

Robotic Assembly of Complex Planar Parts: An Experimental Evaluation

P. Robuffo Giordano, A. Stemmer, K. Arbter and A. Albu-Schäffer

Abstract—In this paper we present an experimental evaluation of automatic robotic assembly of complex planar parts. The torque-controlled DLR light-weight robot, equipped with an on-board camera (eye-in-hand configuration), is committed with the task of looking for given parts on a table, picking them, and inserting them inside the corresponding holes on a movable plate. Visual servoing techniques are used for fine positioning over the selected part/hole, while insertion is based on active compliance control of the robot and robust assembly planning in order to align the parts automatically with the hole. Execution of the complete task is validated through extensive experiments, and performance of humans and robot are compared in terms of overall execution time.

I. INTRODUCTION

The target application of the presented system is to autonomously execute assembly operations of parts with complex geometry. To this end, we propose a combination of image processing, vision and force-torque control able to fulfill the steps required for such a high-level task, namely, visual object identification, fine robot positioning, picking and insertion strategies.

There are several aspects of this task that make the automation difficult. The requirement of high sensor capability, the complexity of the task programming, and the limited autonomy and flexibility of industrial robots are certainly some of them, leading to performance and success rates much below those of humans.

Indeed, typical industrial settings rely on highly structured environments like, e.g., dedicated part feeders for avoiding uncertainties during the picking of the parts, fixed (and known) locations of the mounting holes, or simple image processing setups with carefully adjusted illumination conditions. Chamfered parts and specially designed compliant elements for a given task are typically used to simplify the insertion problem. In contrast, the presented work aims at realizing a more flexible and versatile system, with robustness and insertion velocity superior to that of industrial setups. A vision system using natural light conditions and the ability to track even moving parts with the robot during assembly brings a higher degree of flexibility. For assembling general chamferless planar parts with tight tolerance, the presented system utilizes joint torque sensing and Cartesian compliance control. Therefore the system can be easily programmed for new parts. The additional flexibility is relevant in reducing the industrial costs when considering parts with new shapes (which in general would need a new feeder and passive compliance design), and in minimizing the failure rate if

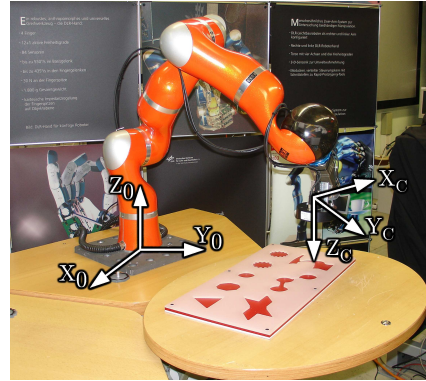


Fig. 1: The DLR light-weight robot equipped with an onboard camera.

unexpected events occur (e.g., unexpected displacements of parts/plate).

Use of vision information for pose control of robot systems has been an active topic in the last decades [1], [2]. Increase of flexibility, improvement of position accuracy, robustness against sensor noise and calibration uncertainties, and re-activeness to environmental changes are among the major benefits. One powerful use of vision for robot pose control is the approach known as *visual servoing*. In this framework the robotic task is directly specified in terms of some image features extracted from a target object. These features are then used to control the robot/camera motion through the scene until the final robot pose is reached, see [1], [3], [4] for a thorough overview. Visual servoing framework provides the needed robustness w.r.t. external disturbances and environmental changes, such as, for instance, the displacement of parts/plate during the task execution. The resulting robot positioning accuracy, however, may not be high enough to accomplish tight assembly tasks, so that, depending on the clearance of the parts, means to limit the contact forces and strategies to overcome the uncertainties are necessary. Industrial robots used for assembly are usually equipped with a passive compliance element, the so-called *Remote Center Compliance (RCC)* [5]. As we used the DLR light-weight robot equipped with joint torque sensors for our experiments, we instead exploited the Cartesian impedance control framework [6], [7] as a means to achieve an active compliance behavior. This solution is more flexible compared to a RCC while being faster and more robust than usual force controlled industrial robots, due to the local torque feedback and the low mass of the arm.

For the alignment of the parts, different strategies have been proposed in literature, starting from a simple (since model-free) but effective blind search [8], up to algorithms

The authors are with the German Aerospace Center (DLR), Oberpfaffenhofen, D-82234 Wessling, Germany {paolo.robuffogiordano, andreas.stemmer, klaus.arbter, alin.albu-schaeffer}@dlr.de

with complex online monitoring of the contact state [9], [10]. Due to the complexity of the parts used in our experiments, we relied on an off-line planning algorithm comparable to [11] in order to generate robust insertion strategies. In particular, by analyzing the parts geometry, the offline planner generates a set of optimal assembly trajectories and active compliance parameters that guarantee robustness w.r.t. initial positioning uncertainties due to vision and robot. Apart from achieving robustness, this planning stage also allows a fast setup of the system when parts with new shapes are considered. Since the insertion strategy has already been presented in [12], the focus of this paper will be on the vision part and the experimental evaluation of the combination of both vision and force feedback for successful assembly.

The paper is organized as follows: in Sect. II a description of the overall experimental setup is given. Next, Sect. III addresses the shape extraction and classification techniques used for visual recognition of the parts. The algorithm for robot pose control and the insertion strategy are outlined in Sect. IV and Sect. V, respectively. Finally, Sect. VI presents experimental results of the complete task execution and a comparison to the performance of humans.

