2011 11th IEEE-RAS International Conference on Humanoid Robots
Bled, Slovenia, October 26-28, 2011

Automating Short-Term Insertion of Parts for Heterogeneous Robots Using a Control Basis Approach.

Juan Rojas* Electrical and Computer Engineering Department
Center for Intelligent Systems
Vanderbilt University
Nashville, TN 37235, USA
rojas70@gmail.com

Richard A. Peters II** Electrical and Computer Engineering Department
Center for Intelligent Systems
Vanderbilt University
Nashville, TN 37235, USA
rap2@vuse.vanderbilt.edu

Abstract— **Robotics technology is quickly evolving and demanding robots to perform more actions and with greater complexity. Modular construction tasks are well suited for heterogenous robots. In this paper we studied if modularizing a distributed multi-agent architecture with a control modular framework was a viable approach to generate complex robotic behavior with short term autonomy. To this end we tasked a heterogeneous robot team to perform an assembly task using force sensing. We also studied which robots would be better suited for a given task. Our experimental results concluded that the modular control approach was viable to generate short-term autonomous complex robotic behavior and our analysis characterized the advantages of each robot for the assembly task.**

I. INTRODUCTION

Robotics technology is quickly evolving and demanding robots to perform more actions and with greater complexity. Such robot systems need to perform those tasks with increased flexibility and autonomy [1]. As robotics moves towards multi-robotic heterogeneous systems, teams of robots must learn to maximize their abilities to most effectively cooperate and coordinate to achieve tasks [2]. In particular, teams of heterogeneous robots are well suited for modular construction tasks, such as those planned by NASA to build a lunar base. Robots will be operated under supervisory control to assemble modular solar arrays, antennas, satellites, and habitation modules. One critical aspect of successfully tasking these robots into space will be their ability to automate short-term tedious tasks. There are at least two reasons for this: a) due to the latency of communications between the Earth and the Moon such robots must be autonomous during the execution of the assembly task to avoid critical damage, and b) such low-level tasks are extremely tedious and prone to errors for teleoperators, specially when they are executed in a team of robots.

In order to achieve short-term and long-term automation, researchers have experienced that distributed multi-agent architectures facilitate the development, integration, and execution of robotic systems [3].

This paper proposed a control strategy that used three new control primitives and three new compound controllers that when, used in concert with our in-house built distributed multi-agent architecture–The Intelligent Machine Architecture (IMA),– enabled a heterogeneous robot team to reactively perform, under short-term autonomy, an assembly task using force sensing. The assembly task was executed under a push-hold scheme in which one robot used force sensing to drive a male truss while the other used force sensing optimize the entry at the female end. The scheme's roles were reversed so that both robots were able to drive the insertion and hold parts.

To this end a high-accuracy HP3JC industrial robot with an NX100 Yaskawa controller [4] was used though limited by the inaccessibility to its low-level control. This robot was also equipped with a JR3 six-axis F/T sensor and a Barret hand mounted on the end-effector. The second robot, ISAC, is a pneumatically actuated anthropomorph [5] using 12 pneumatic McKibben muscles arranged to have 6-DoF. McKibben muscles are highly compliant but are characterized by hard-to-control non-linear properties. ISAC was equipped with two ATI six-axis FT sensors and machined aluminum brackets designed to hold 1.0 in PVC pipes. A male and female truss were made from commercial PVC piping. The male truss was composed of two 0.5 in pipes connected by a 90 degree elbow connector and an inverted chamfer at the tool-tip to facilitate the entry of the female counterpart. The HP3JC's Barret hand held the male truss. The female truss, was an 1.0 in pipe, connected through a t-connector to two 1.0 in pipes that were held rigidly by two machined aluminum brackets and placed at both of the humanoid's end-effectors. Screen-shots of the push-hold scheme are shown in Fig. 1.

Our work showed the viability of the proposed control strategy and successfully executed the assembly task under both roles and in the presence of jamming, wedging, and stiction phenomena. The rest of the paper is organized as follows: Sec. II describes the control basis. Sec. III presents primitive and compound controllers. Sec. IV presents experimental results. Sec. V, discusses interesting implications. Finally, Sec. VI presents key findings and future work.

