which represent primitive actions, and can be derived from a set of control laws. While similar approaches appear in the literature, the way that a control basis factors objectives and captures the declarative structure of a solution was novel when first proposed.[14] In other words, each controller in the basis represents a single objective that is achieved from sensory inputs and effector (or commonly actuator) outputs.

Each controller in the set is a primitive that synthesizes on-line behavior by limiting the closed-loop response to specific stimuli and by engaging specific effector resources on low-dimensional subspaces of the system state-space. Complex behavior is generated by combining and sequencing a number of the primitive closed-loop controllers. The sequence is in effect an "instruction set" for performing complex tasks and provides a structure for larger-scale control policies. That prevents the generation of a monolithic controller, which is of limited applicability in open environments. It also increases robustness by reducing the need for complete and accurate system models.[13]

The feedback controllers are derived combinatorially from control laws. These laws are designed to yield asymptotically stable and predictable behavior under different robotic platforms and contexts. The careful selection of a small set of control laws permits flexible solutions for a wide variety of tasks. Such has been demonstrated by the implementation of grasp control, dexterous manipulation, whole body grasping, bipedal walking and other behaviors.[13,16] Asymptotic stability implies that the control laws partition the underlying continuous space into discrete basins of attraction. Each control law, ϕ_i in a set Φ is designed from a scalar potential function that maps independent configuration space variables to real numbers. Using greedy descent on a given potential surface, the system converges to the attractor within the basin that holds the current state. Within a given basin, a control law compensates for limited perturbations and uncertainty while still converging to the attractor. Asymptotic stability, as asserted by Huber, is then a function of the design of the potential surface and the dynamics of the robot.

3.1 Controller Synthesis

As part of the control basis approach, a controller, $\phi_{\mathbf{i}}$, is synthesized when parameterized in terms of a sensor transform s_i and an output effector transform e_j. The binding of a potential surface function to sensory inputs and effecter outputs is designed to bind the objective of the controller to specific sensor resources and to use specific actuators to accomplish that objective. The *sensor transform*, s_i, effectively represents a mapping from a limited set of available input control resources, $\Gamma_i \in \Gamma$ to a specified domain space, X_i. That is to say, an input control resource such as a position or force sensor would actually deliver a cartesian position or force reading: $s_i : \Gamma_i \rightarrow X_i$.

The artificial potential function then represents a native objective for the system and can be realized in a number of ways depending on how it is bound with possible input sensory resources and output motors. It is important to select domain general potential functions that prevent the system from being constrained to specific strategies and allows tasks to adapt during run-time.[14] As such, the shape of the potential function depends on the input and output resources that are engaged. Each objective function is optimized by descending the gradient of the surface error through greedy descent of the form $\triangle q = -K * \nabla_{X_i}\phi_i$, where

$$\nabla_{X_i}\phi_i = \frac{\partial \phi_i(s)}{\partial X_i} \tag{1}$$

Finally, the *effector transform*, e_k, maps the error result in terms of a subset of the output control resources, $\Gamma_k \in \Gamma$, to the output space, Y_k. This is typically done through a Jacobian matrix that converts either position or force gradient data into the robot's joint space:

$$e_k(\Gamma_k) = \left(\frac{\partial \mathbf{x}_{\gamma_1}}{\partial \mathbf{y}_k}, \frac{\partial \mathbf{x}_{\gamma_2}}{\partial \mathbf{y}_k}, ..., \frac{\partial \mathbf{x}_{\gamma_{|S_k|}}}{\partial \mathbf{y}_k} \right)^T \tag{2}$$

where $\mathbf{x}_{\gamma_1}$ represents the configuration of control resource γ_i. y_k is a point in the output space (i.e. an element of the six DoF joint space vector), and $\Gamma_k = \{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_{|\Gamma_k|}\} \in \Gamma$ is a subset of the control resources (i.e. selecting one end-effector from a set of two available to a humanoid robot).

In summary, the closed-loop controller is implemented by binding an artificial potential gradient, $\nabla_{x_i}\phi_i$, with a sensor transform, $s_j(\Gamma_j)$, and an effector transform, $e_k(\Gamma_k)$. The input data must be of the same domain type as the artificial potential function, and the effector transform must have the same dimensions as the potential function in order to map the result into the convening output space.