II. EXPERIMENTAL SETUP DESCRIPTION

A. Robot manipulator

The DLR light-weight robots (Fig. 1) are designed for interaction with unstructured, everyday environments. They are kinematically redundant arms with 7 degrees of freedom (dofs) and a load to weight ratio of 1:1. Low weight and inherent joint compliance limit the interaction forces, even at high contact speed with the environment. Particularly relevant to the present application is the torque sensing integrated in each joint, allowing accurate, vibration free positioning and velocity control in the presence of elasticity, and high performance impedance control during contact phases. The robot is able to switch within one control cycle (1 ms) between position/velocity control (required for high motion accuracy during the visual servoing phase) to impedance control (required for limiting the interaction forces and compensating the alignment errors during insertion).

B. Parts and plate

The 8 parts and the plate used in our experiments are shown in Figs. 2(a–b). Each part has its shape marked with black tags useful for visual recognition, and a clearance of less than 0.1 mm w.r.t. the corresponding hole on the plate. Shapes of parts/holes range from simple geometrical primitives to more complex and nonconvex structures. Let $\mathcal{P} = \{p_1, \dots, p_8\}$ and $\mathcal{O} = \{o_1, \dots, o_8\}$ be the sets of parts and holes, respectively. With reference to Fig. 2(b), and starting from the upper left corner, we have the following shapes in order:

- o_1 (p_1): an equilateral triangle;
- o_2 (p_2): a regular octagon;
- o_3 (p_3): a circle;
- o_4 (p_4): a star shape with 15 teeth (Fig. 2(c));
- o_5 (p_5): a star shape with 16 teeth (Fig. 2(d));
- o_6 (p_6): the DLR logo;
- o_7 (p_7): the PAPAS logo;
- o_8 (p_8): the KUKA logo.

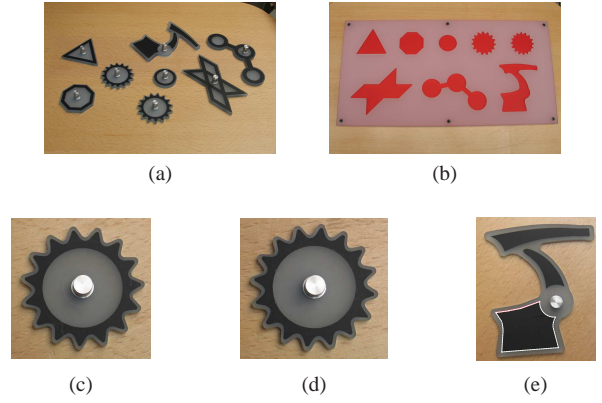


Fig. 2: The 8 parts (a) and the plate (b) used in our experiments. Figs. (c–e) show a detailed view of the star-shaped parts with 15 and 16 teeth, and of the KUKA part where the black tag used for shape classification is highlighted.

C. The overall task

As explained in the introduction, the global high-level task assigned to the manipulator is to locate, pick and insert each part into the corresponding hole on the plate. A suitable temporal decomposition of such a complex task is the following: for each part $p_i \in \mathcal{P}$ and corresponding hole $o_i \in \mathcal{O}$

- 1) hover along a predefined trajectory until p_i is identified and located;
- 2) move to a suitable pose in order to pick p_i ;
- 3) pick p_i ;
- 4) hover over the plate along a predefined trajectory until o_i is identified and located;
- 5) move to a suitable pose in order to insert p_i ;
- 6) insert p_i in o_i .

The algorithms needed to complete each step in the sequence are illustrated in the next sections. In particular, during steps 1 and 4, the vision system task is to recognize a target shape among the various parts/holes present in the images. Robot motion in steps 2 and 5 is governed by visual servoing techniques, while piece insertion in step 6 is realized through a robust force-torque controlled assembly strategy.

III. SHAPE RECOGNITION

This section deals with the techniques adopted to extract and classify the shapes of parts and holes from the video-rate camera images. An initial image preprocessing step, based on standard color binarization, is first applied to the raw camera images in order to mark pixels as belonging or not to a hole/part. Holes are segmented by looking for their red background color, while, for parts, we relied on the black outer tags glued on each of them (see Fig. 2). The subsequent steps of feature extraction and shape classification are based on the *affine-invariant Fourier descriptors* and *linear MSE classification* techniques and are detailed in the following subsections.

A. Affine-invariant Fourier descriptors

Consider two closed curves in the image plane \mathcal{C}^0 and \mathcal{C} representing the boundary of a planar object under two

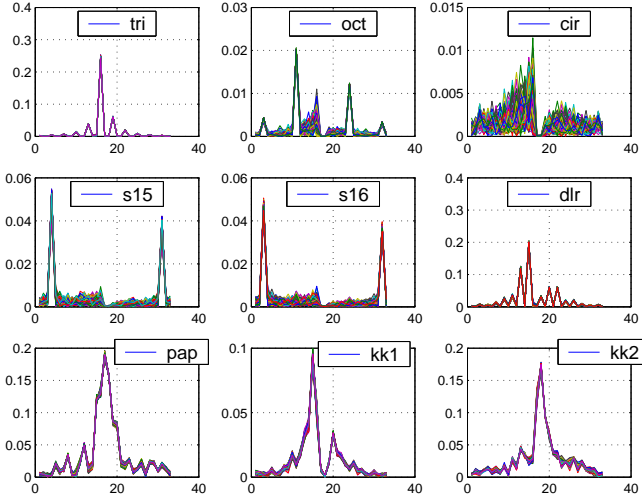


Fig. 3: Superimposed values of the 33 AIFDs chosen as features for pattern classification. For each shape in \mathcal{S} , we considered 80 observations taken from different camera views. The selected AIFDs prove to be sufficiently insensitive w.r.t. noise and different viewing angles (robustness), and show specific patterns from class to class (selectivity).

different camera views. Assuming a parallel projection¹, \mathcal{C}^0 and \mathcal{C} would be equivalent with respect to the affine transformation group: $\mathcal{C}^0 \sim \mathcal{C}$. Under this hypothesis, the set Q_k , $k = \pm 1, \pm 2, \dots$, of affine-invariant Fourier descriptors (AIFDs) [13], [14] provide a solution to the problem of finding a discrete, minimal, and complete set of features able to univocally encode the common shape information shared by \mathcal{C} and \mathcal{C}^0 despite the different camera points of view. Hence, AIFDs yield a suitable choice for shape classification and, thanks to their ordered discrete nature, can be easily compared in contrast to the initial contour representation with its inherent correspondence problem. Therefore, also because of their easy implementation, we chose them as features to solve our shape classification problem.

