A FINITE STATE MACHINE APPROACH TO VISUAL SERVOING TO INCREASE POSITIONAL ACCURACY OF IMPEDANCE CONTROLLED ROBOTS

Kevin R. Ruybal, Ron Lumia, John E. Wood
Department of Mechanical Engineering, University of New Mexico, Albuquerque, NM, USA
lumia@unm.edu

ABSTRACT

A visual servoing controller is programmed using finite state machine logic to increase the positional accuracy of a robotic arm under impedance control during free-space motion, to better align a chamfered peg with a hole, and minimize the possibility of wedging during insertion. This approach allows many types of manipulators commanded through Cartesian trajectory inputs to compensate for positional inaccuracy, enabling improved alignment of the peg and hole knowing only their approximate position within the workspace. The result is a compliant robotic manipulator that is able to achieve peg and hole alignment with an acceptable average lateral error of 0.0025 meters and a rotational error of 0.022 radians, which in our case is sufficient to allow an impedance controlled robot to successfully avoid wedging during peg and hole assembly.

KEYWORDS

Finite State Machine, impedance control, robotics, accuracy, assembly.

1. INTRODUCTION

Automated robotic manufacturing is spreading to new industries. The driving force behind this growth is the advent of new, enabling technologies that allow robotic manipulators to perform tasks that were previously impractical, or even impossible. Assembly of components, such as fasteners, plugs, and sockets, are typical examples. While traditional, position controlled manufacturing robots are repeatable, they lack the compliance needed to assemble such components without damaging them. In the broad category of peg-and-hole assembly, one solution to this problem is to equip robotic manipulators with a compliant end effector, known as a Remote Center Compliance (RCC), which provides a predefined stiffness and center of compliance to the object being manipulated by the robot. This allows the system to absorb positional errors that result from controller error and variations within the manipulator's workspace. This error accommodation is imperative to avoid generating large contact forces between the end effector and external objects due to positional error [1,2].

In some cases, however, the RCC is not practical due to its mechanical limitations. Such instances include horizontal motions in which significant mechanical counter-weighting is needed, or on manipulators that perform multiple tasks that would each require a unique RCC. The alternative to the RCC is to incorporate compliant behavior into the control software of the manipulator. An example of a software-based solution to provide compliant behavior is Raibert and Craig's hybrid force and position controller. This controller allows the end effector of a robot to rigidly track position inputs in one direction, while tracking a trajectory of desired force relationships in other directions [3]. A similar control strategy defines a "configuration space" that applies limits on the applied forces and motions in some directions while allowing the robot to move freely in others [4,5]. Force feedback as developed by Whitney allows the robot to be commanded with measured end effector force instead of torque, position, or velocity [6]. While successful, the above-mentioned methods of imposing compliant behavior come at the cost of additional sensors that must be added to the robot.

Hogan developed a method of compliant control in 1985 that operates without force sensors. Called impedance control, this strategy actively enforces a desired stiffness and damping relationship between the robot and its environment through its joint torque calculations [7]. Aside from its advantage of using fewer sensors because the compliant behavior is calculated using only joint position feedback, Hogan's impedance controller is able to maintain a relationship between position, velocity, and the desired output force. This multiple input relationship property has many advantages when a robot must interact with external forces applied by its environment. Because the impedance controller maintains a stable and constant impedance relationship when a robot's end effector comes into contact with an object, it is capable of easily transitioning between the gross and fine motion portions of assembly [ibid].

Despite these advantages, however, impedance control suffers from positional inaccuracy. This problem is inevitable and is caused by two factors. First, impedance control is based on a kinematic model of the robot. Small errors in the estimation of link weights, joint friction, inertia, etc. all add up to positional error of the robot's end effector. Second, even with an accurate kinematic model, a great portion of positional error is directly related to the desired stiffness of the system. When a robot needs to be compliant to perform a delicate assembly task, its potential to exert control forces needed to achieve a desired Cartesian position is limited. This limits its ability to overcome gravity, friction, and inertial forces to rigidly adhere to a Cartesian trajectory. There have been some attempts to increase the positional accuracy of impedance control. Kelly et al. created an adaptive impedance controller to overcome problems that arise from unmodeled dynamics in a robot system using joint position and velocity sensors in addition to an end effector force sensor [8,9]. Other closed-loop hybrid impedance control systems that improve the robot's dynamics model can be found in [10,11]. While these types of model-improvement impedance control schemes increase robot control accuracy, which is a large benefit, they do little to help the robot interact with its environment or successfully complete a specific manufacturing task.

The system presented in this paper utilizes an external computer vision system as a means to provide feedback and compensation to overcome impedance control's positional accuracy limitations for a peg-in-hole insertion task. It combines the desired impedance control behavior of Hogan's implementation with the positional accuracy achieved through visual servoing. The result is a robot controller that is capable of providing improved positional accuracy for a compliant robot while helping to avoid assembly failure due to wedging.