II. THE CONTROL BASIS APPROACH

A control basis decomposes a complex control system into a set of modular control elements that when connected appropriately synthesize a variety of behaviors. A control basis consists of any number of closed loop controllers (that represent primitive actions) derived from a set of

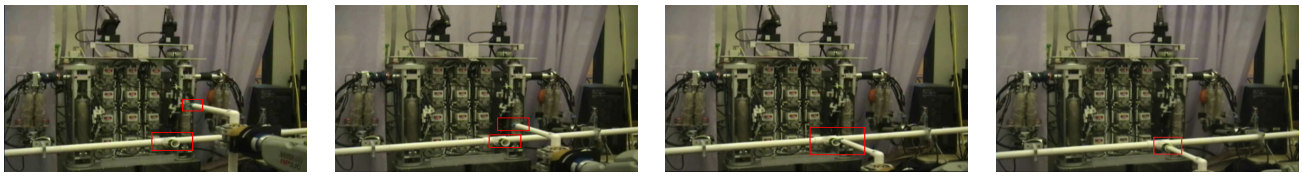


Fig. 1. Two heterogeneous robots perform a push-hold assembly.

control laws. As asserted by Huber [6], the control laws are designed to yield asymptotically stable and predictable behavior for different robots. That is, each carefully selected control law is designed to be robust under a wide range of conditions and largely independent of robot kinematics. Each control law discretizes the continuous space into discrete basins of attraction. The control law can compensate for a limited range of perturbations and uncertainties while still converging to the attractor. Similar approaches are found in the literature [7], [8], but the control basis framework is different in that it factors controllers into *objectives* and can combine any number of controllers through the use of nullspace compositions in any order to achieve a wide range of behaviors. A careful selection of a small set of control laws is realized as controller objectives implemented through the use of selected sensor and actuator resources to produce flexible structural solutions. In effect, the approach allows control elements to be re-used and generalized to different solutions depending on the context [9].

A. Mathematical Derivation

Primitive controllers ϕ_i, where $i = 1 \sim n$, are elements in a basis of controllers, Φ, such that $\phi_i \in \Phi$. A primitive controller optimizes a partitioned portion of a designated control space and can be understood as the minimization of a discrete basin of attraction. The basins of attraction are formulated through artificial potential functions defined over a typed domain, X_i, which are defined as the square of the error, where $\phi_i(\rho) = \rho^T \rho$ and ρ, is the difference between the reference input and the plant input, $\rho = \mathbf{q}_{ref} - \mathbf{q}_{des}$, at every time step.

Each controller reaches its objective by performing greedy descent, $\nabla\phi_i$, on the artificial potential function, while engaging sensor and motor resources. The minimization of the surface potential function in a specified domain space, X_i, over time is defined as:

$$\nabla_{x_i}\phi_i = \frac{\partial \phi_i}{\partial t}. \tag{1}$$

Each primitive is bound to a selected subset of input control resources $\gamma_j \in \Gamma_j$ and output control resources $\gamma_k \in \Gamma_k$ relevant to the task. In order to bind input and output control resources to the controller, corresponding sensor transforms, s_j, and effector transforms, e_k, are used. The sensor transform maps incoming sensory resources to a specified domain space such that $s_j : \Gamma_j \rightarrow X_l$. To ensure that a task is guaranteed to operate within the region of a corresponding basis we require that the output range of a sensor transform matches the artificial potential function domain's data type. Similarly, the effector transform maps the control law error's result to an appropriate output space, Y_k. The mapping is typically effected using a Jacobian matrix as in Equation 2.

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

where, $\mathbf{x}_{\gamma_i}$ represents the controller update for a sensor control resource γ_i. $\mathbf{y}_k$ is a corresponding point in the output space and $\Gamma_l = \gamma_1, \gamma_2, ..., \gamma_{|S_l|}$ is a subset of selected control resources for a given task. In order to match an effector transform with an artificial potential function, the rowspace of $e_k(\Gamma_l)$, X_i, must match the potential function's data type.

The closed-loop controller is implemented then, when the error between the incoming sensor information and the reference position is minimized within the discrete artificial potential basin, $\nabla_{x_i}\phi_i(\mathbf{x}_{ref} - s_j(\Gamma_j))$, and the gradient result is mapped to the output configuration space through an effector transform, $e_k(\Gamma_l)$. Given that the input data is of the same domain type as the artificial potential function, and the effector transform is of the same dimensions as the potential function, the controller's output, $\nabla_{y_k}\phi_i$, is defined as:

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

For convenience, the above expression is expressed in simplified notation as $\phi_i \,|_{e_k(\Gamma_l)}^{s_j(\Gamma_j)} (\mathbf{x}_{ref})$. If the controller has zero reference, then it can be omitted: $\phi_i \,|_{e_k(\Gamma_l)}^{s_j(\Gamma_j)}$.