B. Contour classification

With the sole exception of part p_8 (KUKA logo — see Fig. 2(e)), the black tags on the parts match the shapes of the corresponding red holes. Therefore, for what concerns contour classification, all the segmented parts/holes shapes can be collected in a set of 9 classes

$$\mathcal{S} = \{tri, oct, cir, s15, s16, dlr, pap, kk1, kk2\}, \quad (1)$$

where the last two classes account for the distinct shapes of the KUKA part/hole². The goal of the classifier algorithm is then to decide for one of these classes given as input a feature vector λ describing a candidate part/hole shape. Once

¹Parallel projection model [1], sometimes called *weak perspective* model, is the best linear approximation of the standard pin-hole camera projection model. The approximation is very close if the mean distance along the optical axis between the camera and the object is large compared to the distance variation.

²Obviously, the introduction of the two distinct classes $\{kk1, kk2\}$ could be avoided by segmenting the whole KUKA part in place of the black tag glued on it (see Fig. 2(e)), thus obtaining a single class for both KUKA part/hole. However, in this case we chose to tolerate this little “redundancy” because of the generally higher reliability of the segmentation process for the black tags.

a class is identified, the color information (black/red) of the raw image allows to univocally disambiguate between those holes and parts belonging to the same class in \mathcal{S} .

We chose to consider $n_\lambda = 33$ different AIFDs Q_k for the classification task, and to define vector λ as

$$\lambda = [Q_{-18} \dots Q_{-2} Q_2 \dots Q_{17}]^T \in \mathbb{R}^{n_\lambda}.$$

Now assume that, for a class $s_i \in \mathcal{S}$, N_i distinct observations are taken from different camera views, resulting in N_i feature vectors λ_i . Let the $n_\lambda \times N_i$ matrix $\lambda^{[N_i]} = [\lambda_1 \dots \lambda_{N_i}]$ be the collection of such feature vectors. Because of the invariant properties of AIFDs, we expect the columns of $\lambda^{[N_i]}$ to match almost completely, with the only discrepancies due to noise or lack of validity of the parallel projection model. Figure 3 shows the superimposed values of the columns of $\lambda^{[N_i]}$, with $N_i = 80$, for all the classes in \mathcal{S} . From these plots, we can verify the low variation of individual feature elements within each class despite noise and different viewing angles (robustness), and the presence of specific patterns from class to class (selectivity). These preliminary considerations motivated us to use a set of linear discriminant functions, obtained by the so-called MSE (minimum squared error) approach [15], in order to solve the classification problem. This method is optimal in a least-square sense and requires less computational effort than more sophisticated techniques.

At the core of the linear MSE approach is a $9 \times (n_\lambda + 1)$ matrix Θ , called *weight matrix*, which is used against a given feature vector λ in order to decide the class to which it belongs. Suppose that a vector λ_i , observation of a class s_i , is considered. Matrix Θ is designed such that, by letting $\gamma_i = \Theta \lambda_i$, $\gamma_i \in \mathbb{R}^9$, the i -th component of γ_i will (most likely) be close to 1 and the others close to zero. Hence, class i can be decided through inspection of γ .

In order to assess the performance of the linear MSE approach, we tested it against $N_i = 40$ independent feature sets for every class i , for a total of 360 samples collected by varying continuously the camera pose. Figure 4 (left) shows the responses of the 9 linear discriminant functions (the components of γ) to the set of $9 \times N_i$ test samples. It is then possible to verify that, for each class s_i , only one component of γ is close to 1 (the i -th one), while the other components remain almost zero. Therefore, AIFDs allow to unambiguously assess the correct class membership for the candidate shapes in \mathcal{S} .

As a comparison, we tested the same linear MSE approach by relying on the popular 7 Hu moments [16] often used for pattern recognition of arbitrary shapes. The classification results are shown in Fig. 4 (right). It is interesting to note that, for the $\{tri, dlr, pap, kk2\}$ shapes, the outcome is substantially equivalent to the previous case, i.e., the correct component of γ keeps close to 1, the others are almost zero, and class membership can be easily decided. On the other hand, when considering the remaining $\{oct, cir, s15, s16, kk1\}$ shapes, results become ambiguous: the strong separation obtained before is lost, and several components of γ assume intermediate values in the range $[0, 1]$, thus preventing a robust classification of the shapes. Indeed, a finite set of low order moments (such as the Hu’s one) does not allow to