2. THE GENERAL ASSEMBLY PROBLEM

The peg-and-hole assembly problem has been broadly studied due to its many applications in real world manufacturing. Whitney showed that the success of peg-and-hole assembly depends on how the parts interact with each other as they pass through distinct contact states during insertion [1]. Each contact state presents different internal forces, which can cause two specific types of assembly failure as shown in Figure 1. The first type of failure is wedging. Wedging occurs when contact forces between the peg and hole become compressive and store energy applied to the peg, holding it stationary. The second type of failure is called jamming. Jamming occurs when the resultant insertion force is too far off of the insertion axis (φ becomes too large) to push the peg into the bottom of the hole [12]. While jamming can occur from one or two point contact, wedging always a two-point contact phenomenon. Both types of failure are most likely to occur at the onset of 2-point contact between the peg and hole.

Jamming is avoided by allowing the peg and hole to rotate and correct for misalignment. This is accomplished through the compliant behavior of the manipulator, which allows the relationship

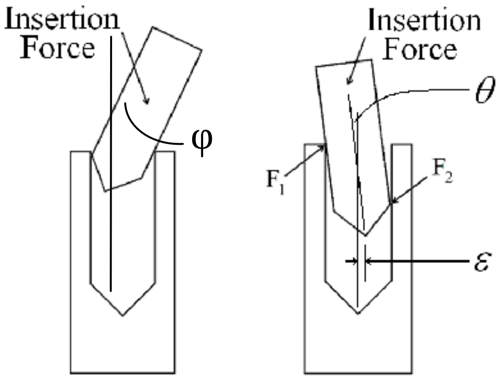

Figure 1. Two primary types of peg and hole assembly failure: jamming, left, and wedging, right. Φ is the angular misalignment that creates jamming. θ is the angular misalignment for wedging, with ε the resulting error in the peg's tip.

between the insertion force, lateral force, and reaction moment applied to the peg to naturally vary and change the direction of the single resultant force [1]. Using impedance control allows instances of jamming to be corrected so assembly can be completed successfully.

Unlike jamming, however, once the peg has become wedged in its hole, there is no way to correct the failure without completely removing two-point contact between the peg and the hole. Because having to repeat an operation is not acceptable in an automated assembly process, it is desirable to avoid wedging altogether. Wedging is primarily a function of initial lateral, ε, and rotational, θ, error between the peg and the hole as they make initial contact, part geometry, and friction. Whitney performed a detailed analysis of wedging in the case of a cylindrical peg and chamfered hole [1]. Using physical measurements, e.g., from the peg-and-hole assembly system, we are able to determine the conditions under which the peg is likely to become wedged in its hole. These conditions are commonly visualized with a two-dimensional plot of the possible combinations of initial lateral and rotational errors that will not result in wedging. This plot is shown in Figure 2.

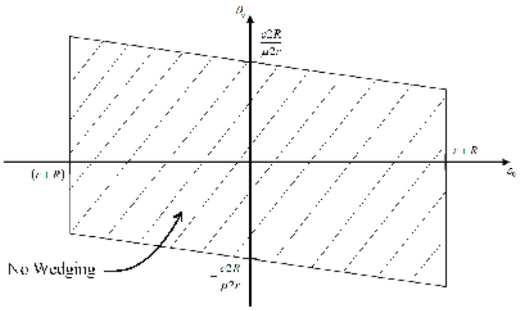

Figure 2. The wedging diagram sets boundaries for the amount of initial lateral and rotational error between the peg and hole that can be tolerated before wedging will result. These boundaries are a function of clearance ratio, c, inside radius of the hole, R, outside radius of the peg, r, coefficient of friction, μ.

The region enclosed by the parallelogram in Figure 2 indicates acceptable combinations of initial lateral and rotational error that will avoid wedging. The negatively sloped horizontal boundary lines are the result of a relation between lateral stiffness (Kx), rotational (Kθ) stiffness, and compliance center location (L). This analysis shows that less alignment error will result in greater protection from wedging during assembly. Thus, wedging avoidance through the ability of the system to accurately align the peg and hole during assembly becomes our design goal for the visual compensation system developed in this paper.

Whitney also produced a jamming diagram that visualizes the combinations of insertion force, contact forces, and moment applied to the peg during insertion that protect the peg and hole from jamming. While accurate alignment of the peg and hole reduce the possibility of jamming, measuring a direct correlation between alignment and the jamming diagram is not within the scope of this paper, as insertion is not being performed. We are instead only focusing on the alignment of the peg and hole prior to assembly. This is the only stage of the assembly process during which visual servoing is active. Further, the axes of the jamming diagram are measured in terms of forces and moments that do not exist prior to insertion, so we are unable to use the jamming diagram as a measure of success during the proposed experiments. Finally, the visual feedback system presented in this paper is used only to prevent wedging, as it is implied that compliant control of the robot performing the insertion is the primary means of preventing jamming through the accommodation of contact forces generated during assembly.

3. COMPUTER VISION FEEDBACK

In the previous section, it was explained that successful peg-and-hole assembly requires a system to avoid jamming through the use of compliant control, and wedging through accurate initial alignment of the parts. From this, we can conclude that the ideal system to perform this task is an impedance-controlled robot with high positional accuracy. Starting with a pair of existing impedance controlled robots, the goal of this paper is to use computer vision to provide adequate positional accuracy to the system.