To concurrently optimize multiple goals, secondary control updates are projected onto the nullspace of primary control updates. This relationship is expressed in a compound controller π as having the secondary controller ϕ_2 be *subject-to* the primary controller ϕ_1, and is expressed as:

$$\nabla_y(\phi_2 \triangleleft \phi_1) = \nabla_y\phi_1 + \mathcal{N}(\nabla_y\phi_1^T)\nabla_y\phi_2, \tag{4}$$

where,

$$\mathcal{N}(\nabla_y\phi_1^T) \equiv I - (\nabla_y\phi_1^T)^+(\nabla_y\phi_1^T), \tag{5}$$

and, *I*, is the identity matrix, *y* is an *n*-dimensional space, and the nullspace of $\nabla_y\phi_1^T$ is a (*n*-1) dimensional space orthogonal to the direction of steepest descent [10]. For convenience, Eqtn. (4) is written as $\pi_k : \phi_1 \triangleleft \phi_2$. Through nullspace composition techniques there is no need to specify how control resources will be shared across sub-controllers as long as the same control resources are used.

III. DEVISING A CONTROL BASIS FOR COOPERATIVE ASSEMBLY TASKS

According to [11], insertion assemblies require that the peg approximates the hole with a cautious motion until an

optimal location for insertion is achieved. A compliant insertion follows to correct misalignments. With this in mind, two compound controllers were designed to execute the push part of the scheme: a *Guarded Move Controller* and a *Compliant Insertion Controller*. The former controller was designed to reactively and autonomously displace the male truss to an optimum location for insertion. Upon reaching a good locale, the control policy transitions to the latter controller to drive the insertion task. ISAC also used a third controller designed to counterbalances the forces exerted by an incoming truss and is referred to as the the *Counterbalance Controller* π_{CB}.

These two compound controllers are formulated by combining position, ϕ_p, force residual, ϕ_{fr}, and moment residual, ϕ_{mr}, controllers in different order, with different references, and with different sensor and effecter transforms. The primitive controllers will be presented next and after that how they are compounded.

A. Control Basis Primitives

The following derivations apply to three 6 DoF manipulator with PUMA 560 configurations.

1) The Position Primitive: The position controller is based on the Jacobian transpose control method, where at each cycle, joint displacements are updated according to:

$$\triangle \mathbf{q} = J^T K_p \mathbf{e}, \tag{6}$$

where, $\triangle \mathbf{q} \in \mathcal{R}^{\mathbf{6x1}}$ is a displacement of joint angles, $J^T \in \mathcal{R}^{6x6}$ is the manipulator Jacobian, $K_p \in \mathcal{R}^{6x6}$ is the position gain, and $\mathbf{e} \in \mathcal{R}^{\mathbf{6x1}}$ is the error in cartesian positions.

For the position primitive, the sensor transform converts image coordinates to cartesian positions: $s_p(\gamma_{visual\_sys})$, while the effector transform maps the updated cartesian position to the robot's current joint configuration: $e_p(\gamma_{joint})$. The square of the cartesian error is used as the error function: $\phi_p = \frac{1}{2} K_p \mathbf{e}^T \mathbf{e}$, such that the gradient is:

$$\nabla_x \phi_p = K_p \mathbf{e}. \tag{7}$$

The basis controller can thus be defined as:

$$\nabla_q \phi_p = e_p(\Gamma_{joint})^T \nabla_{x_p} \phi_p (\mathbf{x}_{ref} - s_p(\Gamma_{visual\_sys})), \tag{8}$$

or, more succinctly as $\phi_p \mid_{e_p(\Gamma_{joint})}^{s_p(\Gamma_{visual\_sys})} (\mathbf{x}_{ref})$.