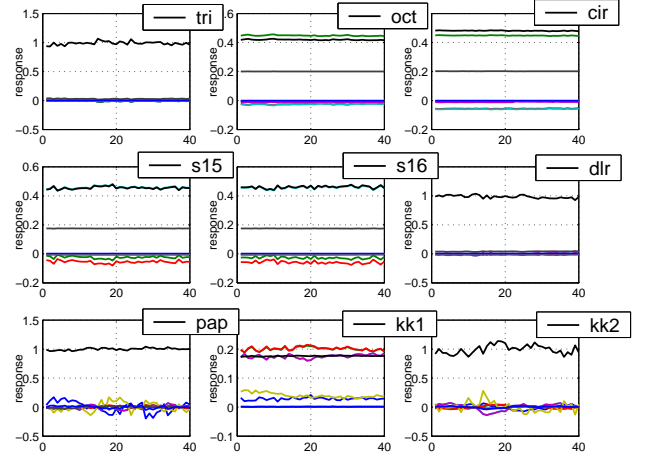
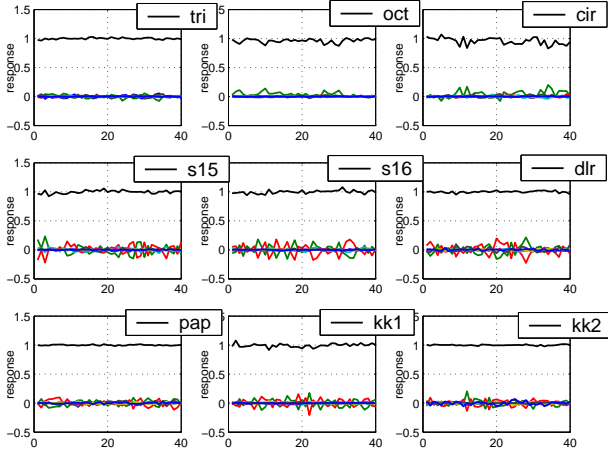


Fig. 4: The response of the linear MSE approach to a set of 360 testing samples by using AIFDs (left) and Hu moments (right).

discriminate among a circle, an octagon, or the two stars: in an *integral* sense such shapes share almost the same circular-like contour, and any small deviation is dominated by noise. AIFDs, on the other hand, prove to possess the needed resolution for a robust classification of all classes thanks to their high sensitivity to periodic patterns in the input contour.

IV. ROBOT POSE CONTROL

As explained in Sect. II-C, during the part/hole approaching phase motion of the robot is controlled by means of visual servoing techniques. The goal is to position the manipulator close and precisely enough to the selected part/hole such that the part can be picked or the insertion strategy can be started. An example representative of both picking and insertion final poses is given in Fig. 1: let $\mathcal{F}_O : \{X_O, Y_O, Z_O\}$ be the inertial manipulator base frame and $\mathcal{F}_C : \{X_C, Y_C, Z_C\}$ be the moving camera frame with Z_C coincident with the optical axis, and let vector $[v_C^T \ \omega_C^T]^T \in \mathbb{R}^6$ represent the camera linear/angular velocity expressed in \mathcal{F}_C . The setup is arranged such that the (parallel) planes where parts and plate lie are fixed in \mathcal{F}_O and perpendicular to Z_O . In order to meet the assumption of (almost) parallel projection needed by AIFDs, we chose the final pose of the manipulator such that $Z_O = -Z_C$, i.e., with the camera optical axis normal to the parts/plate plane and directed towards it. This choice also simplifies the servoing scheme, since 2 camera dofs (direction of Z_C) can be directly controlled in Cartesian space by exploiting the robot forward kinematics. The remaining 4 dofs (camera translation and rotation about Z_C) are instead regulated by means of visual information.

A. Task definition

Choice of which image features to exploit for visual control largely depends on the specific case one has to deal with, and can range from point coordinates, line parameters, ellipse centers and radii, etc., see [17] for a thorough analysis. When considering complex shapes, solving the correspondence problem, i.e., reliably tracking local structures like corners or edges, may result in a hard task. In these cases, more

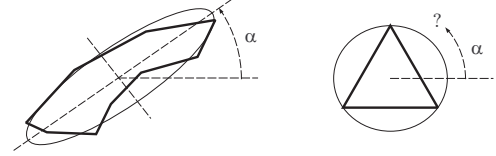


Fig. 5: Main orientation α is well-defined for a generic shape (left), but is meaningless for a shape where minor and major axes are equal (right).

general (integral) image features, such as Fourier descriptors or image moments, may be used. Whatever the choice, however, the availability of the so-called *interaction matrix*, i.e., the differential link between ‘feature motion’ and camera motion, is always required when devising visual control schemes. While the explicit expression of the interaction matrix for any image moment is known [18], a similar result for generic Fourier descriptors (or combinations of them) is not available except for special situations. On the other hand, as highlighted in the previous section, Fourier descriptors possess the ability to discriminate among the planar rotation of all the considered shapes, as opposite to standard (low order) image moments. Therefore, these considerations led us to adopt both image moments and Fourier descriptors for robot pose control.

Indeed, from [18], it is known that area a , barycenter (x_g, y_g) and main orientation α can be directly exploited to control the needed 4 camera dofs. However, in our case the ‘circular’ symmetry of most shapes ($o_1/p_1 \dots o_5/p_5$) prevented the use of α to control rotations about Z_C . Consider Fig. 5: for a generic shape, α gives the major axis orientation of the ellipse which fits ‘best’ to the shape. Clearly, if a circular-like shape is considered, no major axis can be extracted and α becomes meaningless. This problem can be overcome by replacing α with an equivalent quantity obtained from Fourier coefficients, i.e., the relative rotation ρ among two shapes belonging to the same class. Assuming a camera optical axis almost perpendicular to the object plane, angle ρ can be determined from two Fourier coefficients up to an ambiguity of f , i.e., a rotation ambiguity of $2\pi/f$ rad,

where f is the degree of symmetry of the selected shape (with $f = 1$ for nonsymmetric cases). Therefore a ‘de facto’ unique solution for angle ρ can always be found by exploiting Fourier coefficients, and ρ can be fully exploited for what concerns control of the rotation about Z_C .

