2015 IEEE International Conference on Robotics and Automation (ICRA)
Washington State Convention Center
Seattle, Washington, May 26-30, 2015

Mobile Manufacturing of Large Structures

David Bourne, Howie Choset, Humphrey Hu, George Kantor,
Chris Niessl, Zachary Rubinstein, Reid Simmons, Stephen Smith
Robotics Institute, Carnegie Mellon University

Abstract— **Assembly of large structures requires large fixtures, often referred to as monuments. Their cost and massive size limit flexibility and scalability of the manufacturing process. Numerous small mobile robots can replace these large structures and, therefore, replicate the efficiency of the assembly line with far more flexibility. An assembly line made up of mobile manipulators can easily and rapidly be reconfigured to support scalability and a varied product mix, while allowing for near optimal resource assignment. The challenge to using small robots in place of monuments is making their joint behavior precise enough to accomplish the task and efficient enough to execute subtasks in a reasonable period of time. In this paper, we describe a set of techniques that we combine to achieve the necessary precision and overall efficiency to build a large structure. We describe and demonstrate these techniques in the context of a testbed we implemented for assembling a wing ladder.**

I. INTRODUCTION

A popular approach in industry for assembling large-scale structures is to establish assembly lines that use large, mounted fixtures to provide precise positioning of parts, relative both to other parts and to the tools used in the assembly. While these lines are efficient, the permanence of these fixtures requires costly infrastructure and results in an assembly space that is not adaptable to changes in demand for different product lines. In addition, the lines are vulnerable to issues that arise during assembly and frequently result in the entire line being delayed until they can be resolved. To address these drawbacks, we are investigating a new approach to assembly where the fixtures and the tooling are all onboard coordinated mobile robots. The mobility of the robots allows for rapid reconfiguration of the assembly space, which means better adaptability to meet changing product demand and the ability to work around line issues.

The primary challenge to replacing mounted fixtures with tooling on small mobile robots is achieving the precision required to reliably assemble the products. Positioning the mobile robots and the parts to millimeter accuracy is difficult due to the uncertainty in both the environment and the controllers of the robots. In addition, complex coordination is needed to get the robots, the parts, and the tools at the right place at the right time for each step in the assembly.

We have built a demonstration facility to explore the efficacy of assembling large structures with a number of heterogeneous mobile robots. We have developed a sequence of coordinated behaviors that accomplishes an example assembly task by combining four component techniques: i) mobile manipulator hardware design, ii) fixture-free positioning, iii) multi-robot coordination, and iv) real-time dynamic scheduling. In this paper, we present each of these techniques, describe how they contribute to the precision and efficiency of assembly, and discuss how the techniques can be sequenced to achieve precise, efficient assembly of large-scale structures.

The long-term goal of this work is to create a rigorous framework for rationally sequencing these techniques to achieve the necessary precision for assembly in the most efficient sequence (see Figure **??**) . After working through the details of our current multi-robot assembly example, we propose a generalized framework to accomplish similar tasks that employs a "funnel" [12] or sequential composition [1] concept in which multiple controllers are sequenced together to mitigate uncertainty.

II. RELATED WORK

Assembly planners create a graph [26], [27] that encodes the order of motion operations and the geometric constraints ensuring that each motion can be executed. Typically, the part must move in contact with other parts in the subassembly, so an assembly can be viewed as a sequence of compliant motions [13] where a transition from one compliant motion to another occurs at a change in contact state. Often, a planner uses a graph, called a *contact state graph,* as well as knowledge of the parts' geometries to determine the sequence. [11], [14], [6]. Much effort has been dedicated to automatically generating the contact state graph [7], [15], [28].

Regardless of the choice of sequence planner, the overall system must be able to accurately align the part for final assembly, which is often quite difficult when the part's pose has uncertainty. Here, compliant motions can be used to minimize this uncertainty through a sequence of compliant motions [12], [17]. In our work, we use compliance (i.e., springs) to achieve compliant and force-controlled execution and assist in part localization. Previous work has used passive compliance to account for misalignment between a robot and task constraints [24]. Series elastic actuators [18] introduced compliance to reduce shock loads and enhance force control accuracy and stability. We have found from our previous research that passively compliant mechanisms can improve precision without losing performance. If a part contacts a structure prematurely, it will comply to the shape of the contact area without causing damage. By taking advantage of known features in a given structure, passive compliance is both a precise and fast operation to position a part.

978-1-4799-6922-7/15/$31.00 ©2015 IEEE

Our work to date has assumed the existence of a hierarchical assembly planning graph, and this process description, which dictates both the order in which parts should arrive for assembly and the coordinated sequence of actions that must be carried out to successfully join specific parts, has driven the design characteristics of the robots required to execute assembly subsequences. The robots that we created are sometimes referred to as mobile manipulators because they combine aspects of both mobile robots and manipulator mechanisms [22]. While some systems [9] use multiple mobile manipulators to manipulate a single object, we are not aware of a multi-mobile manipulator system that was designed for assembly. As such, prior multi-robot mobile manipulators typically assume that all poses of the robot is known at all times, at least with respect the mobile bases.

