Visually Guided Coordination for Distributed Precision Assembly

Michael L. Chen, Shinji Kume, Alfred A. Rizzi, and Ralph L. Hollis
The Robotics Institute, Carnegie Mellon University
{mc, shkume, arizzi, rhollis}@ri.cmu.edu

Abstract

We document our initial efforts to instantiate visually-guided cooperative behaviors between robotic agents in the minifactory *environment. Minifactory incorporates high-precision 2-DOF robotic agents to perform micron-level precision 4-DOF assembly tasks. Here we utilize two minifactory agents to perform visual servoing. We present a detailed description of the control and communication systems used to coordinate the agents. To provide a suitable communications infrastructure, we describe the development of a new inter-agent communication system which uses low-latency protocols carried by a commercial 100 Mb Ethernet network. Finally, we conclude by presenting experimental results from our first coordinated multi-agent task, the visually guided positioning of a small medical device.*

1 Introduction

The need to develop self-calibrating easily reconfigured automation systems coupled with the relentless improvements in performance and reductions in size of computational and communications hardware has made the development of modular distributed automation systems both attractive and practical. Unfortunately the resulting distributed systems can, if not thoughtfully designed, be difficult to program and control. To avoid this pitfall it is important that distributed systems incorporate suitable sensing capabilities, control strategies, and communications capabilities to support the effective coordination of disparate entities.

Within the Microdynamic Systems Laboratory[1] we are developing an example of such a system called *minifactory* [1]. Minifactory, which fits within the larger conceptual framework of the Agile Assembly Architecture (AAA), relies on collections of 2-DOF robotic agents to form rapidly reconfigurable table-top sized automation systems suitable for the precision assembly of complex electro-mechanical products [2]. Two classes of robotic agents dominate the minifactory environment. *Courier* robots operate in the plane of the factory floor to transport sub-assemblies and precisely position them during assembly operations, and overhead *manipulator* robots perform high-precision pick and place operations on the sub-assemblies. By operating in close coordination, pairs of these robotic agents can perform cooperative 4-DOF assembly tasks traditionally performed by SCARA robots, while providing significant advantages in terms of compactness, precision, and flexibility [3].

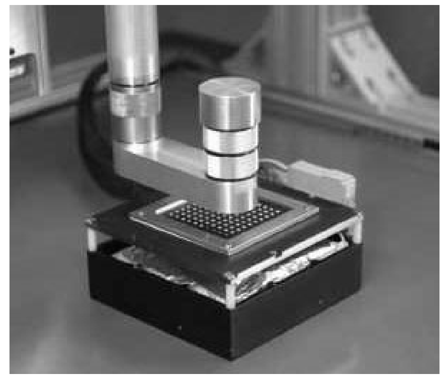

Figure 1: Photograph of the *courier* (lower box shaped robot) presenting a part to the *manipulator* (upper robot).

To realize these advantages, the individual robotic agents involved in a cooperative assembly operation must provide and share suitable capabilities and strategies. For minifactory agents, the sensing capabilities are organic to the individual agents, the control strategies (described in Section 2) are designed to interconnect in a modular fashion, and the communication capabilities support distributed control strategies through the use of a semi-custom communications infrastructure (termed AAA-Net, and described in Section 3). This paper presents our initial efforts to instantiate one such cooperative behavior – visually guided coordinated motion. In the example task, a courier and manipulator coordinate their activity and share sensing resources to ensure that parts are held at a fixed relative locations despite external disturbances. To accomplish this task, the manipulator, equipped with a monocular monochrome field-rate vision system, visually acquires and tracks the part carried by

[1]See `http://www.cs.cmu.edu/~msl`.

the courier. Motion of the courier is then controlled to keep the part centered in the vision system's field of view, despite arbitrary motions by the manipulator. We have thus created a single robotic visual servoing system that spans the two agents. Figure 1 shows a view of the the actual system, with the courier robot presenting a part to the manipulator's vision system.

Section 2 of this paper describes the capabilities of the robotic agents, the control strategies used to coordinate their behavior, image processing techniques, and the organization of the communication between the various computational tasks used to implement the system. Section 3 describes in detail the physical communications infrastructure and the protocols used to provide the low-latency, high-performance communication channels needed by the algorithms of Section 2. Finally, Section 4 details our initial experiments with the entire system.