3.2 Example

A cartesian controller for a robotic manipulator is used to illustrate the concept.[17] Cartesian position control is implemented through the Jacobian transpose control method. In assembly, cartesian control is used to displace the tip of a mating part to a reference location set by the six-element vector x, y, z, r, p, y. The controller computes the position error and multiples it by a gain and the transpose of the manipulator jacobian. The result is a vector of joint angle updates. The updated positions are then passed to a low-level joint angle controller which actuates the robot to the desired position. The last step of the loop uses forward kinematics to transform the output into desired cartesian coordinates. The controller is a primitive in the set of basis controllers used in this work. In the position primitive, the Jacobian transform serves as the sensor transform and the forward kinematics serve as the effector transform. The difference between desired position and actual position is the error measure for the controller: $\phi_i(\mathbf{x}_{ref} - s_j(\Gamma_j))$, and the controller descends this potential through greedy descent $\nabla_{x_i}\phi_i$. This basis primitive is mathematically described as:

$$\nabla_{y_k}\phi_i = e_k(\Gamma_k)^T \nabla_{x_i}\phi_i(\mathbf{x}_{ref} - s_j(\Gamma_j)). \tag{3}$$

Throughout the rest of the paper, this mathematical notation is represented for simplicity as:

$$\phi_i \mid^{s_j(\Gamma_j)}_{e_k(\Gamma_k)} (\mathbf{x}_{ref}) \tag{4}$$

Note that the reference input in the controller could also be the output of another basis controller as long as the task space was the same. This allows for nested controllers to be implemented:

$$\phi_1 \mid^{s_{1j}(\Gamma_{1j})}_{e_{1k}(\Gamma_{1k})} \left(\phi_2 \mid^{s_{2j}(\Gamma_{2j})}_{e_{2k}(\Gamma_{2k})} \right). \tag{5}$$

When more than one controller is at work within a system, multiple goal states are demanded from the robot's actuators. A multi-objective composition technique is used to ensure the optimization of multiple goals.

3.3 Multi-Objective Composition

Often when robotic systems are asked to execute a task, robots use an excess of sensory and motor degrees of freedom. This redundancy can be used to achieve a number of varying goals. In so doing, one must be careful to ensure that the pursuit of one goal does not affect the execution of another goal. For such an end, the Moore-Penrose pseudo inverse has been a foundational mathematical tool.[14] This concept was originally used to optimize secondary objectives across controllers.[18] The pseudo-inverse method can be used to project the gradient result of the secondary objective onto the equipotential manifold of the primary objective, thereby achieving two simultaneous goals. Platt introduced the formal approach to incorporate the Moore-Penrose pseudo inverse to the control basis and called it: *null space composition*.[18] Null space composition enabled controllers to run independently of each other while pursuing their own goal states. Controllers could be combined hierarchically, where dominant and subordinate controllers exist. The latter were limited to the null space of the dominant controllers, yet the goal of the subordinate controllers could still be optimized. A simple notation was used to represent this composition and it was said that the subordinate controller is *subject-to* the dominant controller: $\phi_s \triangleleft \phi_d$. In assembly, the force controller can be subject-to a moment controller to concurrently achieve a corrective alignment motion and an insertion motion: $\phi_{force} \triangleleft \phi_{moment}$. Consider the gradient output of two primitives, ϕ_m, ϕ_f:

$$\nabla_{y_m}\phi_m = e_k(\Gamma_k)^T \nabla_{x_m}\phi_m(s_j(\Gamma_j)) \text{ and } \nabla_{y_f}\phi_f = e_k(\Gamma_k)^T \nabla_{x_f}\phi_f(s_j(\Gamma_j)), \tag{6}$$

where the gradient outputs of both primitives must reside in the same domain space. When the gradient of the second controller is projected to the null space of the first gradient, the result is represented as:

$$\nabla_y(\phi_f \triangleleft \phi_m) = \nabla_y \phi_m + \mathcal{N}(\nabla_y \phi_m^T)\nabla_y \phi_f, \tag{7}$$

where

$$\mathcal{N}(\nabla_y \phi_m^T) \equiv I - (\nabla_y \phi_m^T)^T (\nabla_y \phi_m^T), \tag{8}$$

and, I, is the identity matrix, y is an n-dimensional space, and $\nabla_y \phi_m$ is a (n-1) dimensional space orthogonal to the direction of steepest descent.[17] In this way, the "subject-to" constraint, $\phi_f \triangleleft \phi_m$, always descends the potential surface of the dominant controller that has not yet converged to a minimum, and maintains the stability of the underlying controller without limiting its actions.

3.4 Control Policy Construction

Once there is a number of defined primitives in the set of basis controllers, Φ, the decision of how to combine them must be made. The size of the set is defined by $\Phi \times 2^s \times 2^e$, giving rise to many possible transitions. While this functionality makes the control basis more powerful and flexible to adapt to new tasks, it is necessary to successfully transition between segmental actions.[13]

Closed-loop controllers interact with the environment over time revealing information between the system and the world. The asymptotically stable controllers of the basis tend to equilibria as time goes to infinity: $\frac{\partial \phi}{\partial t} \rightarrow 0, \text{as t} \rightarrow \infty$. The convergence of a controller can be treated as a discrete event and used as criteria to join controllers. This phenomena simplifies the number of possibilities and establishes a framework for controller policy enaction. Sequences of controllers can then be considered states in a non-deterministic, finite-state automata. Each finite state is then an asymptotically stable sequence of controllers and describes patterns of membership across controllers.[14] The next section describes how moment, force, and position primitives can be connected to produce robust low-level assembly tasks.