We believe new mobile manipulators must be created because existing systems either contain more degrees of freedom than necessary or lack the payload to carry out the assembly task. This is because mobile manipulators were not originally conceived to carry out coordinated assembly. To design these new systems, our work is inspired by i) the tool-design for assembly work [25], which developed a tool representation that includes the tool's use-volume, i.e., the minimum space that must be free in an assembly to apply the tool, and ii) [5], which presented a detailed approach for modeling tools, selecting tools, and generating tool movements. Their work allowed tools to be modeled as articulated devices.

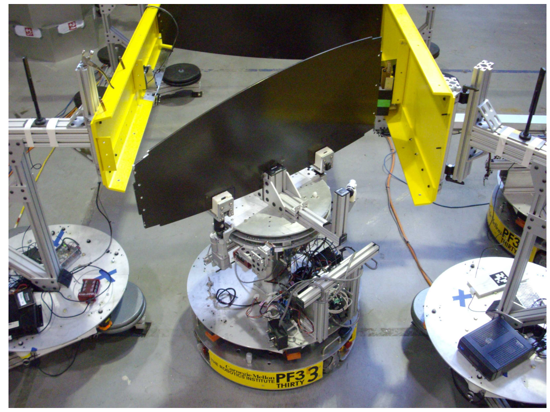

Fig. 1. The yellow spars support the black ribs forming the ladder. Here, a rib-carrying robot is bringing a rib into its final position.

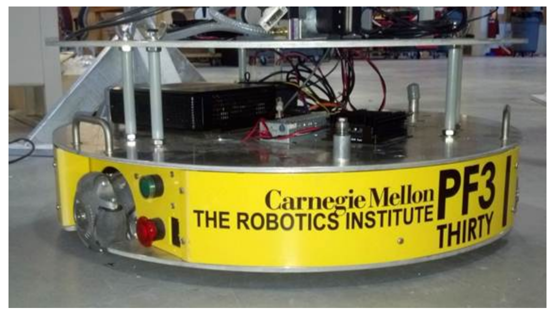

Fig. 2. Omnibases

III. Modular Mobile Manipulator Robot System

The work described in this paper uses assembly of airplane wings as a motivating application, but we believe the ideas are general to assembly of many large structures such as ships, large construction vehicles, and trains. To date, we have focused on two tasks: coordination of two robots to insert ribs into a partially-assembled wing ladder and sensor-based coordinated motion of two robots to carry a flexible skin panel. The rib-insertion task and larger ladder assembly process (the focus of this paper) involve moving ribs (the cross-wise pieces) into place between two spars (the long length-wise pieces). In our testbed, the spars are 10 feet long and accommodate 5 ribs; the ribs vary in length (the spars are not parallel) but average about 3.5 feet and weigh about 20 pounds. The spars have L-brackets spaced every two feet, to which the ribs need to be attached. There are three 3/8 inch diameter holes on either end of the rib that must be aligned with corresponding holes on the L-bracket. The alignment tolerance is about a millimeter.

The robots are 30-inch diameter, 3-wheeled omnidirectional bases, designed at CMU. Each robot carries a manipulator module specialized for the role it plays in the assembly task (examples of these mobile manipulators can be seen in Figures 1, 2, 3, and 4). The bases, which have payloads of several hundred pounds, all have on-board computation, communications, sensors, and power to support themselves and the manipulator modules they carry. Sensors used in the current system include accelerometers, cameras, lasers, potentiometers and limit switches.

The behaviors, algorithms, and framework developed to support the rib-insertion task is the focus of the rest of this paper. The task begins by having the rib-carrier robot approach the end of the wing ladder assembly, carrying a rib length-wise (Figure 1). The robot rotates to enable it to move between the spars with sufficient clearance. Concurrently, the spar-catcher robot aligns a temporary fixture (Figure 3) with the spar cap and then slides along the spar cap until it detects the L-bracket (via a limit switch), at which point it engages a brake to provide a rigid target for the rib. The rib carrier then approaches the target using a guarded move, i.e., a move that stops when the "guard" is triggered. The "guard" sensor is an accelerometer attached to the target. The accelerometer signal is processed by the spar-catcher robot, which communicates with the rib carrier to trigger the "guard." Passive compliance is used to handle the small amount of movement after contact, due to latency caused by the signal processing and communications.

Having hit the target, the system uses a camera mounted on the rib carrier to determine where the rib is along the target (by calculating the percentage of the target not obscured by the rib). The rib carrier backs up a few centimeters, rotates to point at the corner where the L-bracket meets the spar, and then drives towards that point until, once again, the target-mounted accelerometer indicates that contact has been made.