Impedance control [6] operates much like a feed forward control loop because joint torque is computed and sent to the robot without positional error feedback. While this type of control is desirable for stability and controlling the interaction between the robot and its environment, there is no closed loop system to ensure that the end effector accurately follows the desired Cartesian trajectory. To boost the positional accuracy of the complete system, visual feedback has been incorporated into the control system to provide closed loop behavior to the impedance controller. The feedback strategy in this paper is to modify the Cartesian trajectory sent to the robots at each time step. This will allow the system to incrementally adjust the commands sent to the manipulator to compensate for errors between the desired and actual pose of the end effector. The vision system is required to recognize the positions of the tips of the peg and hole as well as their respective rotations with respect to the frame of the images collected by each image sensor. Then, a controller measures the 3D distance between the parts in pixels, and computes the rotation of the main axis of each part with respect to the horizontal axis of the image. An error signal is generated from these measurements and the amount of 6-DOF compensation needed by the chamfered peg to align itself with the hole in a shaft and remove the error is computed. The compensation is represented by a vector that is added to the 6-DOF trajectory input sent to the impedance controller.

In other words, the alignment controller will command an “incorrect” goal pose with a controlled amount of error to counteract the positional uncertainty of the impedance controlled robot. The impedance control equation enforces a desired impedance relationship between the robot and its environment through stiffness, K_P, and damping, K_D, coefficients. Using equation (1), the impedance controller computes the joint torques, T, required to achieve the desired joint positions, q_d, and joint velocities, $\dot{q}_d$, to move the end effector of the robot. This equation also compensates for gravity using the feed forward term, T_g, calculated from joint positions, q [13].

$$T = T_g(q) - J^T(q)[K_P(x - x_d) + K_D(\dot{x} - \dot{x}_d)] \quad (1)$$

The stiffness errors due to the change of Jacobian, (J), and kinematic modeling are the primary causes of positional uncertainty in an impedance controlled robot system. The positional uncertainty is usually most noticeable in the end effector, which often droops below its commanded goal due to the force of gravity. This problem is also apparent in lateral motions when compliant stiffness and damping parameters are chosen, as this weakens the joint torques computed for the robot to overcome friction and inertia. In this example, the alignment controller commands a compensated position that is higher than the desired position, $(x_d + compensation)$, allowing the robot to droop into better alignment. This simultaneously allows high positional accuracy while providing a wide range of acceptable impedance values to perform assembly tasks.

The goal of the closed loop system is to eliminate all of the positional error to exactly align the peg and the hole. This implies that the steady-state characteristics of the controller are more important than the transient behavior, assuming that the controller responds with reasonable speed. For this reason, two types of controllers were considered: PID and Finite State Machine (FSM). Though both controllers will produce similar results, they work in fundamentally different ways. The PID controller accepts the desired trajectory input and pixel coordinate error signal and computes a new trajectory to send to the robot on the fly to achieve the desired trajectory. However, the FSM builds and continuously modifies a cumulative 6-DOF compensation vector using vision-based measurements and the discrete states shown in Figures 3 and 4. This compensation vector is added to each desired 6-DOF trajectory input as it is sent to the impedance controller of the robot to achieve the desired pose. We are primarily concerned with the initial peg-and-hole alignment prior to insertion with this system. So the peg and the hole will be held stationary during our experiments, only rotating and translating as needed to align themselves from an initial starting position. This will allow the compensation vectors for the robot manipulating each part to become constant as the controller runs. This is because the vector is essentially compensating for errors in the kinematic modeling of the robot within the impedance controller, as well as compensating for the control forces weakened by the commanded compliance.

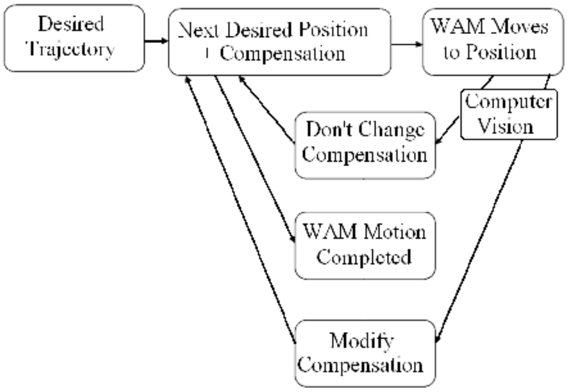

Figure 3. State diagram showing the logic and transitions of the visual finite state machine trajectory compensation controller for a single robot.

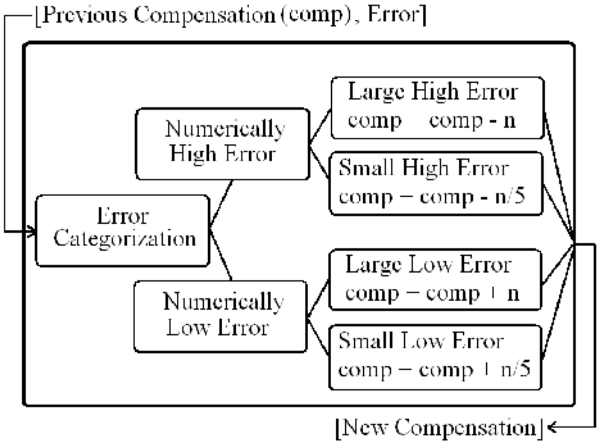

Figure 4. State diagram showing the transitions and logic of the compensation vector modification block from Figure 3 for a single DOF, with configurable compensation variable, *n*, and proportional divider (5 in this case) that allows the response of the system to be tuned for a particular task.