While the interaction matrix of area a and barycenter (x_g, y_g) has a closed-form expression in terms of image moments, the interaction matrix of ρ can be obtained as follows: when Z_C is almost perpendicular to the object plane, i.e., almost parallel to Z_O , $\rho(t)$ behaves analogously to the camera roll angle³ $\phi(t)$ in the sense that $\dot{\rho} = f\dot{\phi}$. Hence, by letting $\dot{\phi} = T_\phi \omega_C$, where 1×3 matrix T_ϕ is the standard mapping from angular velocity ω_C to Euler angle rate $\dot{\phi}$, we have

$$\dot{\rho} = fT_\phi \omega_C = J_\rho \omega_C, \quad (2)$$

where matrix J_ρ can be regarded as the Jacobian of ρ .

As for the direction of Z_C , it can be controlled in Cartesian space since, as explained before, parts and plate plane orientations are fixed w.r.t. the manipulator base. Therefore, it is possible to obtain the direction of Z_C w.r.t. \mathcal{F}_O , and thus w.r.t. the plane of parts and plate, directly through the robot forward kinematics. In our case, the robot final poses are designed with the camera optical axis perpendicular to the parts/plate plane, i.e., with $Z_C = -Z_O = [0 \ 0 \ -1]^T$ in \mathcal{F}_O (see Fig. 1). Hence, a convenient choice is to regulate to zero the first two components (Z_{C_x} , Z_{C_y}) of Z_C through the differential mapping

$$\begin{bmatrix} \dot{Z}_{C_x} \\ \dot{Z}_{C_y} \end{bmatrix} = J_{Z_C} \omega_C, \quad (3)$$

where the Jacobian J_{Z_C} can be obtained from standard kinematics. Note that regulation of $[Z_{C_x} \ Z_{C_y}]^T$ to $[0 \ 0]^T$ through inversion of (3) admits the two stable equilibria $[0 \ 0 \ 1]^T$ and $[0 \ 0 \ -1]^T$ for Z_C , depending on the initial vertical pointing direction of the optical axis. In our case, this ambiguity is avoided because the initial camera pose is always such that Z_C points downwards (the camera looks towards the parts/plate).

Having defined all the needed quantities, we let

$$s = [x_g \ y_g \ a \ \rho \ Z_{C_x} \ Z_{C_y}]^T \in \mathbb{R}^6 \quad (4)$$

be the task vector used for robot pose control, and proceed to illustrate the control algorithm used for regulation of $s(t)$.

B. Control algorithm

Regulation of task (4) to a desired value s^* is addressed at the velocity level by inversion of the differential mapping $\dot{s} = J_s \dot{q}$, where $q \in \mathbb{R}^7$ stands for the manipulator joint configuration vector and J_s represents the 6×7 task Jacobian matrix in terms of joint velocities. Note that our robot is redundant w.r.t. task s , with degree of redundancy 1. Among the various techniques available for velocity-level motion and redundancy resolution [19], we chose the *Projected Gradient* (PG) method which generates joint velocity commands in terms of a reference task velocity \dot{s}_r as:

$$\dot{q} = J_s^\dagger \dot{s}_r + \sigma(I - J_s^\dagger J_s)u_0, \quad \sigma > 0, \quad (5)$$

³Given a ZYX Euler angles representation of the orientation of \mathcal{F}_C w.r.t. \mathcal{F}_O , roll angle ϕ is defined as the rotation about Z-axis.

where J_s^\dagger is the Moore-Penrose pseudoinverse of J_s , vector $u_0 \in \mathbb{R}^7$ the gradient of a given criterion $H(q)$ to be optimized through the redundancy, and σ a suitable optimization stepsize. Expression of the task Jacobian J_s can be obtained from

$$\dot{s} = \begin{bmatrix} J_m \\ 0 & J_\rho \\ 0 & J_{Z_C} \end{bmatrix} \begin{bmatrix} v_C \\ \omega_C \end{bmatrix} = \begin{bmatrix} J_m \\ 0 & J_\rho \\ 0 & J_{Z_C} \end{bmatrix} J_c \dot{q} = J_s \dot{q}. \quad (6)$$

Here, J_ρ and J_{Z_C} are defined in (2) and (3), J_c , from forward kinematics, gives the camera velocity (v_C, ω_C) in terms of \dot{q} , and the 3×6 matrix J_m is the area a and barycenter (x_g, y_g) interaction matrix

$$\begin{bmatrix} \dot{a} \\ \dot{x}_g \\ \dot{y}_g \end{bmatrix} = J_m \begin{bmatrix} v_C \\ \omega_C \end{bmatrix}.$$

In case of regulation tasks, vector \dot{s}_r in (5) is typically chosen as

$$\dot{s}_r = K(s^* - s), \quad K > 0. \quad (7)$$

This choice yields an exponential convergence to zero of task error $e(t) = s^* - s(t)$ for any initial error $e(t_0)$ and with convergence rate tuned by gain matrix K . In practical implementations, however, because of neglected velocity controller dynamics, external disturbances, limited actuator capabilities, modeling errors, noise, etc., a large initial error coupled with a big K may lead to instability or, in the visual servoing case, to the loss of visual features during the motion, causing failure of the task. In our case, for instance, the completely unmodeled robot dynamics proved to be a major limiting factor when imposing fast transients. Apart from reducing K , such effects can be attenuated by avoiding large values for $e(t)$, and in particular for $e(t_0)$. To this end, we added a planning stage to the control law by defining an artificial signal $s_d(t)$ which linearly interpolates the initial task value $s(t_0)$ with the final desired value s^* , thus obtaining $e(t_0) = s_d(t_0) - s(t_0) = 0$. In practice, we impose exactly the same motion direction in task (and hence image) space as in (7), but with the reference task trajectory $s_d(t)$ traveling with constant velocity along the line connecting $s(t_0)$ and s^* .