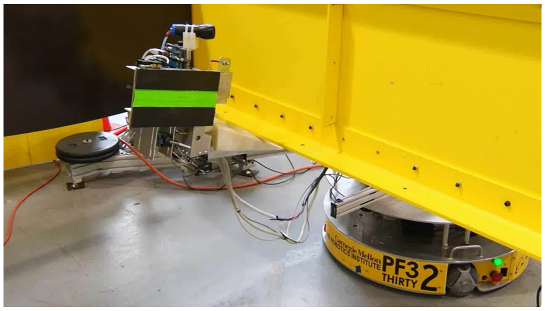

Fig. 3. Spar catcher, on the other occluded side of the yellow spar, is about to engage with the sparcap in order to register the black and green feature against the L-bracket in the spar.

The rib carrier then moves in an arc to “close the door,” stopping when a limit switch on the target is contacted, which indicates that both ends of the rib are now flush with the L-brackets. Again, the passive compliance in the rib carrier module (aluminum disks below the rib carrying mechanism in Figure 1) enables the system to compensate for small amounts of overshoot.

Finally, the system aligns the holes in the rib and L-bracket. Two cameras mounted on the spar catcher head provide high-resolution images of two of the holes in the L-bracket. The system collects background images of the holes prior to the arrival of the rib carrier and compares them against images taken with the rib in place to determine the degree and direction of overlap between the bracket holes and the rib holes. The data from both images are used to determine how to move a 4-bar linkage that supports the rib (Figure 4) in order to reduce the overlap. This servoing continues until the holes are adequately aligned, at which point a person can insert cleco fasteners without any resistance. The robot then lowers the tooling that holds the rib (which mechanically unlocks the fingers that hold the rib in place), and it drives out of the wing ladder to receive another rib and begin the task anew.

The broader problem of moving parts (ribs and spars) by robots from storage areas to assembly stations to achieve an efficient overall ladder assembly process is considered in simulation (Figure 5). Movement tasks are dynamically allocated to robots as assembly of wing ladders proceeds at multiple assembly stations. Robot movements are scheduled so that parts arrive just in time to support their insertion, thus minimizing overall assembly time while also minimizing traffic congestion.

The rib insertion assembly sequence and broader ladder assembly process combines the four component techniques presented in the introduction. Global and relative positioning is used to move the robots into positions where they are ready to begin the assembly task. The temporary fixture and the robot compliance that are used in the guarded move step employ both passively compliant hardware and actively compliant sensing and control. The execution of the individual steps and the logic to switch between them represent intelligent behavior assembly strategies. Component behavior assembly

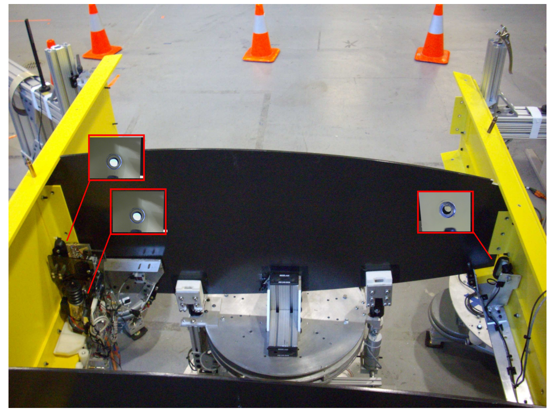

Fig. 4. Final visual servoing step to align holes.

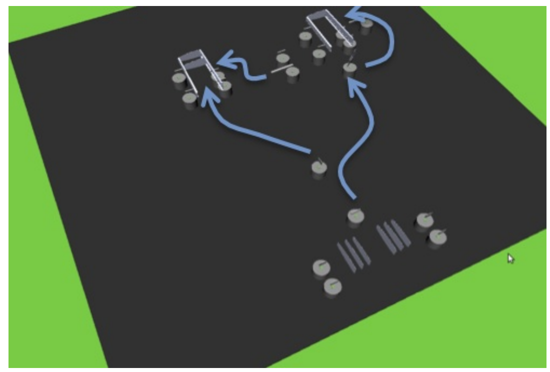

Fig. 5. Simulation of the broader ladder assembly process.

strategies are assigned to robot teams and coordinated in space and time using real-time, dynamic scheduling. Starting with the mechanical components, each of these techniques is described in more detail in the following sections.

A. Mechanism Design: Mechanical Features and Passive Compliant Devices

1) Omni Robot Bases (PF3-30): Each mobile manipulator is built upon the same mobile base. The bases are 30 inches in diameter and are driven by three omni-directional wheels that can currently carry up to 300 pounds each. Omni-directional robot are needed because the mobile manipulators require enhanced maneuverability in tight spaces as the assembly is being formed. Each robot base is divided into four distinct layers: (1) the motor controllers and power modules including lithium batteries, (2) the computational layer, which is based on high speed Intel quad-i7 CPUs, (3) network connections and sensor preprocessing modules, and (4) task-specific tooling. The first two layers are common among all of the robots, while the third and fourth layers are customized according to the assigned role of a particular robot.