The block diagram representing the proposed closed loop control system is shown in Figure 5. In this figure, A(s) represents the impedance controller of the robot holding the peg while B(s) represents the reaction forces generated by the external forces like gravity and friction. These two systems essentially interact to form their own closed loop. The vision controller must precede this system to intercept and modify the incoming desired trajectory data to compensate for the positional errors of the system. The error signal is continuously generated from the vision system, so there is no sampling delay in the feedback measurements.

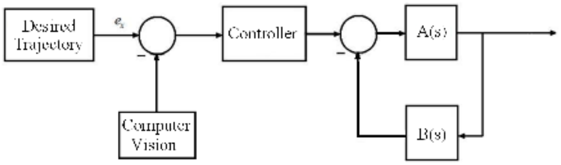

Figure 5. Block diagram showing the relation of the trajectory compensation system ("Computer Vision" and "Controller blocks") to the impedance control system. The "Controller" is represented either as a PID controller or the proposed FSM controller. For the independent impedance control system, A(s) represents the commanded motion of the peg and B(s) represents the reaction to the movement caused by environmental effects such as friction and gravity. The output of this system is the actual trajectory followed by the peg.

The first type of controller considered for use was the PID, or proportional-plus-integral-plus-derivative, controller. This type of controller was chosen because it typically offers very good transient and steady-state tracking behavior due to its derivative and integral actions, respectively. The Simulink block diagram shown in Figure 6 was created to model the behavior of the PID controller in response to both a step and sinusoidal input. To run simulations, the robot system was approximated as a spring-mass-damper system. From this approximation, a transfer function of input Cartesian force to output Cartesian velocity was achieved. Using this transfer function, an accurate impedance controller model is calculated and used for the plant during controller simulations. The PID controller was designed using the Matlab Control System Design Toolbox and tuned for the optimum trade-off with respect to steady-state error, rise time, and stability through Matlab's controller design GUI. The resulting PID control law, measured in Newtons, is shown in equation 2. The performance of this system is shown and compared to that of the FSM controller in the following paragraphs.

$$F(s) = 341 + 21903 \frac{1}{s} + (-.00853)\frac{39971}{1 + 39971\frac{1}{s}} \qquad (2)$$

The second type of controller considered for the visual feedback compensation loop takes a different approach. The FSM controller operates by accepting input error data and classifying it into one of two general magnitude categories. Then the controller transitions into the appropriate state to add or remove compensation from the cumulative compensation vector sent to the impedance controller. Then the FSM returns to its initial state to accept more data. The FSM designed for this system is shown in Figure 3.

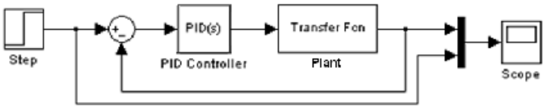

Figure 6. *Simulink* block diagram used to simulate the performance of the PID compensation controller in response to a step input.

This controller starts in the “Next Desired Position” state, in which the next desired Cartesian goal point for the robot is input into the system from the desired trajectory. The compensation vector, which is initially a zero vector, is also added to the goal point at this time. If the trajectory has not ended, the next goal point is sent to the impedance controller of the robot, which executes the command and moves the robot to the new position. The peg and hole should be aligned at all times during the assembly trajectory, so if the vision system detects that there is no error between the alignment of the peg and hole in the new position, the system moves to a state where no additional compensation is added to the system. If, however, the vision system detects that there is an error in the alignment of the peg and the hole, the system instead moves to the “modify compensation” state. In this state, each element of the system’s continuously modified 6-DOF compensation vector is either increased or decreased to correct the system’s alignment error. This state essentially classifies the error as greater than or less than several error threshold values, and increases or decreases the compensation vector a fixed amount that is proportional to the magnitude category of the error. Figure 4 shows the block diagram representing this process. This compensation vector is then added to the next desired goal point that is input to the impedance controller to compensate for the errors measured in the previous step.

Another Simulink model was created to test the step and sinusoidal input response performance of the FSM controller. The FSM structure was implemented using an embedded Matlab function because it runs slightly faster than the comparable Matlab Stateflow models. The completed simulation model is shown in Figure 7. This model uses the same transfer function used for the simulation of the PID controller.

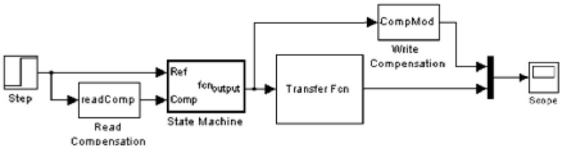

Figure 7. *Simulink* block diagram used to simulate the performance of the finite state machine in response to a step input.