2) The Force and Moment Primitives: Two controllers, force and moment primitives, update joint angle configurations so as to apply desired forces or moments. The force controller updates the first three joints configurations of both robots 6 DoF, while the moment controller updates the last three joint configurations. The joint angle updates for both control law are defined as:

$$\triangle \mathbf{q_{1-3}} = K_j^{-1} J^T K_f (\mathbf{f}_{ref} - \mathbf{f}), \tag{9}$$

$$\triangle \mathbf{q_{4-6}} = K_j^{-1} J^T K_m (\mathbf{m}_{ref} - \mathbf{m}), \tag{10}$$

where, $(\mathbf{f}_{ref} - \mathbf{f})$ and $(\mathbf{m}_{ref} - \mathbf{m})$ are the force and moment errors conformed by the first three and last three elements in a $\mathcal{R}^{6x1}$ vector respectively; and, K_f and K_m are the first three and last three force and moment gains along a diagonal matrix in $\mathcal{R}^{6x6}$ that multiplied by the Jacobian transpose $J^T \in \mathcal{R}^{6x6}$ generate torque updates for the appropriate joint configurations. The inverse of $K_j \in \mathcal{R}^{6x6}$ and the other gains are precomputed to generate corresponding joint angle update for each cycle of the force or moment controller.

The force and moment residual controllers have sensor transforms $s_f(\gamma_{force})$ and $s_m(\gamma_{moment})$ that return the F/T sensor data respectively. The artificial potential functions for the force and moment residual functions are proportional to the square of their errors:

$$\phi_{fr} = \frac{1}{2} K_f (\mathbf{f}_{ref} - \mathbf{f})^2, \quad \phi_{mr} = \frac{1}{2} K_m (\mathbf{m}_{ref} - \mathbf{m})^2, \tag{11}$$

and they are differentiated with respect to their joint angle configurations to displace the trusses and minimize residuals:

$$\nabla_q \phi_{fr} = -K_f (\mathbf{f}_{ref} - \mathbf{f}), \quad \nabla \phi_{mr} = -K_m (\mathbf{m}_{ref} - \mathbf{m}). \tag{12}$$

The controllers also have effector transforms $e_{fr}(\gamma_{torque})$ and $e_{mr}(\gamma_{torque})$ that converts torque updates into joint angle updates by multiplying the inverse position gains and Jacobian transpose

$$e_f r(\Gamma_{torque_k}) = K_j^{-1} (J_{\gamma_1}, ..., J_{\gamma_{|\Gamma_k|}})^T \tag{13}$$

$$e_m r(\Gamma_{torque_l}) = K_j^{-1} (J_{\gamma_1}, ..., J_{\gamma_{|\Gamma_l|}})^T \tag{14}$$

to produce the following primitive controllers:

$$\phi_{mr} \mid_{e_m(\Gamma_l)}^{s_m(\Gamma_l)}:$$
$$\nabla_q \phi_{mr} = e_m(\Gamma_l)^T \nabla_m \phi_{mr} (\mathbf{m}_{ref} - s_m(\Gamma_{moment})). \tag{15}$$

$$\phi_{fr} \mid_{e_f(\Gamma_{torque})}^{s_f(\Gamma_k)}:$$
$$\nabla_q \phi_{fr} = e_f(\Gamma_{torque})^T \nabla_f \phi_{fr} (\mathbf{f}_{ref} - s_f(\Gamma_{force})). \tag{16}$$

B. Compound Controllers

1) Guarded Move Controller: The guarded move controller π_{GM} uses ϕ_p as the dominant controller and ϕ_{mr} as the subsidiary controller. The hierarchy of π_{GM} was decided empirically by prioritizing the need for an optimal insertion location. The position controller displaces the truss to a reference location (x_{ref}) and a subordinate moment controller minimizes any unexpected disturbances by realigning the truss while it optimizes the greedy path trajectory. The position controller receives a 3D reference cartesian position from a stereo visual system that detects color fiducial marks placed at the fixture's tips [12]. The compound controller, π_{GM}, is then implemented as:

$$\pi_{GM} =$$
$$= \phi_{mr} \mid_{e_{mr}(\gamma_{torque})}^{s_{mr}(\gamma_{moment})} (\mathbf{m}_{ref}) \triangleleft \phi_p \mid_{e_p(\gamma_{joint})}^{s_p(\gamma_{visual\_sys})} (\mathbf{x}_{ref}). \tag{17}$$

2) Compliant Insertion Controller: The compliant insertion controller π_{CI} minimizes residual moments and forces experienced during the assembly's insertion stage. As stated earlier, aligning trusses takes precedence over fixing its position during the insertion stage. Such alignment is performed by the minimization of moment errors with zero