With these settings, vector \dot{s}_r becomes

$$\dot{s}_r = K(s_d(t) - s(t)), \quad (8)$$

yielding a linear exponentially stable closed-loop error system driven by \dot{s}_d

$$\dot{e} = -Ke + \dot{s}_d.$$

Finally, the optimization function $H(q)$, from which $u_0 = \nabla_q H(q)$ is derived and used in (5), is designed for joint limit avoidance [19].

V. ROBUST ASSEMBLY STRATEGY

After the visual positioning task over the selected hole is completed, the insertion has to be executed without vision support, since in this phase, the target hole exits the camera field of view. By exploiting sensitive Cartesian compliance control, however, local convergence of the assembly task and

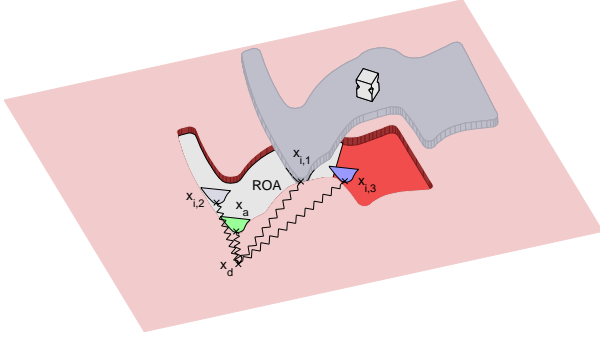


Fig. 6: A typical region of attraction (ROA) for a sample part. The inserted corner will be guided automatically to position x_a if the alignment process starts anywhere within the ROA (e.g. from $x_{i,1}$ or $x_{i,2}$). If it starts outside (e.g. from $x_{i,3}$), a successful alignment cannot be guaranteed.

limitation of contact forces can be ensured under the remaining positioning uncertainty. The parts have no chamfers, therefore insertion is not trivial and an optimized strategy is needed in order to align and insert the parts robustly. This section will outline the basic idea of the insertion strategy, a detailed description of the algorithm can be found in [12].

The main idea of the insertion planning is visualized in Fig. 6. Consider the compliance controlled robot having inserted a corner⁴ of the part into the hole at the initial configuration x_i . The desired position of the controller is now set to x_d , and the stiffness value to K . For a certain set of starting configurations (called the *region of attraction ROA*), the inserted part will converge to the desired alignment position x_a . In the given example, $x_{i,1}$ and $x_{i,2}$ belong to the ROA, $x_{i,3}$ does not. The alignment can be seen as the settling of a nonlinear dynamical system with several equilibria whereof one is the desired configuration. It is possible to determine the ROA for any desired equilibrium x_d and for any stiffness matrix K . Its size can be used as a direct measure for the robustness of the assembly trajectory, the optimal robustness is achieved for those insertion parameters that maximize the ROA.

Obviously, the ROA depends heavily on the inserted corner, the selected desired and initial positions x_d and x_i , the parameters of the impedance control (in particular K), and the shape of the hole. Whereas the latter is given, the remaining parameters can be selected freely and are used for offline optimization. Combined with a user interface for providing the geometries from a CAD system or from sensor data, this results in a toolbox for industrial robot programmers to generate robust assembly programs automatically. The output of this toolbox, desired trajectories and control parameters, can then be used in the execution phase without any model knowledge about the parts. If the uncertainty of the visual servoing and the robot positioning (leading to a deviation of x_i) is smaller than the best ROA, it is thus possible to rely on this offline planning and execute the planned motion starting from the final pose reached with the visual servoing algorithm.

In Fig. 7, a complete example of an assembly sequence

⁴Corner in this context means the relevant part of the contour which is involved in a one-point contact.

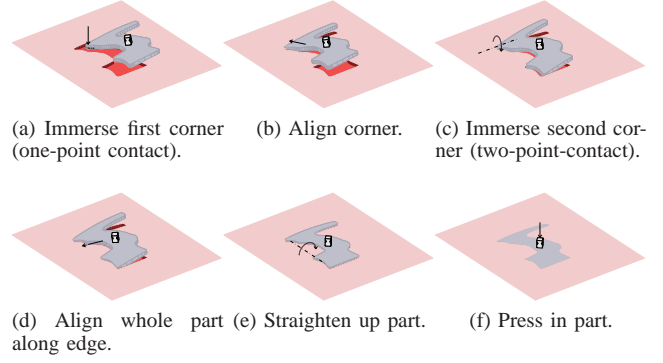


Fig. 7: Example of the presented insertion strategy.

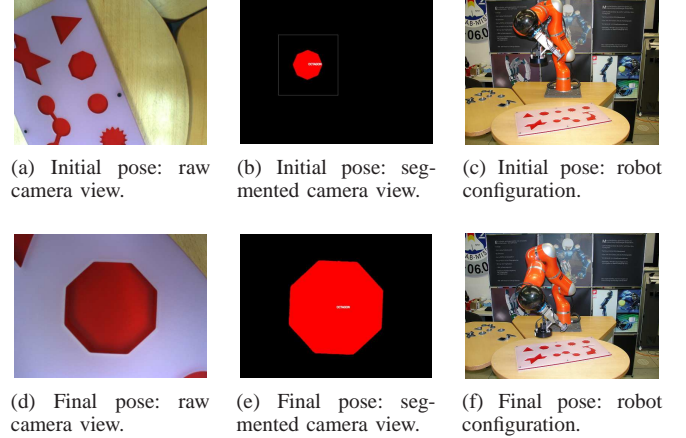


Fig. 8: Initial and final poses during the picking of part p_2 .

relative to the KUKA logo is shown. The assembly starts by slightly tilting the part and then by immersing a first corner into the hole. In order to have a smooth motion, the desired position is interpolated from x_i to x_d . Prerequisite for the success is that the initial position of the immersed corner lies within the ROA of the appropriate hole's corner. After the first corner is aligned properly, a second contact point is needed to ensure the rotational alignment of the part. Therefore, the same compliance based procedure can be used. Having reached a stable two point contact, it is easy to straighten up the part and insert it completely. Since the parts considered are planar (their height is small compared to their lateral dimensions), jamming during the insertion is a minor problem if the center of compliance of the impedance control is chosen correctly.