2 Coordination Strategy

For our initial investigation of visually guided coordination in the minifactory environment, we have chosen to undertake a simplified position regulation task. The courier agent will carry a sub-assembly of a small medical device measuring 3 mm on a side, and the manipulator agent will observe this device through its vision system. The task will be to keep the sub-assembly centered within the field of view of the camera while the manipulator undergoes a rotation. The only information transacted between the two agents will be commands from the manipulator indicating the velocity at which the courier should move, based on the manipulator's visual observations.

2.1 Robotic Agents

The manipulator agent is capable of vertical (z) and rotational (θ) movement. The z range of motion is approximately 125 mm and achieves a resolution of 2 μm, while the range of motion in θ is approximately $\pm 270°$ with a resolution of $0.0005°$. The manipulator also incorporates a field-rate frame grabber for image acquisition and its end effector contains a monochrome camera and a vacuum suction pickup device, providing an eye-in-hand robotic configuration.

The courier agent provides motion in the x, y plane – both translation and a limited range of rotation. It incorporates a magnetic position sensor device, providing 0.2 μm resolution (1σ) position measurements [4] and closed loop position control [5].

2.2 Agent Communication

Figure 2 shows the structure of the communication channels between the two agents and their computational processes. In this diagram, circles represent processes, and the two separate shaded areas represent the courier and manipulator agents. Internal communication between processes on the individual agents utilize shared memory structures, providing fast data sharing. For the visually guided coordination task the manipulator uses three processes, in addition to the AAA-Net server. The first (Executor) executes low-level control, the second (Head) interprets user-level commands and scripts, and the third (Vision) analyzes images from the frame grabber. The courier has a similar structure but lacks the vision system and associated process.

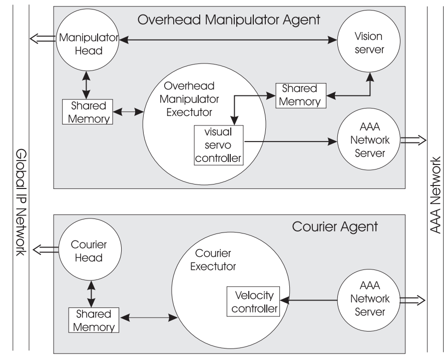

Figure 2: Communications within and between manipulator and courier agents.

2.3 Image Processing

To accomplish the visually guided coordination described above, the manipulator, which manages the camera and frame grabber, must perform all the image processing tasks. The camera system contained in the end effector has a monochrome CCD camera and an adjustable two lens optic. The resulting magnification of this system was set at 0.75 so that one pixel corresponded to approximately 10 μm of motion. Tsai's coplanar method [6] was used to calibrate the intrinsic parameters of the camera system, as well as the extrinsic parameters. Wilson's implementation [7] was used to numerically optimize the parameter values from measurements of a calibration fixture.

Performing visual servoing requires both global and local feature localization strategies. The global scheme finds the top corners of the sub-assembly without any initial position estimates, while the local scheme tracks the sub-assembly at field rate given recent position estimates. The global search begins by locating the centroid of the sub-assembly's image, providing an initial estimate for the location of the part. Performing a Hough transform eliminates pixel noise and allows the use of linear regression techniques to recover the location of lines representing the part edges. As seen in figure 3, the intersection of the lines provide accurate

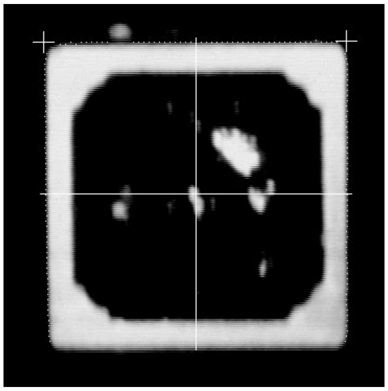

Figure 3: Results of global search for the top corners of the sub-assembly – the recovered center and corner locations are marked.

positions of the top two corners of the device.