Now that the simulation models for each controller type were completed, the performance of each could be compared. The first performance test measured the response of each controller to a step input. The results of each controller are shown in Figure 8. The PID controller has very good transient response with a 0.00888 second rise time. However it also demonstrates 10.5% overshoot and settles with some steady state error. This steady state error is due to the natural zero in the numerator of the transfer function used as the plant during simulation. The force-velocity relationship of the physical system represented by the transfer function will always have a zero when $s = 0$. This zero means the integral action of the PID controller will be unable to achieve zero steady state error. The FSM on the other hand, essentially operates as a proportional-plus-integral (PI) controller without this limitation. This is because the cumulative compensation method continuously adds compensation to the system whenever there is a non-zero error signal. The FSM controller has slower rise time due to its lack of derivative action. However, the FSM demonstrates 0% overshoot and no steady state error within the limits resulting from the camera resolution. Both of these factors, coupled with its reasonable rise time, suggest that the FSM controller is the better choice for this system. As shown in Figure 6, there are two distinct size classes for "large" errors and "small" errors. The transition of the amount of alignment error from one category (large) to the other (small) can be seen by the change in the slope of the response line in the Figure 8. The slope of this line can be adjusted as described in Figure 6 to achieve the desired response time.

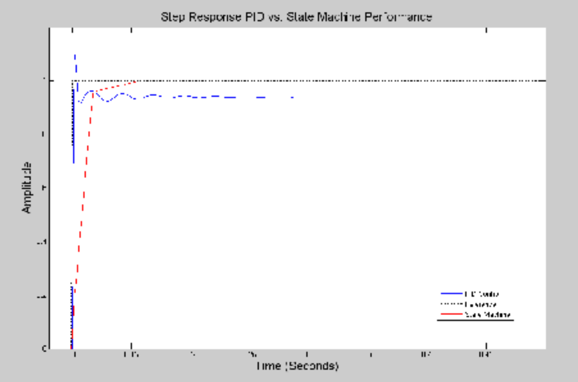

Figure 8. Step response of the PID trajectory compensation controller versus the step response of the FSM compensation system.

The next simulation demonstrated the ability of each controller to follow a transient input, which was modeled by a sinusoidal wave. The results of each controller for this test are shown in Figure 9. The same conclusion can be drawn from this simulation. While the PID controller has a slight advantage in transient response, the FSM controller has a significant advantage in steady-state error, which is the more important design goal for complete peg and hole alignment. Due to these simulations, the FSM controller was the design implemented in the final system.

The FSM controller also has another advantage over the PID controller that does not show in these tests. This advantage is that unlike the PID controller, a properly designed FSM cannot be unstable. This is due to the explicitly bounded compensation allowed by the system's programming. There has been limited research on the stability of state machine controller design. Gore presented two concepts that indicate a stable FSM design [14]. The first stability concept says that changes made to the behavior of a single state in the total state machine should not affect the implementation of the rest of the system. The second concept is similar to the idea of mesh size in numerical computation. This idea explains that the state machine should be designed such that every state is used during a run and there are sufficient interfaces to move to each state [14]. In another paper, Tarraf et al. [15] used a mathematical model of the finite state machine to

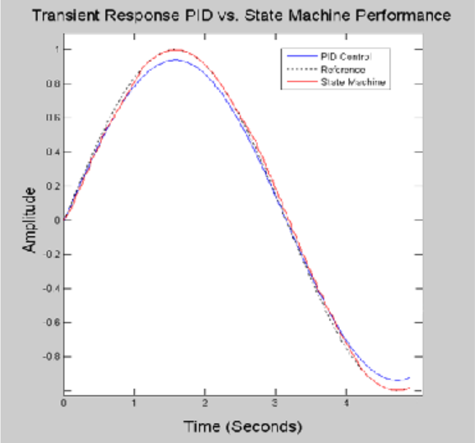

Figure 9. Transient trajectory following performance of the PID trajectory compensation controller and FSM compensation system for a sinusoidal input to the simulated robot between -1 and 1 meters.

determine the stability of the controller using a linear algorithm that checked the system for three properties. These properties were finite gain input-output stability, external stability, and incremental input-output stability. Each of these requirements focuses on the need of the system to handle inputs consistently and in a bounded way. The FSM presented in this section meets all of these requirements, so we can conclude that it will remain stable for all cases.

4. Experimental System Hardware

With the FSM controller completed, the system was physically tested using a cooperative robot assembly system composed of two Barrett Technology Whole Arm Manipulators (WAMs) [16], a two-camera vision system, reflective memory, and a control computer organized as shown in Figure 10.

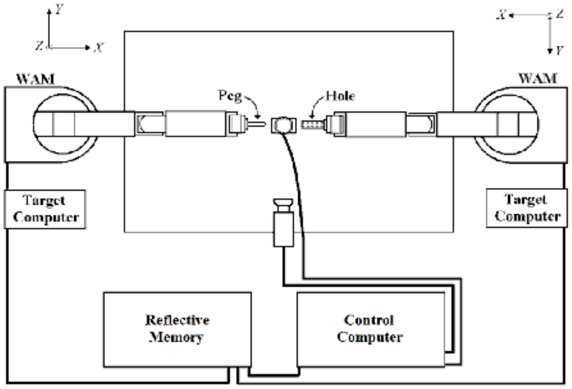

Figure 10. Organization of the cooperative robotic work cell used during experimentation. Also shown are the individual coordinate systems attached to each WAM.