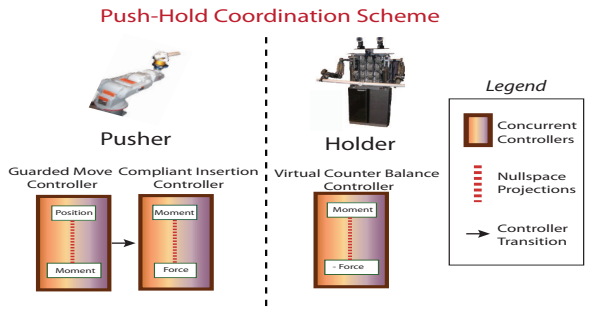


Fig. 2. Concurrent and sequential controllers used for cooperative assembly.

reference moment. For this reason, the moment residual primitive ϕ_{mr} is the dominant controller, while a force residual controller ϕ_{fr} acts as the subordinate controller. As part of the implementation of a reactive and autonomous push-hold scheme we decided to design the force reference parameter (f_{ref}) in the subordinate primitive to be the the driving parameter for the push factor in the assembly. To do so, it uses a reference value less than or greater than zero, depending on the robot, to drive the insertion (see Sec. IV. The compliant insertion controller is defined in Eqtn. 18. An illustration of the assembly is shown in Fig. 2.

$$\pi_{CI} = \phi_{fr} \,|^{s_{fr}(\gamma_{force})}_{e_{fr}(\gamma_{torque})}\, (\mathbf{f}_{ref}) \triangleleft \phi_{mr} \,|^{s_{mr}(\gamma_{force})}_{e_{mr}(\gamma_{torque})} \,. \quad (18)$$

3) Counterbalance Controller: As part of the push-hold schemes, a composite controller was devised to implement a force guided system that would maintain the static fixture's position in place while updating its orientation. Following the design evidence of the compliant insertion controller, the counterbalance controller, π_{CB}, seeks to facilitate the insertion process and is composed of a dominant moment residual controller and a subordinate force controller:

$$\pi_{CB} = \phi_{fr} \,|^{s_{fr}(\gamma_{force})}_{e_{fr}(\gamma_{torque})}\, (\mathbf{f}^{*}_{\mathbf{ref}}) \triangleleft \phi_{mr} \,|^{s_{mr}(\gamma_{force})}_{e_{mr}(\gamma_{torque})} \quad (19)$$

Though similar to the compliant insertion controller, π_{CB}, opposes the incoming truss, whenever there is contact, by applying an equal but opposite force relative to the incoming force, $\mathbf{f}^{*}_{\mathbf{ref}}$. In so doing, it displaces the end-effectors so as to create an optimal entry angle for the mating truss. In this way, it minimizes moment and force residuals throughout the task. A similar version of the three prior controllers was adapted to be used with ISAC [12].

The reactivity of the controllers is key to successfully deploy robots of different morphologies in a cooperating team under different roles. The difference between robots lies in the gain values for the primitive control and the PD joint controller of each robot (Sec. V). Note that the male truss tool center point (TCP) lies at the truss' tip, while the female truss TCP lies at the center of the outer ring.

IV. EXPERIMENTS

The push-hold task was set-up so that the HP3JC truss was positioned in front of the female truss through a mobile platform. The male truss may offset from the female truss, rigidly held by ISAC, in all three planes as long as it remains in the workspace. Trials in the experiment used random locations to simulate a semi-structured environment. Color segmentation and image processing were used to compute the cartesian coordinates for both male and female trusses' [12]. Nine trials were run. Six were successful. Three failed due to slightly-off reference cartesian coordinates from the visual system.

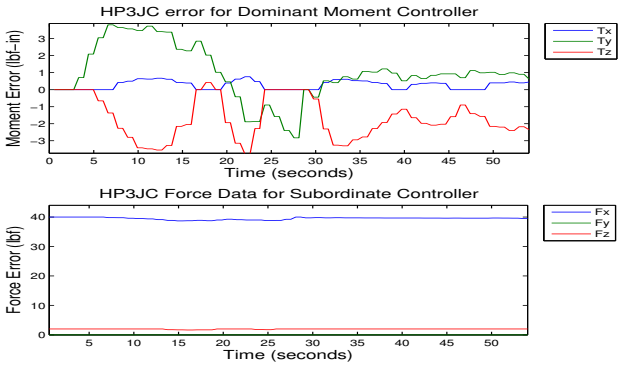
The control policy effected use the π_{GM} and π_{CI} controllers for the HP3JC robot. The visual system triggered the HP3JC's motion by providing a reference position to the guarded move controller. ISAC remained at the home position using π_{CB} and optimized the orientation and pose of it's writs to optimize the male truss entry. The compliant insertion controller used a reference force of Fx=20 lbs (slower insertions) for three trials and Fx=40 lbs (faster insertions) for three trials.