Given these estimates of corner locations, local image processing routines can track the corners at field-rate (60 Hz). We utilize the XVision package [8] to perform these tracking tasks. The flexible nature of XVision allows programming constraints between the two corner trackers we used. This improves robustness of the system by allowing it to recover from transient occlusion events. For example, if one corner tracker fails, information from the other corner's position can be used to reinitialize the failed tracker. Once the occlusion passes, the failed tracker can relocate the appropriate corner.

2.4 Visually Guided Control

Visually guided control was accomplished by processing image information in the manipulator agent and sending velocity commands to the courier agent and its controller. The vision server, as shown in Figure 2, tracks the position of the sub-assembly at field rate, and in turn passes the image-plane position information to the visual servo controller running as part of the executor process. The visual servo controller computes a desired velocity for the courier to keep the sub-assembly at the desired position in the image plane. These commands are finally sent to the courier controller at field-rate via the AAA-Net.

A simplified model of the visual servoing control system is shown in Figure 4. This depiction emphasizes the three main components of the system: the visual servo controller, the courier controller, and the courier motor. The visual servo controller is implemented on the manipulator agent and can be classified as an image-based visual servo controller since control values are directly computed from image features [9]. In the block diagram, **H** encapsulates the camera and the image processing routines used to locate and

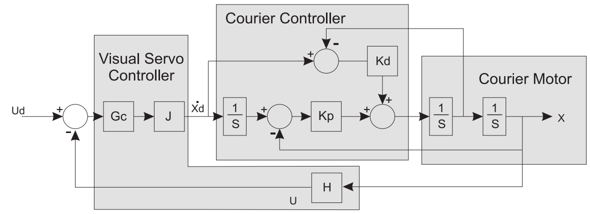

Figure 4: Simplified model of visual servoing control system.

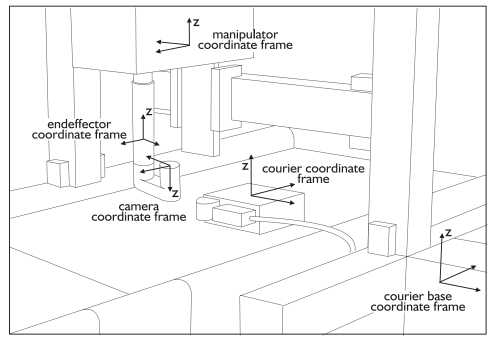

Figure 5: Minifactory coordinate frames.

track the sub-assembly within an image. The block labeled $\mathbf{G_c}$ represents the adjustable gains of the visual servoing controller – for improved performance this includes a proportional term, as well as an integral term, which both operate on the error between the desired and current image plane position of the sub-assembly. The block labeled **J** represents the image Jacobian and depends on the rigid transform between the camera coordinate frame and the courier coordinate frame. Figure 5 shows our convention for frame placement in minifactory. Note that the frames attached to the courier and end effector depend on the agents' current positions and are constantly recalculated, while the remaining frames are precisely located during the factory self-calibration process.

The remaining major components of the simplified visual servo system model represent the courier and its controller. The dynamics of the courier motor are approximated by a double integrator, and the courier controller is replaced by a simplified proportional-derivative control scheme which has been modified to accept velocity commands from the manipulator agent. The details of courier behavior are beyond the scope of this paper and can be found in [5].

3 Network Infrastructure

To support seamless cooperation between physically distinct agents in the minifactory, each agent is equipped with two network interface devices. These

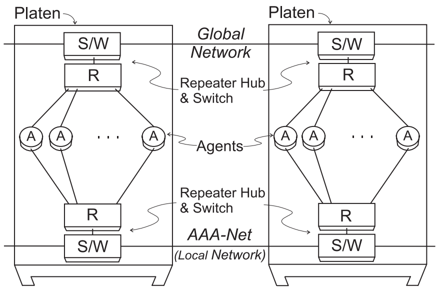

Figure 6: Physical network structure.

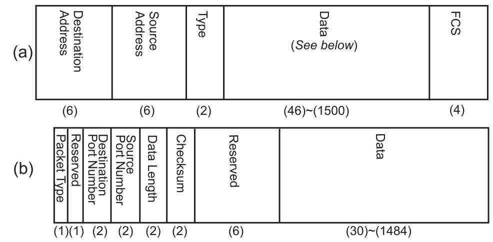

Figure 7: Ethernet frame format used by AAA-Net. (a) Standard IEEE-802.3 frame format. (b) AAA-Net data format.

provide a scalable communications infrastructure throughout the minifactory system. The first interface connects to a network which carries non-latency-critical information, such as user commands or information destined for the factory interface tool. This network utilizes standard IP protocols [10]. The second interface connects to a network (AAA-Net) which carries real-time information critical to the timely coordination of activity between agents. This includes the master-slave coordination described in Section 2 of this paper as well as other activities performed by multiple agents transiently acting as a single *machine*.