In addition to this layered robot architecture, there are several hardware extensions that are used for global localization. This localization hardware consists of a sensor package and a fiducial harness designed to be easily sensed by a matching

Fig. 6. Spar catcher.

sensor package on the other robots. The current localization hardware is identical between robots, allowing us to use a single harness design, but the system can be extended to support any number of different harnesses. The details of the harness and sensor package are the following:

a) *Fiducial Harness:* The fiducial harness is a set of panels mounted to the robot to form a hexagonal profile. A hexagonal profile was chosen as a compromise between maximizing the number of distinct corners and flat edges viewable at a time, while minimizing wasted space on the round PF3-30. A unique AprilTag [16] fiducial is mounted on each flat segment of the harness for each robot.
b) *Sensor Package:* The sensor package comprises planar laser rangefinders and cameras. We use one Hokuyo UTM-30LX (270° FOV) planar laser rangefinders and one Microsoft LifeCam USB webcam (70° FOV), though this number could be increased for greater sensing coverage. The sensors are mounted at the corners of the fiducial harness hexagon, with unfilled sensor mount points occupied by sensor dummies to avoid gaps in the profile. Assuming that the environment has relatively flat floors, the sensor package senses the horizontal profile of each robot at the elevation of the fiducial harness. Feedback from the PF3-30's wheel encoders are also used in the localization system.

2) Spar Catcher: The spar catcher is a PF3-30 base on which is mounted a long compliant arm and a compliant gripper coupled with a complex fixture and sensor package (Figure 3). In essence, this robot is an intelligent hook that can blindly attach itself to a spar and then align itself by several simple discrete motions: (1) pull back to square up gripper under compliance, (2) move left to contact assembly feature, and (3) lock position with a braking mechanism.

The main role of the spar catcher is to emplace a temporary fixture (called a "rib target") that is used to help position the rib. The spar catcher also acts as a sensor platform to detect ribs when they approach and assist in the final visual-servoing alignment step.

3) Rib Carrier: The rib carrier is a PF3-30 base on which is mounted a compliant rotary table that provides softness in the yaw dimension of a gripped rib. In turn, the table is connected by four compliant springs to provide softness in the two remaining degrees of orientation (pitch and roll).

With this arrangement of compliant members, the rib carrier can both manage and account for uncertainties in the flatness of the manufacturing floor and the relative flatness of the spars and the to-be assembled rib. However, the greatest range of compliant motion is in the rotary table, which manages and accounts for potentially gross uncertainty in the robot's orientation relative to the targeted assembly. If the robot is misoriented, the rotary table simply winds up under contact and measures the mis-orientation for corrective motions.

The rib carrier also includes an active four-bar linkage, driven by three motors (Figure 4) to do very fine positioning of the ribs. Two of the motors lift and orient the rib (in the plane of the rib), while the third motor is used to move the rib side-to-side.

B. Fixture-free Positioning: Relative and Global Positioning

Robots cooperatively carrying large components cannot be localized with overhead vision systems due to occlusion, and the density of robots near the workspace may make it difficult for robots to localize using environmental landmarks. Instead, we opt for an "environment-agnostic" approach where robots sense only other robots. In our approach, robots collect sensor data and process it onboard to identify and track neighboring robots. The resulting relative pose estimates are sent to a central server, where they are assembled into a pose graph that is optimized to generate a consistent estimate of all robot poses in a common global coordinate frame.

Data from the laser scanner and camera pass through various stages and produce relative position measurements to detected fiducial harnesses, which are fused in a local EKF to produce a relative pose measurement for each observed robot. AprilTag software[1] is used to identify and measure relative poses to AprilTag fiducials fixed to the fiducial harnesses. Laser scanner point clouds are segmented, and iterative closest point (ICP) scan matching is used to attempt to fit segments to the known horizontal geometry of the fiducial harness. When sufficient matches are found, ICP also provides a relative pose measurement of the observed robot. Laser and camera pose measurements are fused with robot odometry in a local EKF, providing single relative pose estimate for each observed robot. The estimates are sent to a central server along with odometry information where they are combined with relative pose estimates and odometry from other robots to form a multi-robot pose graph as shown in Figure 8. Globally consistent pose estimates for all robots are generated using the g2o graph optimization package [10].