Additionally, three metrics were used to measure the assembly tasks' performance: (a) time-to-completion, (b) the sum of the absolute value for maximum moment residuals in the x-, y-, and z-directions (referred to hereafter as "moment errors"), and (c) the reference force parameter. The latter produced faster and slower insertions driven by the industrial robot. Note that an insertion was assumed complete after the male fiducial mark is covered and that the end-point of the male truss never contacts the interior back wall of the female truss. The authors would like to highlight that our decision to select this rule as a *complete insertion*, in retrospect, was not the best choice. When we consider that the male truss did not hit the back of the female truss wall and that the radius of the female truss is twice as large as the male truss, we created a scenario where successful insertions take place even though the controller has not fully converged. Nonetheless, the authors have chosen to publish the data in its current format since they are representative of successful insertions and generally converging profiles albeit not all of them. The authors will comment on this whenever the data shows such situations.

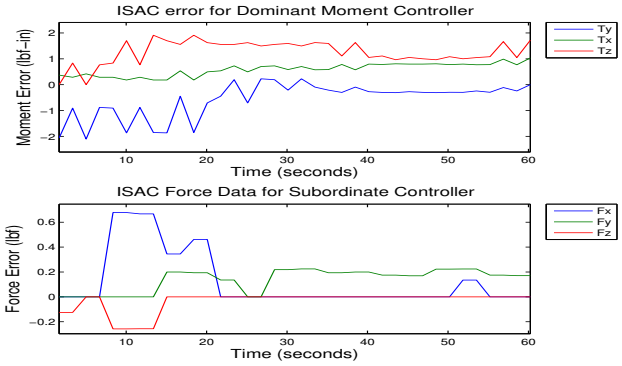
A. Experiment 1: HP3JC-Push, ISAC-Hold

In the first demonstration, the industrial robot drove a male truss into a female fixture held by ISAC. The HP3JC robot used the π_{GM} and the π_{CI} controllers to actively perform the insertion, while ISAC used the π_{CB} controller to counteract, but optimize entry forces exerted by the industrial robot. The tool center Note the reference parameter for the compliant insertion controller—a force reference—is what enacts the forward motion of the truss. In other words, this parameter affects the speed and force of the insertion. Force sensing, in this sense, drives the insertion, while adjusting the position of the truss in the vertical and horizontal planes.

Sensory data for one trial is presented for both robots in Fig. 3(a) and Fig. 3(b) in two sub-plots representing the: (i) the moment residuals, and (ii) the force residuals. The 54 second duration of this and other experiments may seem significant. The reason for such durations was the inability to



(a) Force data for the HP3JC-pushing robot



(b) Force data for the ISAC-holding robot.

Fig. 3. Exp 1: Force signatures for the HP3JC robot and ISAC.

access the industrial robot's low-level control loop. The impediment forced us to operate through the industrial robot's API which prevented pre-emptive motion and delayed the overall response of the system. With respect to moments, the residual in the y- and z-directions were reduced by both robots. In the z-direction a small residual remained in both robots indicating the approach by the truss also contained a horizontal displacement. Small residual moments in the y- and z- directions were expected. The compliant insertion controller is reactive in nature and rejects disturbances as they appear. Given that the tolerance in our experimental set-up is of 0.25 in, trusses never align perfectly, giving rise to a zig-zag motion of the male truss as it entered the female truss. Additionally, for ISAC, the counter balance controller seeks to eliminate force errors by adjusting the female fixture position. Forces in the x- and z-directions converged to zero. The y-direction residual force error is a response to the presence of moment in the z-direction from the male truss, which had not fully converged by the time the assembly was finalized.