3.1 Physical Network

Unlike the majority of industrial field networks [11, 12, 13], AAA-Net makes use of commercial off-the-shelf 100 Mb Ethernet hardware. This provides access to a wide variety of low-cost and compact hardware options. The result is *i*) reduced investment in network infrastructure hardware, *ii*) simplified installation and maintenance, and *iii*) direct access to rapidly improving Ethernet technology. Unfortunately, as a result of the CASM/CD media access rule used by IEEE 802.3 Ethernets, the timing of packet delivery is not deterministic. Thus transmission latency can not be absolutely bounded, and packet delivery can not even be guaranteed. However, our specification for agent coordination only requires the delivery of 100 byte data packets at a maximum rate of 1 kHz between agents that are physically collocated. Given the maximum agent density, we can conservatively bound the local bandwidth needs of the entire minifactory system at roughly 10 Mbps. By choosing to use 100 Mbps Fast Ethernet technology [14], we are able to provide an infrastructure that will operate well below its total capacity and thus minimize the risk of packet collisions and the associated delays and losses in communication.

Figure 6 depicts the communications infrastructure of the minifactory system. As can be seen, both the AAA-Net and the global IP network are configured as a chain of star topology local-networks, with a Fast Ethernet repeater hub at the center of each star and Fast Ethernet switches forming the connections between the local-network segments. The repeater hub allows each agent in a network segment to directly communicate with its immediate neighbors, while the frame relay switches allow for arbitrary daisy chaining of the network, overcoming the topology limitations that are fundamental to Fast Ethernet. The switches also serve to localize communications within the factory system by not transmitting data packets destined for local agents to the remainder of the factory and by selectively transmitting those packets destined for other network segments toward their destination yielding a scalable communications infrastructure[2].

3.2 Communications Protocol

To facilitate a wide variety of local interactions between agents, the AAA-Net protocol supports both a connection-based guaranteed data transmission scheme as well as a connection-less non-guaranteed data transmission scheme. These services are not unlike the familiar TCP and UDP services commonly used on IP networks, although they are significantly simplified to improve performance in the highly structured network environment of AAA. Figure 7 depicts the simplified header used by the AAA-Net protocol. For non-guaranteed communication, the TYPE field in Figure 7(b) is set appropriatly, and the body of the message is placed in the DATA field. The packet is then immediately placed on the Ethernet network with no effort made to guarantee its delivery. This form of communication is appropriate for the exchange of rapidly changing values, such as the velocity commands sent from the manipulator to the courier described in Section 2, where the loss or delay of a single packet is insignificant since additional data will follow shortly.

This same packet format is also used to implement a connection-based guaranteed delivery protocol. The packet TYPE is again set, and the fields marked as "reserved" are used by the protocol to ensure reliable

[2]Note that in AAA/minifactory the bulk of high-bandwidth communication will be between agents that are attached to the same network segment, while the remainder will involve agents that are typically attached to neighboring segments [2].

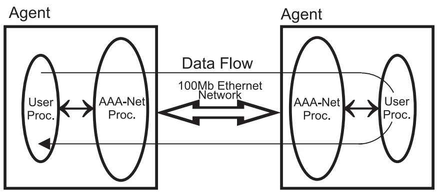

Figure 8: Test set-up for AAA-Net latency measurements.

Number of Agents	Round-trip (μsec)			
	100 byte message		1000 byte message	
	mean	std	mean	std
2	566	20	804	22
4	562	19	816	34
6	567	21	826	38

Table 1: Measured AAA-Net round-trip times (mean and standard deviation) for varying network loads.

delivery of the data stream. This protocol is appropriate for the exchange of larger more complex data structures or messages that will only be sent once, and whose receipt must be guaranteed.