4. DEVISING A CONTROL BASIS FOR JOINT ASSEMBLY TASKS

Assembly tasks involve the insertion of one part into another, *i.e.* the insertion of a truss into a fixture. Such an operation can be carried out successfully if a robot displaces its tool-tip to minimize the moment and force residuals experienced while aligning participating assembly parts. The points of contact between assembly parts in this work are referred to as "contacts". A number of controllers are designed to respond to the construction of an assembly process consisting of two stages: (a) a guarded approach, and (b) a compliant insertion. A series of primitive controllers are used, combined and sequenced to produce the following composites: the "guarded move controller" and the "compliant insertion controller". A control policy that starts with the guarded move controller and that transitions to the compliant insertion controller is used to achieve an insertion.

For an insertion task to be executed successfully a robot must first be able to displace the assembly part to an optimum location for insertion. If the approach ends at a location outside the interior hole of a fixture the assembly cannot succeed. Similarly, if the approach is too quick and jams the truss into a fixture, a successful insertion is unlikely. The *Guarded Move Controller* is designed to generate a guarded approach that positions the tool at an optimum location for insertion. Similarly, once an appropriate insertion location has been achieved, controllers must drive the insertion as smoothly as possible. To do so, misalignments must be corrected to decrease the friction, jamming, and wedging phenomena. The *Compliant Insertion Controller* was designed to do just that. Before presenting the derivation of the composite controllers, the concept of a wrench residual controller is used to introduce force error minimization.

A wrench is a 6-dimensional generalized vector of forces and torques $\vec{w_i} = [\vec{F}\ \vec{T}]^T$. In insertions, a wrench vector is generated when an assembly part makes contact with a mating fixture. A wrench residual ϵ, is then, the sum over forces and moments generated across n contact points:

$$\epsilon = \sum_{i}^{n} w_i. \tag{9}$$

As introduced in Section 3.1, a control primitive is a scalar potential function that represents a control goal. In the assembly problem, correct insertion requires that the wrench residual is zero. This become a goal of the control basis. Therefore, the wrench residual potential surface ρ, is defined as the square of the wrench residual:

$$\rho = \epsilon^T \epsilon \tag{10}$$

To reach the goal of the function, greedy descent is used to converge to the attractor of the surface. Descending the surface implies differentiating the wrench residual with respect to the joint configuration of the robot, then displacing the contacts in direction of the gradient vector. Coelho and Grupen noted that local minima sometimes led to spurious states when the wrench residual was minimized.[15] To prevent that from occurring, they decomposed the wrench residual into two orthogonal force and moment residuals, such that:

$$\rho_{fr} = \left(\sum_{i=0}^{n} \mathbf{f}_i\right)^T \left(\sum_{i=0}^{n} \mathbf{f}_i\right) \quad \rho_{mr} = \left(\sum_{i=0}^{n} \mathbf{r}_i \times \mathbf{f}_i\right)^T \left(\sum_{i=0}^{n} \mathbf{r}_i \times \mathbf{f}_i\right), \tag{11}$$

where $\mathbf{f}_i$ is a unit vector normal to the object surface applied by the i^{th} contact, and $\mathbf{r}_i$ is the location of the i^{th} contact point in a given coordinate frame.

For grasp tasks, Coelho and Grupen determined that the force played a more dominant role in the task. They proceeded to descend the force residual first and then the moment residual and combined both contributions through a control composition policy. Later, Platt proposed a null space projection (see Section 3.3) where the result of the subordinate force residual was projected onto the dominant moment residual and both could be minimized correctly. A similar approach is used for the compliant insertion controller used in this work and is introduced next.

4.1 Compliant Insertion Controller

For a given assembly task, the robot manipulator holds a truss with an inverted chamber on the end. When a force is applied at the tip of a truss, the exerted force is primarily experienced as an applied moment by the force-torque sensor which sits on the wrist of the robotic manipulator. Through the insertion itself, the truss experiences moments as it collides with the walls of the mating part. Additionally, in order for the "insertion motion" to be carried out, there must be a driving force in the direction of the axis of insertion that pushes the truss until a connection is completed.

In effect, the insertion process is subordinate first to the moments of the task and then to the forces of the task. The former deals with the alignment of the parts which is crucial for the task to be successful. The latter deals with the forward motion which drives the insertion process as misalignments are corrected. With these factors in mind, a composite controller known as a "Compliant Insertion" controller was designed. This basis controller is slightly different from the wrench residual introduced earlier. It designates the moment primitive as the dominant actor and the force primitive as the subordinate actor in correspondence to the goals of the task. The derivation of this hierarchical component is presented next.