VI. EXPERIMENTAL EVALUATION

In this section we present an experimental assessment of the whole application. First, we analyze the visual control law detailed in Sect. IV in terms of execution time and control effort. Then, robustness and reliability of the overall method are extensively tested on a run of 20 complete sequences. Finally, we propose a comparison among the execution time needed by the robot and humans (children of age 5–7 and adults).

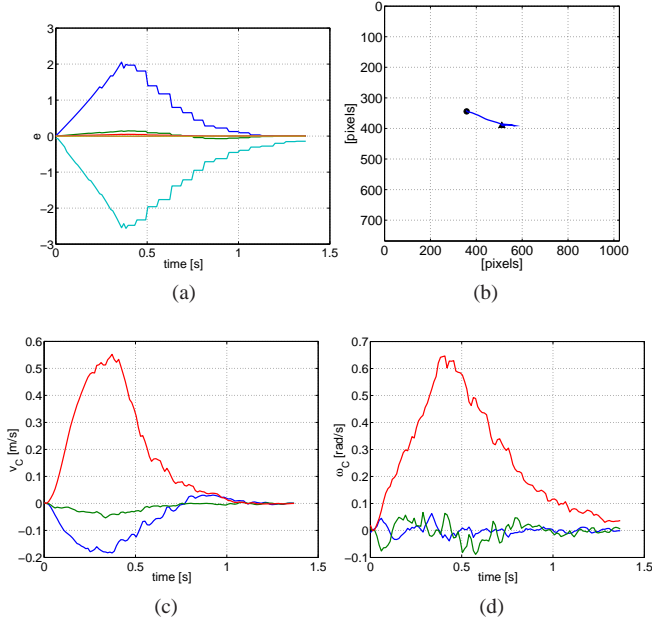


Fig. 9: Experimental data during the approaching to hole o_2 . Fig. (a): the 6 components of $e(t)$. Fig. (b): Motion of barycenter (x_g, y_g) on the image plane from \bullet —start to \blacktriangle —end. Fig. (c): Camera linear velocity v_C . Fig. (d): Camera angular velocity ω_C .

A. Visual Control

We consider the approaching task to hole o_2 (the octagon), see Figs. 8(a–f) for screenshots of the initial and final poses. As explained in Sect. IV-A, the circular symmetry of this part poses a challenge for visual control, since main orientation α cannot be used to control camera rotation about Z_C . On the other hand, by using angle ρ from Fourier coefficients, we get $\rho(t_0) = 2.8$ rad and $\rho^* = -0.08$ rad as initial and final values (Figs. 8(b) and 8(e)). Recall that ρ measures shapes rotation up to their degree of symmetry f , i.e., for an octagon with $f = 8$ a physical rotation of $2\pi/f$ rad results in a 2π rad rotation for ρ . Figures 9(a–d) show some relevant quantities collected during the experiment. As can be seen from the plots, the overall motion is quite fast, lasting about 1.3 sec. with a peak value of $\|v_C\| \simeq 0.6$ m/s and $\|\omega_C\| \simeq 0.65$ rad/s. In Fig. 9(a) the behavior of $e(t) = s_d(t) - s(t)$ is reported. Note that, as expected, $e(t_0) = 0$ because of the definition of the reference signal $s_d(t)$ (Sect. IV-B). As a consequence, no initial jump is present in the commanded camera velocity (v_C, ω_C) (Figs. 9(c), 9(d)). Avoiding velocity jumps during fast transients, like in our case, has a major relevance. Indeed, since robot dynamics is neglected, a too high acceleration request, coupled with an high speed profile, would violate this approximation and potentially lead to failure of the visual task. Finally, Fig. 9(b) shows the image plane motion of the octagon barycenter (x_g, y_g) during the servoing, which results close to a straight line as a consequence of the decoupling properties of control (5)–(8).

B. Complete sequence

Experimental results of the insertion strategy only (without vision) have been published in [12]. For the complete sequence, we recorded statistical data over 20 cycles, that

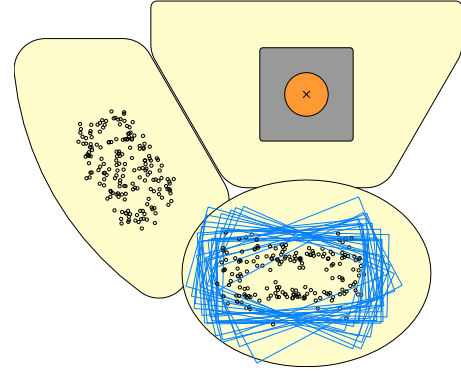


Fig. 10: Top view of the distribution of parts and plate during the statistical evaluation of the automated assembly. The small circles represent the gripping handle for parts in the starting position (left) and inserted in the plate (bottom).

is we collected data from 160 assemblies (see Fig. 10 for the distribution of the parts and the plate). Altogether, the robot was able to insert 154 parts successfully (96.25%). In four cases the assembly failed because the vision system was not able to detect the part or hole reliably (lost tracking during motion three times, once the part was not found initially). In the remaining two cases, visual servoing was completed successfully, but the insertion failed because the robot missed the ROA. No error recovery was implemented for these experiments, in case of errors (like lost tracking) the current part was dropped and the sequence continued with the next part.