3.3 Network Performance

In addition to the inter-agent cooperative behaviors demonstrated in the remainder of this paper, we have measured the performance of the non-guaranteed AAA-Net protocol and underlying network hardware. A collection of agents connected to the same network segment exchanged messages as shown in Figure 8 at 1kHz. Data was sent from the user process on one agent, received by another agent, and retransmitted back to the original agent, and the round-trip time measured. To evaluate the impact of network contention this test was performed with one, two, and three pairs of agents communicating simultaneously. Messages of both 100 and 1000 bytes in length were transmitted, and the results are shown in Table 1. From our past experience, we estimate that between 45 and 50% of the average round-trip time is spent passing through the AAA-Net daemon process the four times a message requires to make the complete circuit.

4 Results

The visually guided coordination experiment was composed of four major steps: automatic calibration of the factory, presentation of a sub-assembly to the manipulator, initialization of the visual servoing system, and movement of the manipulator's θ axis as a disturbance input. Calibrating the factory provided precise coordinate transforms between various parts of the minifactory as shown in Figure 5. These transforms were

Gains G_c		u error (mm)		v error (mm)	
K_p	K_i	mean	std	mean	std
7.0	0.0	-0.286	0.012	0.007	0.013
7.0	0.005	0.000	0.013	0.001	0.014
12.0	0.0	-0.167	0.017	0.005	0.026

Table 2: Steady state image plane position error for the given visual servo controller proportional and integral gains.

needed by the visual control system as described in section 2.4.

After calibration, the courier moved to present a sub-assembly to the manipulator. An image was acquired and searched to initialize a pair of corner trackers. To guarantee that the trackers had time to settle, both agents were prevented from moving for 1 second. Figure 9 shows that the desired and measured positions of the sub-assembly, as measured by the vision system during a typical experiment, were different at this point in the experiment. While keeping the manipulator fixed, the courier was allowed to move, and as can be seen, the initial error in the position of the sub-assembly was quickly accommodated.

After the vision system was initialized, the manipulator's θ axis was rotated clockwise at a constant rate of 0.052 rad/s. This served as a disturbance to the visual servoing system starting at 2 seconds. Table 2 shows mean and standard deviation of the visual error signals (both u and v directions) for three different settings of the proportional and integral gains. Note that with the integral term disabled ($K_i = 0$), the resulting u axis error was much greater than the v axis error (note that 1 pixel corresponds to roughly 0.01 mm). The difference in error magnitude between the axes is a direct result of the camera configuration, since θ motion of the manipulator maps directly into a u motion on the image plane. Setting $K_i = 0.005$ significantly reduced the error during motion, as shown in Figure 9 and Table 2, at the cost of slightly slower transient performance.

The overall results shown in Figure 9 were encouraging. The manipulator's θ axis rotation caused the courier's tangential velocity to be approximately 5 mm/s, while image plane measurements showed a standard deviation of less than 15 μm in both axes of the vision system. However, the peak-to-peak error was approximately 0.04 mm in the u axis and 0.05 mm in the v axis, corresponding to image movements of nearly 5 pixels. The courier angle plot in Figure 9 provides one possible cause for this problem. As can be seen, the recorded peak-to-peak motion was approximately 0.002 rad, even though the courier was commanded to hold its orientation throughout this experiment. This "wobbling" is believed to result from miscalibration of the courier position sensor [4] and contributed to most of the error.