[1] http://people.csail.mit.edu/kaess/apriltags/

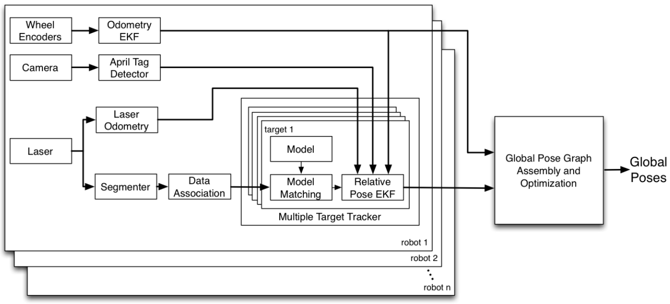

Fig. 7. Block diagram illustrating pipeline for localization system. Camera and laser data are locally fused into relative pose estimates based on observations between robots, and the results are relayed to a central server where they are assembled into a pose graph, which is optimized to produce globally-consistent pose estimates of all robots in the system.

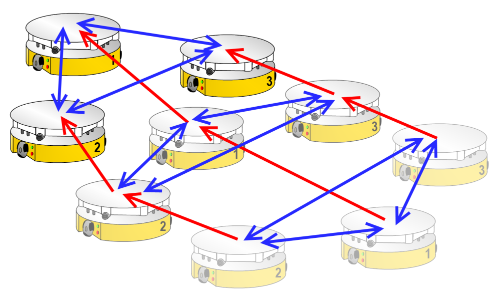

Fig. 8. Illustration of how relative pose estimates and odometry measurements are fused into a multi-robot pose graph. Red arrows represent odometry measurements for individual robots. Blue arrows represent relative pose measurements generated by robots observing one another. These pose estimates and measurements become edges in the pose graph, while the poses of the robots at times when observations were made are the pose graph nodes.

C. Multi-robot Coordination: Coordination at the Micro-Level

Assembly planning is done using a geometric model of the wing parts and how they are supposed to connect together, plus a rough model of the uncertainty in motion and positioning of the robots. For instance, the amount the rib carrier initially turns before moving between the spars is calculated from the length of the rib being carried, the minimum clearance between the spars up until the point where the rib is to be inserted, and the uncertainty in the global positioning system (plus some padding). Similarly, the arc that the robot should take to "close the door" is calculated based on the robot's current pose vis-a-vis the right-side L-bracket, the length of the rib and the distance between the two spars. This enables the system to emplace spars anywhere along the ladder using the same algorithm and, in fact, enables the system to be adaptable to a range of other wing designs.

Successful execution of such behaviors requires the robots to coordinate in passing information and being at the right places at the right times. For efficiency, the robots move concurrently, when feasible, pausing until each has achieved its goal. This coordination is facilitated by a layered architecture, named Syndicate [19], that we developed for a previous project. Syndicate has capabilities to define data-driven behaviors and hierarchically defined tasks that can be distributed amongst multiple robots. The tasks decompose into subtasks, bottoming out in tasks that activate and deactivate goal-achieving behaviors (such as moving and rotating the spar catcher until its limit switches indicate that the end effector is aligned with the spar cap). Temporal constraints between subtasks are represented using the Task Description Language [20], an extension of C++ that includes syntax for task decomposition, temporal coordination, execution monitoring, and error recovery. Exchange of data is performed using IPC [21], an efficient publish-subscribe messaging system that is cross-platform and cross-language (while most of the system is written in C++, the vision processing is in Python). Together, these tools enable us to specify complex coordination strategies succinctly, and enable them to be carried out efficiently, with only a small latency in communications.

In addition, Syndicate enables us to flexibly define behaviors that can take advantage of robot resources and availability. In the current system, for instance, the *guarded move* behavior consists of one robot that moves and another robot that senses some condition that acts as the guard. The guards that we use are either accelerometers or limit switches. The type of sensor being used and the robot on which that sensor is mounted can be indicated at run time, which provides for a large degree of flexibility. While we currently do not change the mappings at run time, we do use the same guarded move behavior in several different ways, mixing and matching robots and sensors to get the most effective behavior for achieving a particular subtask.

D. Dynamic Scheduling: Coordination at the Macro-Level

At the ladder assembly level, we take a dynamic scheduling approach to allocating transport, assembly, fixture and assist tasks to robots, and coordinating their execution in space and time. Our approach combines several basic concepts to reduce the overall uncertainty introduced by our mobile assembly approach and retain manufacturing efficiency. First, we assume a structured factory floor, where parts are transported from storage areas to assembly sites along well-defined corridors. This assumption allows us to manage interactions between robots exclusively at intersections and assembly sites, and promotes efficient overall traffic flow. The corridor network also provides a structure for decomposing part delivery and insertion tasks into sequences of localized component behaviors (e.g., corridor segment travel with following and queuing behaviors). To retain flexibility in exceptional situations (e.g., a failed robot), we advocate integration with contemporary path planning techniques [23], [2] to temporarily reconfigure the corridor network.