Each WAM is a lightweight, seven DOF redundant robot arm that runs an impedance controller programmed through a compiled Simulink model. Each WAM communicates with the control computer through a reflective memory network. The control computer hardware consists of 2.53 GHz Intel Core2 Quad processor with four gigabytes of memory. This computer simultaneously sends trajectory commands to each WAM and runs the visual trajectory compensation controller.

These tasks are programmed as Simulink models that run compiled user-defined functions. Image data is input from two BASLER A602-fc cameras through FireWire connections.

The peg and hole end effectors used for these experiments are made from machined aluminum and are rigidly attached to the wrist of their respective WAM. These end effectors are much stiffer than the joints held in place by the control forces generated by the impedance controllers. This allows Whitney's wedging analysis of compliantly supported rigid parts from Section 2 to be applicable. Using the equations shown in Figure 2 to create a wedging diagram for our specific peg and hole hardware, the following limits for allowable initial lateral error, ε, and initial rotational error, θ, were calculated. These values will be used to measure the success of our experimental system with respect to the wedging diagram in Figure 2.

$$c = \frac{R - r}{R} = 0.0255107 \qquad R = 0.010035m \qquad r = 0.009779m$$

$$\mu = 0.45 \qquad \frac{c2R}{\mu 2r} = 0.05817 = \theta_{0_{MAX}} \qquad r + R = 0.019814 = \varepsilon_{0_{MAX}}$$

A basic computer vision system was implemented in Simulink to provide the feedback information needed by the trajectory compensation controller to align the peg and hole. The system consists of two cameras positioned orthogonal to the X-Z and X-Y planes, respectively. Only two cameras are needed because the cylindrical geometry of the parts makes axial rotations in the Y-Z plane insignificant. These cameras capture 320x240 RGB images and send them to the control computer. The control computer then applies a color filter to the image data and approximates the pixel-coordinate position and orientation of the peg and hole in each image plane. The peg and hole are colored red and blue, respectively, to aid object recognition. This simple setup does not require calibration since measurements are output in pixel coordinates. Also, the use of pixel based measurements offers sub-millimeter accuracy of the error signal with the image resolution and camera placement strategy used in this experimental setup. This system is able to update the visual feedback measurements at a rate of 80Hz.

5. Experimental Results

All experiments focus on the validation of the FSM visual alignment controller. As a result, the peg-and-hole end effectors are simply being aligned in the following experiments, and no insertion or assembly is taking place. The first test required the WAM holding the peg to move the tip of the peg to a height of 158 millimeters above its base. The hole end effector of the other WAM was commanded to a position where its center was exactly at a height equal to the target 158 millimeters, but located about ten centimeters in the positive X direction away from the peg's starting position. A short trajectory was then sent to the WAM holding the peg to command it to align itself with the hole. While the WAM was moving, it also output a structure containing its actual trajectory measured by the robot's joint encoders to Matlab. If all of the parameters of the impedance controller perfectly modeled the physical WAM system, it would be able to perfectly follow this trajectory and align the peg and the hole without help from the visual compensation system. However, this is not the case. Initially, the impedance controlled system actually does not reach its target alignment as a result of modeling error and the moderate stiffness gain of approximately 700 N/m applied to the impedance controller in all 3 lateral dimensions, i.e., the Kp of Equation (1). This is especially apparent in the vertical direction, which is the most affected by gravity. While the uncompensated peg may only be 1.5 millimeters below its goal height in this test, even small initial lateral errors increase the likelihood of the peg becoming wedged in its hole. For the next trial, the visual compensation loop was turned on to compensate for the droop in the WAM due to the impedance controller error. On this run, the peg is able to very accurately follow its target trajectory and align itself vertically with the hole. The results from this test are shown in Figure 11.

It was now time to allow the system to align the peg and hole for assembly. To perform this experiment, the peg and hole manipulators were commanded to have high compliance, with a lateral stiffness (K_p) of approximately 400 N/m along the X, Y, and Z axes and a rotational stiffness (K_θ) of 5 N-m/radian along all three rotational degrees of freedom. Each WAM was then sent trajectories to move the peg and hole to a point of near contact, where the two parts would be perfectly aligned. The position achieved by the WAMs using impedance control alone can be seen in the first panel of Figure 12. It is obvious that the high compliance and lack of closed loop feedback has caused there to be considerable rotational and translational error between the peg and hole after the gross movement. At this point, the visual trajectory compensator was activated to adjust the commands being sent to the peg and hole manipulators. For this experiment, the tips of the peg and hole were each sent a trajectory composed of a single desired pose to hold the peg and hole stationary and aligned with approximately 3 cm of space between them. The error initially present in the pose of the peg and hole would be adjusted by the FSM until the peg was aligned with the hole to achieve the desired position. After the alignment controller was run for 2.5 seconds, the peg and hole were sufficiently aligned as shown in the second panel in Figure 12, and were within the “no-wedge” box of Fig. 2.