The moment primitive, ϕ_{mr}, uses the sensed moments to update the joint angle configuration of the manipulator and, therefore move, the rigidly held tool to a state where there are no net moments. The sensor transform, $s_q(\gamma_{moment})$, maps the sensed moments in joint torques. The effector transform, $e_q(\gamma_{torque})$, transforms those updated values into joint coordinates, q. The controller is represented in simplified notation as:

$$\phi_{mr} \,|^{s_q(\gamma_{moment})}_{e_q(\gamma_{torque})} \;. \tag{12}$$

Similarly, the force residual, ϕ_{fr}, displaces the joint angle configuration of the robot to achieve a reference value. The reference value can be used to implicitly drive the motion of the insertion:

$$\phi_{fr} \,|^{s_q(\gamma_{force})}_{e_q(\gamma_{torque})} (f_{ref}). \tag{13}$$

The objectives of both primitives can then be attained through null space composition. The goal vector of the force controller is projected onto the null-space of the moment vector which allows correct alignment and insertion to take place concurrently:

$$\nabla_q \phi_{mr} + \mathcal{N}(\nabla_q \phi_{mr}^T) \nabla_q \phi_{fr}, \tag{14}$$

where

$$\mathcal{N}(\nabla_q \phi_{mr}^T) \equiv I - (\nabla_q \phi_{mr}^T)^T (\nabla_q \phi_{mr}^T). \tag{15}$$

and the sensor and effector transforms have been omitted for clarity. Composite controllers which result from the use of the null-space operation can be represented in simplified notation through the use of a new parameter, π_i. The Compliant Insertion controller is then defined as:

$$\pi_{CI} \,|^{s_{CI}(\gamma_{force})}_{e_{CI}(\gamma_{torque})} = \phi_{fr} \,|^{s_{fr}(\gamma_{force})}_{e_{fr}(\gamma_{torque})} (f_{ref}) \triangleleft \phi_{mr} \,|^{s_{mr}(\gamma_{force})}_{e_{mr}(\gamma_{torque})} \tag{16}$$

4.2 Guarded Move Controller

Prior to an insertion action it is necessary for the assembly fixture to have reached an appropriate basin of attraction. The assembly fixture must reach a near optimal location - preferably one in which initial and slight contact is made between the incoming truss and the interior hull of the fixture. The goal of the guarded move controller is then to reach a joint configuration that allows the potential surface to converge to a state of low moments and forces such that the transition to the compliant insertion controller is smooth.

The guarded move controller is a composite controller consisting of a dominant position controller and a subordinate moment residual controller. The position controller displaces the manipulator and the rigidly held truss to an optimal location for insertion. The subordinate moment controller in this case acts as an identifier to determine whether contact has been made. The position controller receives its goal cartesian position from the stereo visual system. The latter uses color fiducial marks placed at the tips of the fixtures to estimate 3D cartesian positions. Color segmentation and noise filtering are performed on incoming video and processed to compute image coordinates of the segmented foreground blob that correspond to the marker. The error centroid position of the blob from the right and left images ($\triangle c_r, \triangle c_l$) is fed through a back propagated neural net (NN_{bp}) to output appropriate updates for camera motor servos. The cameras verge towards the region of interest until the object is centered,[19] and represented by the following equation:

$$servo = k * \triangle q = k * (NN_{bp}(\triangle c_r, \triangle c_l)) \tag{17}$$

The motors' output pan and tilt angles are used to localize the cartesian coordinates of a blue and a green tool-tip. The orientation of the tools is not computed by the visual system. In the experiments the orientation of the wrist is assumed to always point approximately parallel to the robot's $\vec{x}$ base-coordinate. This orientation is best for insertion tasks since both robots apply force perpendicularly to the plane on which they are pushing. The cycle is performed sequentially for each of the two tool-tips involved in the task. The difference in cartesian coordinates between the tips is used to update the desired goal position in the corresponding position controller.

The guarded move controller drives the assembly tool to the cartesian position generated by the sensor transform $s_p(\gamma_{visual\_sys})$ and returns the current joint configuration of the robot. The latter uses forward kinematics as the effecter transform: $e_p(\gamma_{joint})$. The position controller is thus defined as:

$$\phi_p \,|^{s_p(\gamma_{visual\_sys})}_{e_p(\gamma_{joint})}\, (x_{actual} + \triangle x). \tag{18}$$

The moment residual controller is bound by a sensor transform, $s_{mr}(\gamma_{force})$, that returns the moments experienced by the force sensor, and by an effector transform $e_{mr}(\gamma_{torque})$, that converts torque updates into joint angle updates. The moment residual controller is defined as:

$$\phi_{mr} \,|^{s_{mr}(\gamma_{force})}_{e_{mr}(\gamma_{torque})}\, . \tag{19}$$

The goal of the guarded move controller is to reach an appropriate location for insertion; namely, one that is close to the interior hull of the fixture and where there are no net moments. If contact takes place, the controller displaces the end-effector of the robot to minimize the residual error and transitions to compliant insertion controller, π_{CI}, which in turn compensates for the disturbances through the insertion.