Second, we assume that basic traffic control protocols are used to govern robot movements through intersections. Using knowledge of this protocol together with a planning model that accounts for both temporal and spatial constraints on part movement, it becomes possible to reliably predict part delivery durations. This capability in turn, enables a hierarchical planning approach, similar to that proposed in [8], where robot plans are generated according to an abstract model, and expanded closer to execution time when uncertainty is reduced. In contrast, prior work in task allocation for mobile manufacturing has either assumed simpler, homogenous (e.g., tress) assembly structures [30], [29], where assembly mass computations can effectively drive allocation decisions and prediction of part delivery durations is not important, or mobile robot manufacturing applications (e.g., cooperative painting of large structures) where robot movement constraints can be reasonably ignored [4].

Third, part delivery and insertion plans are generated dynamically as a function of the current execution state. Exploiting an abstract assembly planning graph, a forward *look-ahead* projection is repeatedly used to determine 1) the earliest time at which a given assembly station will go idle (i.e., waiting for the next part to arrive) and 2) the lead time required to deliver the next part, and this look-ahead information provides a basis for scheduling "just-in-time" part deliveries. The forward projection utilizes a state-dependent shortest-path calculation to determine the most efficient path for a given part to take at a given time, and a hedging parameter is utilized to account for remaining durational uncertainty. Timing the start of part delivery tasks for arrival just when they are needed is advantageous in two respects: planning decisions are made with the most current (and hence most certain) information possible about the global execution state, and traffic congestion on the floor is minimized as a side-effect, which minimizes the uncertainty associated with look-ahead prediction. As new tasks are allocated to robot teams and scheduled, constituent subtask sequences are communicated to Syndicate, and mapped to corresponding behaviors for execution. As execution proceeds, results are communicated back to the dynamic scheduler to update its model of the current execution state.

IV. Experiments

We ran a series of trials to help quantify the performance of our rib-insertion system (minus the visual-servoing alignment behavior, which had not been completely integrated at the time of the experiment). Each trial started about two meters from the entrance to the spars at different left-to-right distances from the center. The result of each trial was classified qualitatively, in terms of overall success/failure (i.e., did the robots have to be stopped manually before completing the final "close the door" behavior) and quantitatively, in terms of the final position of the rib with respect to the L-bracket.

25 trials were performed. A distribution of starting positions was chosen for the trials, ranging plus or minus 60 cm from the center of the two spars and plus or minus 70 degrees from facing towards the spars. 22 trials (88%) percent of the trials were successful, as defined above. Two of the failures were caused by the brake on the spar carrier not holding sufficiently, and the rib pushed the target plate back and was emplaced on the far side of the L-bracket. In the other failure, the robot stalled during the "close the door" behavior.

Each successful insertion took on average 88 seconds (maximum of 108, standard deviation of 9.4). For the successful trials, the average distance from the desired position of the rib was 0.68cm in the vertical direction (maximum of 1.0cm, standard deviation of 0.32cm) and 0.77cm in the horizontal direction (maximum of 1.0cm, standard deviation of 0.30cm). The average gap between the top hole of the rib and the right L-bracket was 0.86cm (maximum of 2cm, deviation of 0.79cm) and that of the top rib hole and the left L-bracket was 0.53cm (maximum 1.2cm, deviation of 0.40cm). In all successful cases, the bottom of the rib was firmly pressed against both brackets. Note that all these figures are significantly less than the precision of global localization system. This demonstrates the utility of using temporary fixtures and compliance in achieving precision assembly. We can still further improve these results. The current visual servoing implementation requires an initial error of .8cm, which the "close the door" behavior provides in more than half of our trials. In the near term, we will either improve the visual serving to tolerate a 1cm initial error, improve the "close the door" behavior to result in a sub-centimeter accuracy, or both.

V. A Proposed General Assembly Framework

The overall assembly sequence and each of the component techniques described above provide an effective solution to the example assembly task, however the sequencing and the implementation of the components relied on the experience and intuition of designers. It would be desirable to provide a more general framework that can be used to inform the creation of similar assembly sequences. In this section, we propose such a framework based on the idea of "funneling". Typically, a manipulation task requires a sequence of actions through which an object is oriented and re-oriented until it arrives at a desired configuration. This strategy has been investigated in the context of funnels: given a part and a set of pose constraints, one can prescribe a controller to orient and position a simple part through a sequence of actions [12], [3]. An individual funnel can be viewed as a policy that has a domain of initial configurations, a goal set of final configurations, and a controller that directs motion from the domain to the goal set. This idea was generalized to a concept called sequential composition [1] where feedback controllers are defined over local domains such that their action brings the system to a goal set within the domain. One policy, denoted by Φ, is said to prepare another if the goal set of the first policy is contained (either wholly or partially) within the domain of the second policy. A graph, called a *prepares graph,* is formed where a node corresponds to a controller's domain and a directed edge from policy one to policy two indicates the controller of policy one converges on a goal set which is a subset of policy two. Once the prepares graph is formed, a planner can search the graph for a path between a node that contains the initial state and a node that contains the overall goal state.