A summary of metric results across trials is shown in Fig. 4. The first three trials were run under the slower force parameter Fx=20. With respect to moment residuals the HP3JC robot experienced residuals greater than ISAC. The last five experiment trials used the faster force parameter Fx=40 producing generally larger moment errors for both robots. Two of the first three trials contained larger than

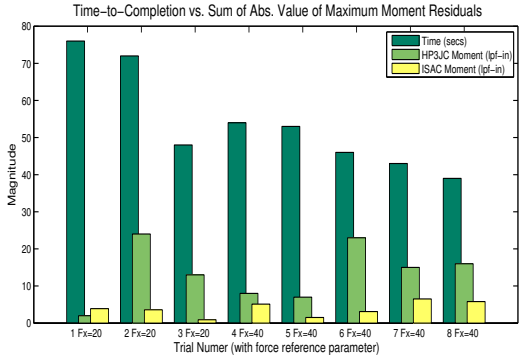


Fig. 4. Exp.1: Results summary across 8 random trials.

normal moment errors due to one wedging in one trial and jamming in another.

B. Experiment 2: ISAC-Push and HP3JC-Hold

In this experiment ISAC drove the assembly. This experiment began with the HP3JC holding the male truss at a ready-position in front of the female fixture held by ISAC. The humanoid's compliant insertion controller, π_{VCI} was responsible for driving the insertion. ISAC used a force parameter of Fx=-0.5 lbs in its compliant insertion controller. The results are shown in Fig. 5. A slower speed was selected for the pneumatic robot to achieve greater accuracy. From clock time 1–200 seconds, two important patterns are present in the data: a) there is a roughly constant increment in the moment residual error in the y-direction, and b) the force reference force is gradually canceled over this period. Both patterns indicate the existence of stiction in the task. This is further corroborated by the quick fall seen in the left torque reading in the y-direction and the quick increase in the force reference value Fx in the negative direction. The controllers are unable to decrease residuals caused by stiction until the corrective force generated by the controllers overcome the sticking forces present through the artificial muscles. After the sticking forces were overcome, the insertion took place quickly. The sudden completion of the task did not allow the compound controllers enough time to converge back to their

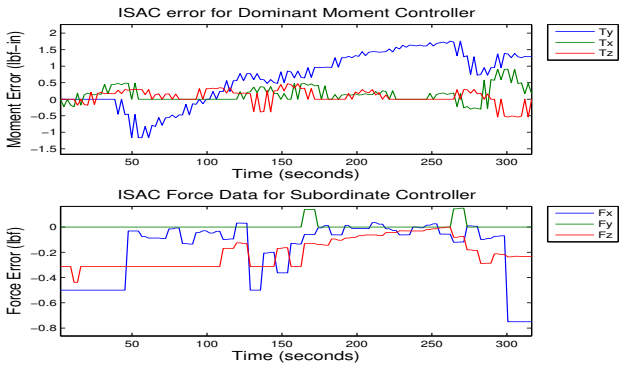


Fig. 5. Exp 2: Quick changes in torques due to stiction effects caused by ISAC's compliant nature.

reference goals.

For ISAC the averaged moment errors generated in this experiment were greater than in Exp. 1. ISAC experienced greater stress as it drove the insertion than when it used π_{CB}. On the other hand, the averaged moment residual errors experienced by ISAC are lower than the ones experienced by the HP3JC robot. This suggests that the use of artificial muscles eases impact or stress induced as compared to rigid industrial robots. A summary of the results across six trials is found in Fig. 6. Three trials experienced longer times-to-completion due to more prominent stiction phenomena.

V. Discussion

The multi-agent architecture encapsulated low- and high-level abstractions and worked in concert with the control basis framework to modularize the control problem. By selecting well defined control laws, compound controllers for manipulation, insertion, and counterbalancing were deployed in two robots of very different morphologies. The controllers converged reactively for both roles even in situations where stiction was present. The results are promising. Nonetheless, the setup for this experiment was not simple, requiring loading and running tens of agents across multiple computers [12]. These agents were event-driven and responded to sensory information in the environment. Six of the nine experiments failed due to color segmentation inaccuracies. Also the speed of the assemblies was limited by difficult to control pneumatic actuators and inaccessibility to the industrial robot's low-level control.

While there has been extensive research on assembly for the past four decades [12], to date there are no examples of humanoids working autonomously doing assembly in semi-structured environments with another robot of different morphology and attributes. Several researchers have developed adaptive and flexible controller techniques to attenuate the limitations posed by monolithic controllers and unstructured environments [6]. However, these techniques have been executed with planners and only with teams of homogeneous robots [13], [14].The control basis approach allowed reactive autonomous complex behavior by the way it abstracts the control problem. Still, in this work gains were selected manually and behaved similarly to Natural Admission Control whereby there was a trade-off between torque controller gains and velocity gains. Lower torque gains provide the system with more stability but less responsiveness. Increasing torque gains and lowering velocity gains increase stiction in the task. In this sense, the control primitives suffer a limited response and contributed to the accumulation of error in the tasks.