The composite guarded move controller, π_{GM}, is synthesized by having the moment controller be subject-to the position controller:

$$\pi_{GM} \,|^{s_{GM}(\gamma_{joint,torque})}_{e_{GM}(\gamma_{cart,moment})} = \phi_{mr} \,|^{s_{mr}(\gamma_{moment})}_{e_{mr}(\gamma_{torque})} \triangleleft \phi_p \,|^{s_p(\gamma_{visual\_sys})}_{e_p(\gamma_{joint})}\, (x_{actual} + \triangle x). \tag{20}$$

The two hierarchical components presented in this section are able to generate robust insertion operations for a given manipulator. Our control policy for the task consists of initiating the job with the guarded move controller and as it reaches its basin of attraction, a finite state automata moves the system to the next state where the compliant insertion controller becomes active.

5. EXPERIMENTS

Three experiments are presented which demonstrate the ability of the control basis approach to carry out assembly insertions in an industrial manipulator and also across a team of heterogeneous robot where a compliant humanoid robot uses position control. Experiments also test the accuracy of the visual system used for these experiments. The two controllers presented in Section 4 are used in the industrial manipulator for these experiments. Robotic control is deployed through an in-house distributed software architecture known as the Intelligent Machine Architecture 2.5.[?]

5.1 Experiment 1: HP3JC Stand-Alone Experiment

The purpose of the first experiment is to set baseline performance measures for future robot cooperation in terms of: (a) the overall behavior of the task; (b) controller gains for the moment, force, and position primitives, (c) the force reference vector in the compliant insertion controller, $\pi_{CI}(f_{ref})$; and (d) the determination of the error threshold for a smooth transition between the guarded move controller and the compliant insertion controller.

5.1.1 Results

Initial experiments yielded an unstable behavior due to high gains in the moment residual primitive. As the truss was inserted into the mating fixture the controller over-compensated and coerced the fixture out of place. Table 1A shows the moment and force readings of the JR3 F/T sensor and the moment and force residual errors.

The torque readings in the y-direction increased dramatically during the last 50 seconds of the task. They rose from 0 to a maximum absolute value of 25in-lbs, this is reflected by the moment residual error. The change in the force reference values corresponded to an attempt to modify the behavior of the insertion during execution. Force reference values affect the speed and direction of the insertion approach. A new set of system parameters was selected after running more trials. The new parameters yielded a faster and smoother descent of torque error in half the time of the previous trial, as shown in Table 1B. Half-way through the insertion, the force reference value F_x was attenuated by 50% to study the response of the manipulator's motion. The reduction in value

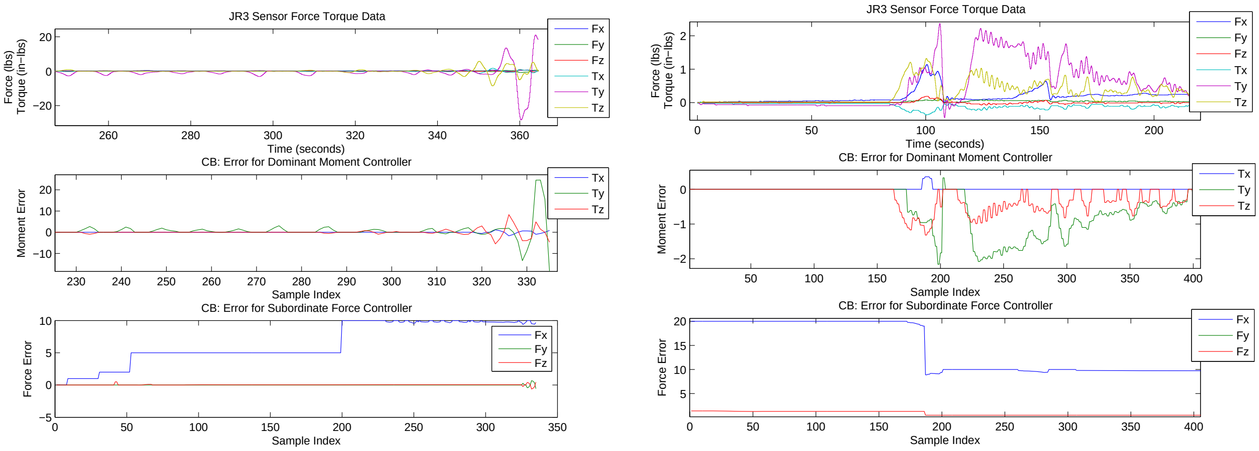

Table 1. Force data for experiment 1.

softened the impact during insertion but the time-to-completion increased. Increased force reference values were used to speed up the assembly task and set to: $< 40\vec{x}, 0\vec{y}, 2.7\vec{z}lbs >$. The assembly ran faster than previous trials, taking 85 seconds to complete. Sensor readings were not as smooth as prior ones. They registered slightly larger torque signatures due to the higher reference force values. Nevertheless, the insertion controller reacted favorably to minimize residual error and successfully complete the insertion.