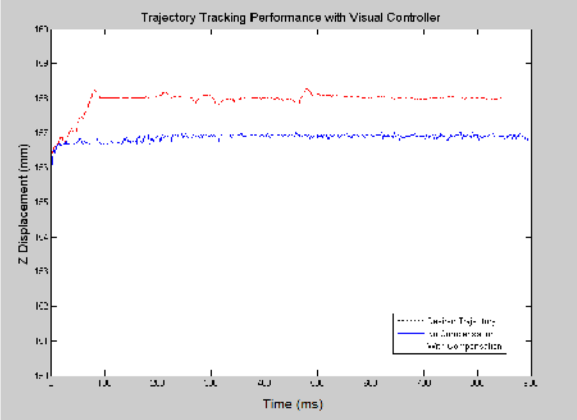

Figure 11. Actual trajectory following performance of the WAM in the most difficult vertical direction with and without visual compensation as a function of motion time.

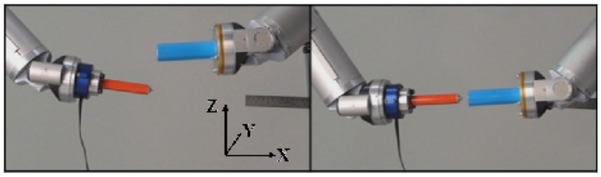

Figure 12. Left image shows the alignment achieved without visual compensation using low stiffness impedance controller parameters. Right image shows alignment achieved with visual compensation added to the input commands.

Figures 13 and 14 show the values of the compensation vector computed by the alignment controller and used to augment the commands being sent to the WAMs by the system controller. As previously mentioned, the manipulators are not moving a significant amount during this process, so we expect the compensation needed to account for the kinematic model errors and high compliance to achieve a constant, steady-state value. The physical system behaved as we expected, as shown in the following figures. Only the X- and Y- axis compensation commands are shown as no change was required along the axis of the peg to keep the desired 3 cm of space between the peg and hole.

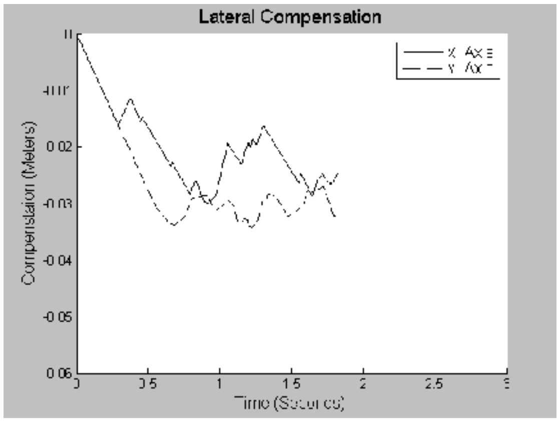

Figure 13. Compensation added to the lateral trajectory commands sent to the peg during alignment along the X and Y axes.

A final test was conducted to show the repeatability of the alignment controller and verify that it met our accuracy goals of 0.05817 radians of allowable rotational error and 0.019814 meters of allowable translational error as defined by the physical geometry of the peg and hole in Section 4 using the wedging diagram. The previous experiment was executed ten times. After the alignment controller was allowed to run for 2.5 seconds, the rotational error between the peg and hole was measured using a digital angle finder. Also, the lateral error between the longitudinal axis of the peg and the hole was measured using a pair of digital callipers. The total error was measured for both rotation and lateral position, instead of breaking the measurement into planar components, to assume the worst case error with respect to the wedging diagram. The maximum observed rotational error was 0.040 radians, with an average measurement of 0.022 radians. The maximum observed lateral error was 0.0032 meters, with an average measurement of 0.0025 meters. Using the maximum observed error, we can show that in the worst case scenario, we met our alignment goals with at least 31% margin in rotation and 84% margin in translation with respect to the wedging diagram.

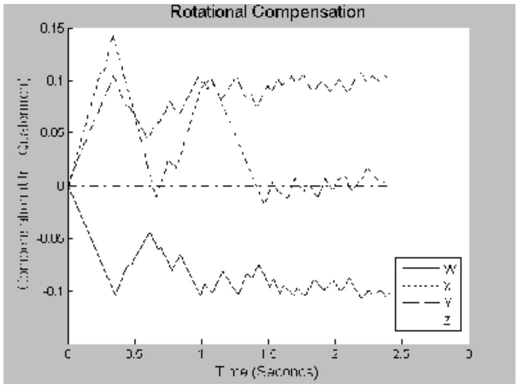

Figure 14. Compensation added to the rotational trajectory commands sent to the peg during alignment. Quaternion representation is used to define rotation, so there are four rotation elements required.

The FSM visual servoing controller implemented in the preceding experiments was allowed to run for a nominal 2.5 seconds. While this yielded very good results, the large margin with respect to the limits of the wedging diagram imply that less time is required to achieve adequate alignment to avoid failure due to wedging. In practical manufacturing applications, where speed is of great importance, the FSM could be redesigned to actively compare the lateral and rotational error measured by the vision system to the limits defined by the wedging diagram. This would allow the system to perform insertion as soon as the peg and hole were aligned well enough to protect the process from wedging, and reduce the total time needed for alignment, and the complete assembly process.

6. CONCLUSION

The experimental testing has proven that the finite state machine visual alignment controller successfully improved the positional accuracy of the impedance controller to meet our goals with respect to the wedging diagram. The result is a control system that is capable of aligning a peg and hole accurately enough to avoid wedging, while still allowing a wide range of compliance settings to provide desired behavior to complete a peg-and-hole assembly task. With these improvements, an impedance controller is capable of performing peg-and-hole insertion tasks more robustly than previously possible, by the active avoidance of wedging failure.