Drawing inspiration from funneling and sequential composition, we believe that precision in our mobile framework may be recovered by a sequence of operations, as opposed to just one. Assume without loss of generality that the control and sensor error of the mobile manipulator can be lumped into one error term. Now, lets consider a series of examples of rib insertion where the robot's error increases. Starting with a very small mobile robot error, we can prescribe a controller for the robot to precisely line up with holes of the rib and the bracket. The problem is, such a low-error mobile manipulator is either impossible to build or too expensive to purchase. Now, lets increase the error where the mobile robot is able to drive the rib to the bracket, but lacks the precision to line up the holes. Instead, the rib holder's four-bar mechanism lines up the holes using visual feedback, i.e., the holes are lined up using visual servoing. Once more, lets increase the error of the mobile robot, so that the rib initially makes contact with the bracket, moves the rib along the bracket until the holes are within some proximity of each other, and then once again relies of visual servoing to finish the task.

Often, the features of the assembly or those in the environment will not be of sufficient size to form a funnel. Lets return to our example where we increase the lumped error of the mobile manipulator. Now, the error is larger than the characteristic length of the bracket, which means the robot cannot reliably drive until it hits the bracket and then drive along the bracket until the holes come into proximity of each other. This is where the spar catchers come in. Since the feature on the spar is too small for the robot to track, we introduce temporary features. As can be seen in Figures 3 and 6, the temporarily deployed feature engages with the spar, and enlarges the bracket feature size so that the error of the mobile robot is smaller than the characteristic size of the feature. In a sense, we formed mechanical funnels with passively compliant features mounted on a mobile robot. We also form "virtual" funnels using vision and range sensors, also mounted on mobile robots. So, multiple robots will cooperate to form a single funnel. The use of multiple robots to create multiple funnels requires coordination among robots, both at the micro-level to achieve the individuals tasks and at the macro level to optimize the allocation of the tasks to the robots and the robots' movements.

VI. Conclusion and Future Work

This paper describes the development of robotic technologies that enable a new vision of collaborative mobile manufacturing by teams of robots. Of particular interest is the assembly of large structures (e.g., aircraft, ships, construction equipment) where in current practice, customized facilities incorporating extensive infrastructure in the form of fixed tooling, permanent fixtures, and special systems are built to enable efficient, high-precision manufacturing of a specific product. To increase capacity or to introduce new products in these industries, tremendously expensive facilities must be built, which can take years.

Our approach is to replace conventional robot systems and large infrastructure with mobile manipulators to carry out the assembly operation. Conventional robotic manipulators have high precision because they are rigidly attached to the ground and are made of heavy fixed objects that tend to be invariant to dynamic parameters such as payload and speed. Mobile manipulators do not enjoy the benefits of high mass and rigid fixturing to ensure precision, and hence we need to create mechanisms and control strategies for the mobile manipulators so they can precisely mate parts for assembly.

Thus far, this project has been driven by experimentation, engineering intuition, and the integration of other robotics fields such as localization and software architectures for robots. Through this experience, we have identified techniques that, when combined, mitigate the uncertainty in precision and efficiency that is introduced by replacing the permanent fixtures with mobile robots. While the efficacy of these techniques has been born out only in the wing-ladder domain, ultimately, our goal is to create a design framework where we can specify an assembly and then produce a planner that directs the motions of all of the robots and their manipulators to carry out the assembly process. This design framework will be mediated by funnels, which we seek to express in a sequential composition framework. Sequential composition allows us to prescribe a complex manipulation task, which requires precision, with a sequence of intermediate motions, each reducing uncertainty in component directions.

In the longer term, we seek to extend this framework to allow people and machine to cooperate and play to their strengths in a manufacturing setting. When doing this, one issue to consider is safety. Today, to maintain safety for human operators, the overall environment is incredibly well-structured. This includes human only and robot only pathways to avoid unnecessary interferences and to simplify motion planning algorithms. However, when humans must share the workspace there needs to be clear warning lights to indicate robot intentions but these necessary interactions also need to have pinch points and other hazards designed out.

References

[1] Robert R Burridge, Alfred A Rizzi, and Daniel E Koditschek. Sequential composition of dynamically dexterous robot behaviors. *The International Journal of Robotics Research*, 18(6):534–555, 1999.

[2] Marcello Cirillo, Federico Pecora, Henrik Andreasson, Tansel Uras, and Sven Koenig. Integrated motion planning and coordination for industrial vehicles. In *Proceedings 24th International Conference on Automated Planning and Scheduling*, 2014.

[3] Michael A Erdmann and Matthew T Mason. An exploration of sensorless manipulation. *Robotics and Automation, IEEE Journal of*, 4(4):369–379, 1988.

[4] Matthew Gombolay, Ronald Wilcox, and Julie A Shah. Fast scheduling of multi-robot teams with temporospatial constraints. In *Robotics: Science and Systems*, 2013.