5.1.2 Summary

The control basis effectively executed a rigid assembly task on an industrial robot. A total of 11 trials were run as part of two tests: a straight approach and a side approach. For each trial, duration and the maximum absolute values for the moment error in all three directions: x, y, z, and the sum were recorded. The first and slowest trial lasted 470 seconds. The fastest trial in the set lasted 70 seconds. The average time across trials was 106 seconds. For all trials the controllers were able to minimize the net moments and forces in all directions. Moment residuals in the y-direction were higher for trials with a straight approach, while residuals in the z-direction were higher for those with a side approach. The average value of the sum of errors for experiment 1 was of 10.7 in-lbs.

5.2 Experiment 2: The Visual System

The visual system uses the error between the position of the centroid of a foreground blob that corresponds to a fiducial and the center of the image to trigger the camera's pan and tilt servos. These in turn are used to estimate the 3D cartesian position. The accuracy of the estimate relies on how accurately the centroid can be centered in the screen. Robust noise reduction algorithms and accurate servo steps are essential for accurate positioning. Additionally, the visual system does not calculate the cartesian positions until it has verified that the centroid error is under an empirical threshold. This system recognized colored fiducial marks: a blue colored tip for the truss held by the HP3JC robot and a green colored tip for the fixture held by the humanoid robot ISAC. The visual system first saccades to the blue color mark, after verifying stability a position is computed. Then it saccades to the green color mark and repeats the process. A transform is derived and used as a reference position for the π_{GM} controller.

5.2.1 Results

For the experimental set-up, the HP3JC robot was placed in front of ISAC while grasping the male truss. Then, the female fixture was: first placed on top of a table in front of ISAC; and secondly, rigidly held by both of ISAC's manipulators. The former allowed for easy set-up and a preliminary analysis of the system. The latter had both tool-tips in positions close to those used in the rest of the experiments. In both cases and throughout a number of trials, physical measurements were taken to find the distance between both tool-tips.

Initial results are shown in Table 2A. The average computed transform (in mm) over 8 trials was of: $< 116.5\vec{x}, -9.3\vec{y}, -73.9\vec{z} >$; with an average error of: $< -3.5125\vec{x}, -9.3278\vec{y}, 46.0999\vec{z} >$. The z-coordinate showed

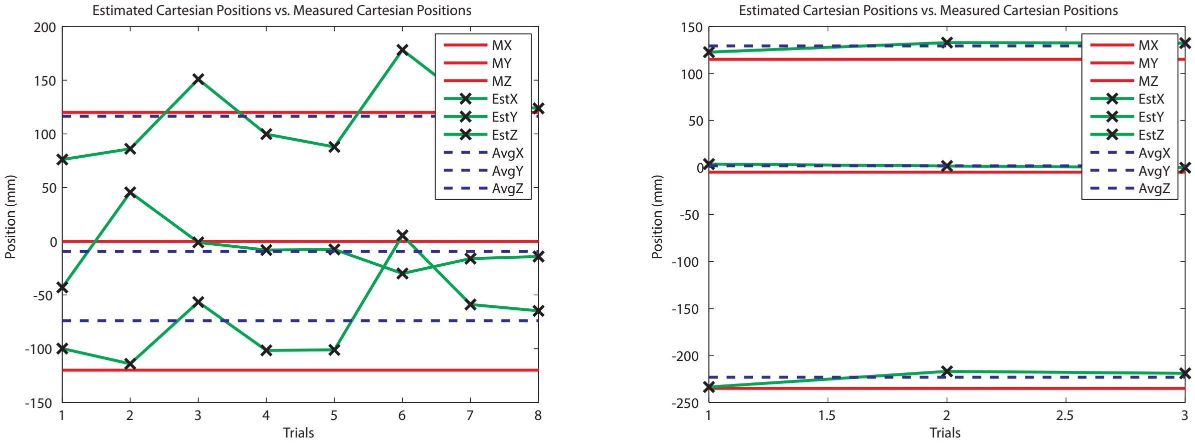

Table 2. Comparison between computed estimates, measured distances, and average computed positions.

the largest margin of error. The cameras presented some jitter motion during these trials. To improve stability, (a) the velocity and acceleration of each camera was decreased by half to aid the integrity of the color segmentation process, (b) the error threshold for head stability was decreased, and (c) narrowed color segmentation parameters. Improved results are shown in Table 2B.

The average position for the new set of trials was: $< 129.2595\vec{x}, 1.6289\vec{y}, -223.2044\vec{z} >$ and errors decreased to: $< 14.2595\vec{x}, 6.6289\vec{y}, 11.7956\vec{z} >$. Compared to the previous set of trials, the disparity between computed and measured position is smaller for $\vec{y}$ and $\vec{z}$ and slightly larger for $\vec{x}$. These results seem to be of good-fit considering that the aperture of the female fixture held by ISAC is of 25.4mm. Nonetheless, a bias of 20mm was applied in the negative x-direction to decrease the likelihood of having the truss surpass the entry point of its mating part during the guarded move approach. The source of error in the x-direction may be due in part to the color segmentation process. The colored fiducial mark for the HP3JCs covers an inverted chamfer which is 4cm long, while the green fiducial is 3cm long. Given that the centroid is computed in both markers, an approximate 3.5cm error may be embedded in the color segmentation process. The visual system parameters derived here successfully provided reference positions for all trials in Experiment 3.