ACKNOWLEDGEMENTS

This work was supported by the US Department of Energy under Grant #DE-FG52-2004NA25590, issued to the University of New Mexico Manufacturing Engineering Program.

REFERENCES

[1] Nevins, J. L., Whitney, D. E., *Concurrent Design of Products and Processes: A strategy for the next generation in manufacturing*, McGraw-Hill Publishing Company, 1989.

[2] Drake, S. H., *Using compliance in lieu of sensory feedback for automatic assembly*, Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, 1977.

[3] Raibert, M. H., Craig, J. J., "Hybrid Position/Force Control of Manipulators," *ASME Journal of Dynamic Systems Measurement and Control, Vol. 102*, 1981, p. 126-133.

[4] Mason, M. T., "Compliance and Force Control for Computer Controlled Manipulators," *IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-11, No. 6,* June 1981, p. 418-432.

[5] Anderson, R. J., Spong, M. W., "Hybrid Impedance Control of Robotic Manipulators," *IEEE Journal of Robotics and Automation, Vol. 4, No. 5,* Oct 1988, p. 549-557.

[6] Whitney, D E., "Force Feedback Control of Manipulator Fine Motions," *Journal of Dynamic Systems, Measurement, and Control, June 1977, p*. 91-97.

[7] Hogan, N., "Impedance control: An Approach to Manipulation," *Transactions of the ASME Journal of Dynamic Systems, Measurement, and Control, Vol. 107,* March 1985, p. 1-24.

[8] Kelly, R., Carelli, R., Amestsgui, M., Ortega, R., "Adaptive Impedance Control of Robot Manipulators," *International Journal of Robotics and Automation, Vol. 4, No. 3,* 1989, p. 134-141.

[9] Kelly, R., Carelli, R., "Unified Approach to Adaptive Control of Robotic Manipulators," *Proceedings of the 27th Conference on Decision and Control,* 1988, p. 1598-1603.

[10] Kang, S. H., Jin, M., Chang, P. H., "A Solution to the Accuracy/Robustness Dilemma in Impedance Control," *IEEE/ASME Transactions on Mechatronics, Vol. 14, No. 3,* June 2009, p. 282-294.

[11] Soltanpour, M. R., Shafiei, S. E., "Design and Stability Analysis of a Robust Impedance Control System for a Robot Manipulator," *Studies in Informatics and Control, Vol. 19, No. 1,* March 2010, p. 5-16.

[12] Whitney, D. E., "Force Feedback Control of Manipulator Fine Motions," ASME Journal of Dynamic Systems, Measurement, and Control, June 1977, pp. 91-97.

[13] Asada, H., and Slotine, J.J.E., *Robot Analysis and Control,* John Wiley & Sons, New York, 1986, pp. 191-192.

[14] Gore, D.,"A Stable Design for the State Design Pattern," *International Journal of Software Engineering and Its Applications, Vol. 3, No. 4,* October 2009, p. 33-50.

[15] Tarraf, D. C., Dahleh, M. A., Megretski, A., "Stability of Deterministic Finite State Machines," *2005 American Control Conference,* 2005, p. 3932-3936.

[16] Barrett Technology, Inc. Website for the Whole Arm Manipulator (WAM), http://www.barrett.com/robot/products-arm.htm

Authors

Kevin Ruybal graduated from the University of New Mexico May 2009 with a BS in Mechanical Engineering. He received the MS in Mechanical Engineering from the University of New Mexico May 2011.

Ron Lumia received the BS degree in electrical engineering from Cornell University, Ithaca, NY, in 1972. He received the MS and PhD, both in electrical engineering, from the University of Virginia, Charlottesville, VA, in 1977 and 1979, respectively. His technical interests include computer vision, microrobotics, robot control, and manufacturing. He has been a Professor in the Mechanical Engineering Department at the University of New Mexico, Albuquerque, NM since 1994. From 1986-94, he was Group Leader of the Intelligent Control Group at the National Institute of Standards and Technology (NIST). Previously, he held academic positions at Ecole Superieure d'Ingenieurs en Electrotechnique et Electronique (ESIEE) in Paris, France, Virginia Tech in Blacksburg, VA, and the National University of Singapore. Professor Lumia remains active in IEEE Conference organization, serving IROS 2003 as Chair of Special Sessions, IROS 2007 as Finance Chair, ICRA1997 and 2006 as Exhibits Chair and ICRA 2010 as Finance Chair. He is currently serving as Treasurer for the IEEE Robotics and Automation Society.

John Wood is Professor of Mechanical Engineering, and Director of the Manufacturing Engineering Program at the University of New Mexico. Previously, he was Professor of Bioengineering, Associate Chair of Bioengineering, and Associate Director of the Center for Engineering Design at the University of Utah. Research interests include robotics, bioMEMS, biomechanics, quantitative physiology, manufacturing, and semiconductor processing and training. Prof. Wood was the General Co-Chair (with Prof. Roger Howe) of the 1990 IEEE MEMS Conference (Napa Valley, CA). He is presently the Coordinating Editor of the Micro Mechanics Section of Sensors and Actuators (Elsevier).