[5] Satyandra K Gupta, Christiaan JJ Paredis, Rajarishi Sinha, and Peter F Brown. Intelligent assembly modeling and simulation. *Assembly Automation*, 21(3):215–235, 2001.

[6] Hirohisa Hirukawa, Yves Papegay, and Toshihiro Matsui. A motion planning algorithm for convex polyhedra in contact under translation and rotation. In *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, pages 3020–3027. IEEE, 1994.

[7] Xuerong Ji and Jing Xiao. Planning motions compliant to complex contact states. *The International Journal of Robotics Research*, 20(6):446–465, 2001.

[8] Leslie Pack Kaelbling and Tomás Lozano-Pérez. Hierarchical task and motion planning in the now. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1470–1477. IEEE, 2011.

[9] O Khatib, Kazu Yokoi, K Chang, Diego Ruspini, Robert Holmberg, and Arancha Casal. Coordination and decentralized cooperation of multiple mobile manipulators. *Journal of Robotic Systems*, 13(11):755–764, 1996.

[10] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g2o: A general framework for graph optimization. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, May 2011.

[11] Christian Laugier. Planning fine motion strategies by reasoning in the contact space. In *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*, pages 653–659. IEEE, 1989.

[12] Tomas Lozano-Perez, Matthew T Mason, and Russell H Taylor. Automatic synthesis of fine-motion strategies for robots. *The International Journal of Robotics Research*, 3(1):3–24, 1984.

[13] Matthew T Mason. Compliance and force control for computer controlled manipulators. *Systems, Man and Cybernetics, IEEE Transactions on*, 11(6):418–432, 1981.

[14] Brenan J McCarragher and Haruhiko Asada. A discrete event approach to the control of robotic assembly tasks. In *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*, pages 331–336. IEEE, 1993.

[15] W. Meeussen, E. Staffetti, H. Bruyninckx, J. Xiao, and J. De Schutter. Integration of planning and execution in force controlled compliant motion. *J. of Robotics and Autonomous Sys*, 56(5):437–450, 2008.

[16] E. Olson. Apriltag: A robust and flexible visual fiducial system. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3400–3407, May 2011.

[17] Anna Petrovskaya and Oussama Khatib. Global localization of objects via touch. *Robotics, IEEE Transactions on*, 27(3):569–585, 2011.

[18] G. Pratt and M. Williamson. Series elastic actuators. In *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 399–406. IEEE, 1995.

[19] Brennan Sellner, Frederik W Heger, Laura M Hiatt, Reid Simmons, and Sanjiv Singh. Coordinated multiagent teams and sliding autonomy for large-scale assembly. *Proceedings of the IEEE*, 94(7):1425–1444, 2006.

[20] Reid Simmons and David Apfelbaum. A task description language for robot control. In *Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ International Conference on*, volume 3, pages 1931–1937. IEEE, 1998.

[21] Reid Simmons and Gregory Whelan. Visualization tools for validating software of autonomous spacecraft. In *Proceedings of International. Symposium on Artificial Intelligence, Robotics and Automation in Space*. Citeseer, 1997.

[22] Siddhartha S Srinivasa, Dmitry Berenson, Maya Cakmak, Alvaro Collet, Mehmet Remzi Dogar, Anca D Dragan, Ross A Knepper, Tim Niemueller, Kyle Strabala, Mike Vande Weghe, et al. Herb 2.0: Lessons learned from developing a mobile manipulator for the home. *Proceedings of the IEEE*, 100(8):2410–2428, 2012.

[23] Glenn Wagner and Howie Choset. Subdimensional Expansion for Multirobot Path Planning. *Artificial Intelligence*, to appear:1–46, 2014.

[24] D. E. Whitney. Quasi-static assembly of compliantly supported rigid parts. *ASME J. of Dynamic Sys., Measure., and Control*, 104:65–77, 1985.

[25] Randall H Wilson. Geometric reasoning about assembly tools. *Artificial Intelligence*, 98(1):237–279, 1998.

[26] Randall H Wilson and Jean-Claude Latombe. Geometric reasoning about mechanical assembly. *Artificial Intelligence*, 71(2):371–396, 1994.

[27] Tony C Woo and Debasish Dutta. Automatic disassembly and total ordering in three dimensions. *Journal of Manufacturing Science and Engineering*, 113(2):207–213, 1991.

[28] Jing Xiao and Xuerong Ji. Automatic generation of high-level contact state space. *The International Journal of Robotics Research*, 20(7):584–606, 2001.

[29] Seung-kook Yun and Daniela Rus. Adaptation to robot failures and shape change in decentralized construction. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 2451–2458. IEEE, 2010.

[30] Seung-kook Yun, Mac Schwager, and Daniela Rus. Coordinating construction of truss structures using distributed equal-mass partitioning. In *Robotics Research*, pages 607–623. Springer, 2011.