5.3 Experiment 3: Assembly with a Compliant Robot

The third experiment studied the effect of a compliant robot on the assembly process. Compliant devices have been commonly used on end-effector but we found no literature reporting on assembly with a compliant robot. While compliant robots are challenging to control, their elasticity can compensate for the high strain forces experienced with industrial manipulators. This experiment used the industrial HP3JC manipulator, the visual system, and the compliant humanoid robot ISAC. The latter rigidly held the female fixture and only used position control to fix the location of the fixture. The industrial robot held the male truss and began the guarded move approach when the visual system computed the reference position. As in experiment 1, the compliant insertion controller was triggered once the reference position was achieved.

Two outcomes are reported here. The first one corresponds to a smooth insertion, while the second had a truss ram against the edge of the female fixture. Both trials used force references of $<< 40\vec{x}, 0\vec{y}, 1.0\vec{z} >>$ for a slow insertion.

5.3.1 Results

The first trial yielded a smooth insertion that lasted 43 seconds. The controllers worked effectively to minimize moment residual error as seen in Table 3A. Torques in the y-direction reached 5in-lbs, while those in the z-direction reach 2in-lbs. Both residual errors were minimized to zero over the course of the task. No force interjections were experienced (forces that pushed the manipulator up or down, or even backwards). Compared

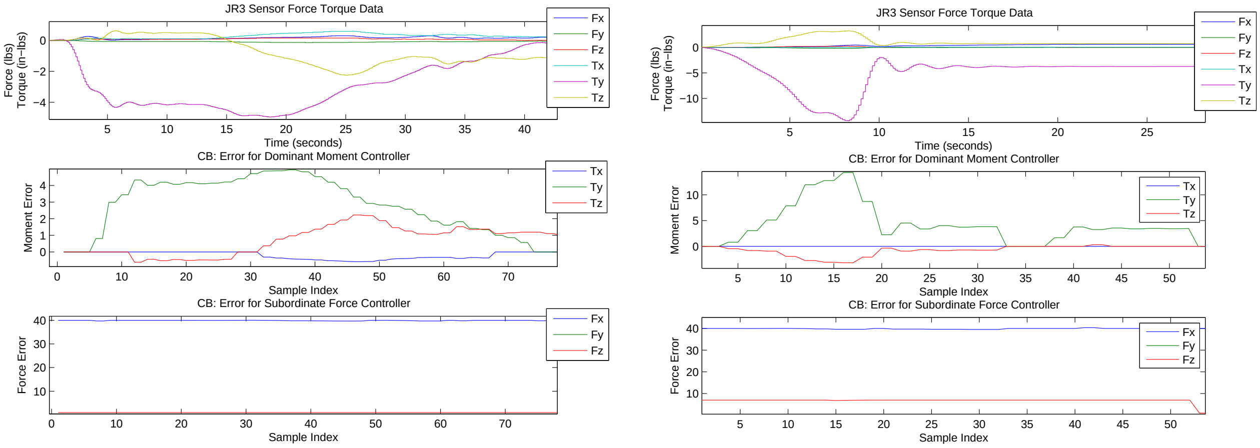

Table 3. Force data for experiment 3.

to the plots in experiment 1, this data exhibits smooth curvature over the duration of the task in response to the compliance present in the task.

Another interesting datum was revealed in the second trial. In this case, the guarded approach by the HP3JC robot forced the truss against the edge of the mating fixture for a few seconds. Over time, the forces experienced by both robots increased steadily, which delayed the time-to-completion of the task. After a few moments, the compliant robot jumped in the right direction and allowed the assembly task to proceed rather quickly. The higher stresses experienced by the industrial robot are shown in Table 3B. The sensor experienced the highest reading for Ty reaching a magnitude of 15in-lbs, which reflects the accumulation of stress by the wrist of the manipulator as it tried to adjust its motion while jamming the fixture. After a backlash motion, the truss quickly enters the fixture and the sensor's torque readings drop drastically, under 5in-lbs for the rest of the task.

5.3.2 Summary

Integrating compliance into the assembly process enhanced results as compared to those in experiment 1. Six trials were run, five of which were successful. Two sets of force reference values were consolidated in this experiment, one set was for "fast" insertions, the other was for "slow insertions". The former used: $< 40\vec{x}, 0\vec{y}, 1\vec{z} >$, and the latter used $< 20\vec{x}, 0\vec{y}, 0.5\vec{z} >$. The use of fast parameters yielded a a trial that lasted 43 seconds. The average time-to-completion for the experiment was 69 seconds, 35% faster than experiment 1. The use of compliance also lowered the average moment readings as compared with the rigid assembly performed in experiment 1. The average sum of the errors was of 8.94, 16.6% less than for its rigid counterpart. Hence, the compliance of the humanoid robot eased stresses in the process and also resolved a jam situation into a successful assembly.