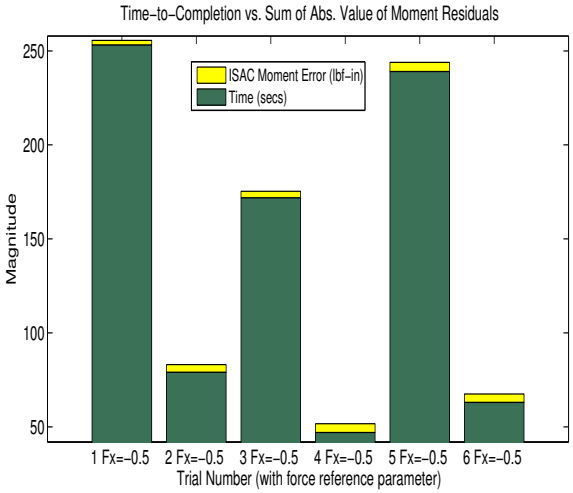


Fig. 6. Exp.2: Summary for results across 6 random trials.

VI. Conclusion

This paper presented a team of heterogeneous robots performing an autonomous and collaborative assembly task that is grounded on a distributed multi-agent architecture and modular control basis approach. The modularity and the flexibility of the agent-based architecture and the control approach worked in concert to bootstrap robust, flexible, and decidedly reactive controllers. Experimental results concluded that multi-agent multi-robotic collaborative systems with modular control approaches is a viable approach to generate complex robotic behavior.

VII. Acknowledgments

This work was supported by NASA grant NNX07AF04G.

References

[1] T. Brogardh, "Present and future robot control development–an industrial perspective," *J. of Annual Reviews in Control*, vol. 31, no. 1, pp. 69–79, 2007.

[2] F. Rehnmark, R. Ambrose, B. Kennedy, M. Diftler, J. Mehling, L. Bridgwater, N. Radford, S. Goza, and C. Culbert, "Innovative robot archetypes for in-space construction and maintenance," *AIP Conf. Proc.*, vol. 746, no. 1, pp. 1070–1077, 2005.

[3] J. Kramer and M. Scheutz, "Development environments for autonomous mobile robots: A survey," *Autonomous Robots*, vol. 22, no. 2, p. 101132, 2007.

[4] Motoman, http://motoman.com/, 2011.

[5] J. Rojas and R. A. Peters II, "Sensory integration with articulated motion on a humanoid robot," *J. of Applied Bionics and Biomechanics*, vol. 2, no. 3-4, pp. 171–178, 2005.

[6] M. Huber, "A hybrid architecture for adaptive robot control," Ph.D. Dissertation, U. of Mass., Sept. 2000.

[7] J. Jameson and L. Leifer, "Automatic grasping: An optimization approach," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 17, no. 5, pp. 806–813, Sept. 1987.

[8] J. Son, R. Howe, and G. Hager, "Preliminary results on grasping with vision and touch," *IEEE Conf. on Int. Robots and Systems*, vol. 3, pp. 1068–1075, Nov. 1996.

[9] O. Brock, A. Fagg, R. Grupen, R. Platt, M. Rosenstein, and J. Sweeney, "A framework for learning and control in intelligent humanoid robots," *Intl. J. of Humanoid Robots*, vol. 2, no. 3, pp. 301–336, 2005.

[10] R. J. Platt, "Learning and generalizing control-based grasping and manipulations skills," Ph.D. dissertation, U. of Mass. Amherst, 2006.

[11] H. Inuoe, "Force feedback in precise assembly tasks," *In Artificial Intelligence: An MIT Perspecrive*, vol. 2, pp. 219–241, 1981.

[12] J. Rojas, "Autonomous cooperative assembly by force feedback using a control basis approach," Ph.D. dissertation, Vanderbilt U., 2009.

[13] M. Namvar and F. Aghili, "Adaptive force control of robots in presence of uncertainty in environment," *2006 American Control Conf.*, pp. 3253–3258, June 2006.

[14] P. Yuan, "An adaptive feedback scheduling algorithm for robot assembly and real-time control systems," *IEEE Intl. Conf. on Int. Robots and Systems*, pp. 2226–2231, 2006.