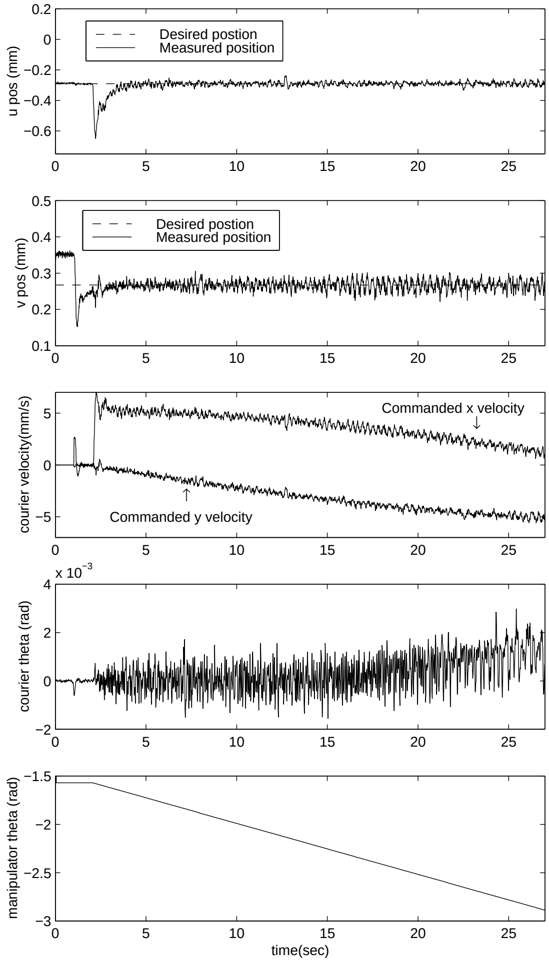

Figure 9: Visual servoing results for $K_p = 7.0$ and $K_i = 0.005$, including measured image plane location of the sub-assembly (u and v), commanded courier velocities ($\dot{x}$ and $\dot{y}$), and measured angular position of the courier and manipulator.

5 Conclusion

We have documented the system level structures used to enable coordination of distributed assembly operations in our modular precision manufacturing environment (minifactory) and presented initial results for a visually guided task. The performance measured in this initial experimental study has been convincing, and we are currently conducting additional experiments to more completely characterize the system. Furthermore, we are beginning to explore additional coordinated behaviors, including compliant precision insertion and force control tasks. These behaviors, in combination with extensions of the visually guided behavior, will enable the minifactory to efficiently undertake a wide variety of assembly tasks and will demonstrate the viability and flexibility of distributed precision assembly systems.

Acknowledgments

This work was supported in part by NSF grants DMI-9523156, and CDA-9503992. The authors would like to thank Arthur Quaid, Zack Butler, Ben Brown, Jay Gowdy, and Patrick Muir for their invaluable work on the project and support for this paper.

References

[1] R. L. Hollis and A. Quaid. An architecture for agile assembly. In *Proc. Am. Soc. of Precision Engineering, 10th Annual Mtg.*, Austin, TX, October 15-19 1995.

[2] A. A. Rizzi, J. Gowdy, and R. L. Hollis. Agile assembly architecture: An agent-based approach to modular precision assembly systems. In *IEEE Int'l. Conf. on Robotics and Automation*, pages Vol. 2, p. 1511–1516, Albuquerque, April 1997.

[3] A. Quaid and R. L. Hollis. Cooperative 2-DOF robots for precision assembly. In *Proc. IEEE Int'l Conf. on Robotics and Automation*, Minneapolis, May 1996.

[4] Zack J. Butler, Alfred A. Rizzi, and Ralph L. Hollis. Precision integrated 3-DOF position sensor for planar linear motors. In *IEEE Int'l. Conf. on Robotics and Automation*, pages Vol. 4, p. 3109–14, 1998.

[5] Arthur E. Quaid and Ralph L. Hollis. 3-DOF closed-loop control for planar linear motors. In *Proc. IEEE Int'l Conf. on Robotics and Automation*, pages 2488–2493, May 1998.

[6] R. Y. Tsai. An efficient and accurate camera calibration technique for 3d machine vision. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 364–74, 1986.

[7] R. Wilson. Implementation of tsai's camera calibration method. http://www.cs.cmu.edu/~rgw/Tsai-method-v3.0b3.tar.Z.

[8] G. D. Hager and K. Toyama. X Vision: a portable substrate for real-time vision applications. *Computer Vision and Image Understanding*, 69(1):23–37, Jan. 1998.

[9] S. Hutchinson, G. D. Hager, and P. I. Corke. A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, 12(5):651–70, Oct. 1996.

[10] J. Postel. Internet Protocol - DARPA Internet Program Protocol Specification – RFC 791. Technical report, USC/Information Sciences Institute, Sep. 1981.

[11] Profibus Home Page. http://www.profibus.com.

[12] ControlNet Online. http://www.controlnet.org.

[13] WorldFIP Home Page. http://www.worldfip.org.

[14] IEEE 802.3u-1995 10 & 100 Mb/s Management, October 1995. Section 30, Supplement to IEEE Std 802.3.