6. CONCLUSION

The control basis approach was an effective approach to modularize and sequence the assembly process for an individual industrial robot that performed assemblies using force sensing in both rigid assemblies and joint assemblies with a compliant humanoid robot. The designed guarded move controller and compliant insertion controller were successfully implemented on the industrial HP3JC robot to achieve single robot autonomous assembly. Testing experience helped improve system parameters to establish appropriate controller gains and force reference values. Time-to-completion and the sum of moment residuals served as metrics to evaluate the performance of the experiments. The third experiment yielded faster assemblies and with lower moment residuals. The data suggests that cooperative mechanisms are more efficient than running tasks with individual robots. Data also indicates that compliance assisted in lowering the forces experienced by the system of robots in the task.

Future work will run force controllers on ISAC and test two coordination schemes: active-active and active-static schemes. These test how assembly performs when two versus one robot drive the insertion. These experiments will study which coordination mechanisms may be most efficient when performing joint assembly tasks.

7. ACKNOWLEDGMENTS

This work was supported by NASA grant NNX07AF04G.

REFERENCES

[1] Peters, R. A., Jenkins, O. C., and Bodenheimer, R. E., "Sensory-motor manifold structure induced by task outcome: experiments with robonaut," in [*International conference on Humanoid Robots (Humanoids 2005)*], 484–489 (2006).

[2] Namvar, M. and Aghili, F., "Adaptive force-motion control of coordinated robots interacting with geometrically unknown environments," *in Proc. IEEE Transactions on Robotics* **21**(4), 678–694 (2005).

[3] Heger, F. W., Hiatt, L. M., Sellner, B., Simmons, R., and Singh, S., "Results in sliding autonomy for multi-robot spatial assembly," *In Proceedings of the 2005 ISAIRAS Conference* (2005).

[4] Tao, J., Luh, J., and Zheng, Y., "Compliant coordination control of two moving industrial robots," *Decision and Control, 1987. 26th IEEE Conference on* **26**, 186–191 (Dec. 1987).

[5] http://www.prisma.unina.it/ramsete/coop.htm (Octobers 2007).

[6] Brookshire, J., Singh, S., and Simmons, R., "Preliminary results in sliding autonomy for assembly by coordinated teams," *IROS* **1**, 706–711 (Oct. 2004).

[7] Hu, Y. and Goldenberg, A., "Dynamic control of multiple coordinated redundant robots," *IEEE Transactions on Systems, Man, and Cybernetics* **22**(3), 568–574 (1992).

[8] Yuan, P., "An adaptive feedback scheduling algorithm for robot assembly and real-time control systems," *Proceedings of the IEEE International Conference on Intelligent Robots and Systems* , 2226–2231 (October 9-15 2006).

[9] Sun, D. and Mills, J. K., "Adaptive synchronized control for coordination of multirobot assembly tasks," *Proceedings of the IEEE International Conference on Robotics and Automation* **18**, 498–510 (August 2002).

[10] Brogardh, T., "Present and future robot control development–an industrial perspective," *Journal of Annual Reviews in Control* **31**(1), 69–79 (2007).

[11] Simmons, R., Singh, S., Heger, F., Hiatt, L. M., Koterba, S. C., Melchior, N., and Sellner, B. P., "Human-robot teams for large-scale assembly," in [*Proceedings of the NASA Science Technology Conference*], (May 2007).

[12] Stroupe A.; Okon A.; Robinson M.; Huntsberger T.; Aghazarian H., B., "Sustainable cooperative robotic technologies for human and robotic outpost infrastructure construction and maintenance," *Journal of Autonomous Robots* , 113–123 (April 2006).

[13] Huber, M., *A Hybrid Architecture for Adaptive Robot Control*, ph.d. dissertation, University of Massachusetts (Sept. 2000).

[14] Borck, O., Fagg, A., Grupen, R., Platt, R., Rosenstein, M., and Sweeney, J., "A framework for learning and control in intelligent humanoid robots," *International Journal of Humanoid Robots* **2**(3), 301–336 (2005).

[15] Coelho, J. A. and Grupen, R. A., "A control basis for learning multifingered grasps," *Journal of Robotic Systems* **14**(7), 554 – 557 (1997).

[16] Platt, R., Grupen, R., and Fagg, A., "Improving grasp skills using schema structured learning," *International Conference on Development and Learning* (May 2006).

[17] Platt, R. J., *Learning and generalizing control-based grasping and manipulations skills*, PhD thesis, University of Massachusetts Amherst (2006).

[18] Platt, R., Fagg, A., and Grupen, R., "Nullsapce composition of control laws for grasping," *In IEEE International Conference on Intelligent Robots and Systems* (2002).

[19] Rojas, J., *Sensory Integration With Articulated Motion On A Humanoid Robot*, Master's thesis, Vanderbilt University (2004).