C. Comparison with humans

The most challenging test for any robot system is the comparison with human performance. As a measurable benchmark, we chose to compare the total time needed for the whole task of inserting all the pieces. The untrained participants were given the assignment to insert the 8 pieces as fast as possible with one hand. During the execution, the overall time as well as the time needed for every piece was recorded. Altogether, 41 persons were tested, whereof 35 were children of age 5–7 and the rest were adults.

	Robot	Adults	Children
avg.	132s	39s	94s
min.	130s	22s	40s
max.	138s	53s	220s

TABLE I: Comparison of average, minimum and maximum times for the complete sequence.

Tab. I compares the total time for the complete sequence. Adults needed roughly 30% of the robot's time for the task, while children needed about 70%. The variation of the robot performance was quite low, since the only nondeterministic part of the strategy was the searching for the pieces on the table. Humans, instead, varied their strategy, trying first to solve the problem as fast as possible, and then refined the strategy in subsequent attempts in case of failure. Some children needed considerably longer than the robot and were able to fulfil the task only with additional hints for recognizing and assembling the star shaped parts.

Fig. 11 compares the times needed for the individual pieces and shows that humans, especially children, had big

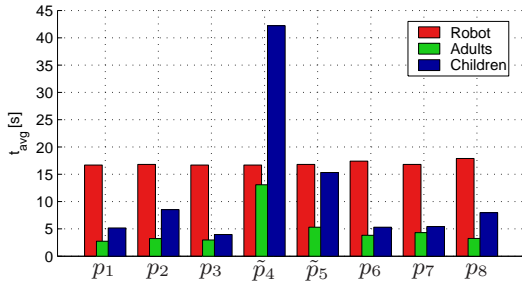


Fig. 11: Average times needed for the individual parts. Whereas the robot shows similar performance for all the parts, humans have difficulties especially with the differentiation and insertion of the star shapes. In this figure, \tilde{p}_4 represents the star that is inserted first (can be p_4 or p_5), \tilde{p}_5 the other one. The octagon p_2 and the KUKA logo p_8 are parts that jam easily which makes them difficult to insert for children.

problems with the stars. The similarity of the parts and the difficulty of the insertion confused not only children, but also adults regarding the right correspondence. For simple parts, however, the robot was not able to compete with the superior performance of the humans. Three main differences could be identified and should be addressed in further research:

- humans have an eye-to-hand configuration and can therefore combine vision with fast movements of the hand. Searching for the parts and the holes is done much faster, as can be seen in the attached video;
- arm (and fingers⁵) of humans are still superior in terms of speed and dexterous manipulation;
- intelligent error handling and refinement of strategies lead to performance optimization.

VII. CONCLUSIONS

In this paper we presented an integrated approach which combines vision, impedance control and robust planning for autonomous robotic assembly of complex planar parts. The overall task is addressed by different points of view, ranging from shape recognition, visual servoing, and active compliance control for robust insertion of the parts. Effectiveness and performance of the proposed approach are assessed through extensive experimental testing on the torque-controlled DLR light-weight robot. Live demonstration of this setup was also presented during the ICRA'07 conference in Rome as well as in the last Automatica'08 fair.

Concerning future developments, we are exploring the possibility to use force and vision feedback altogether during the insertion phase, so as to obtain a combined visual-compliance controlled strategy that would significantly increase robustness and performance of the assembly task. The practical goal, for potential industrial applications, is to obtain a robust visual software which requires tuning of few parameters in order to adapt to new parts and illumination conditions. Moreover, starting from CAD-data of the parts, the emerging robust assembly toolbox will automatically generate the controller parameters and desired robot motions.

⁵The three seconds needed for one part by the most skilled humans correspond roughly to the time needed by the used industrial gripper to open and close!

VIII. ACKNOWLEDGEMENTS

We thank the children from class 1c of Ostendschule Neuburg a. d. Donau and the Regenbogen kindergarten Pasing for participation in our experiments.

REFERENCES

- [1] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," *IEEE Trans. on Robotics and Automation*, vol. 12, no. 5, pp. 651–670, 1996.
- [2] K. Arbter, G. Hirzinger, J. Langwald, G. Wei, and P. Wunsch, *Proven Techniques for Robust Visual Servo Control*. IEEE Press, The Institute of Electrical and Electronics Engineers, Inc., New York, 2000, pp. 109 – 125.
- [3] F. Chaumette and S. Hutchinson, "Visual servo control. I. Basic approaches," *IEEE Robotics & Automation Mag.*, vol. 13, no. 4, pp. 82–90, 2006.
- [4] —, "Visual servo control. II. Advanced approaches," *IEEE Robotics & Automation Mag.*, vol. 14, no. 1, pp. 109–118, 2006.
- [5] D. E. Whitney and J. L. Nevins, "What is the remote center compliance and what can it do?" *Proc. 9th Int. Symp. on Industrial Robots*, 1978.
- [6] N. Hogan, "Impedance control: An approach to manipulation, part I - theory, part II - implementation, part III - applications," *J. of Dynamic Systems, Measurement and Control*, vol. 107, pp. 1–24, 1985.
- [7] A. Albu-Schäffer, Ch. Ott, and G. Hirzinger, "A unified passivity based control framework for position, torque and impedance control of flexible joint robots," *Int. J. of Robotics Research*, vol. 26, no. 1, pp. 23–39, 2007.
- [8] W. S. Newman, M. S. Branicky, H. A. Podguski, S. Chhatpar, L. Huang, J. Swaminathan, and H. Zhang, "Force-responsive robotic assembly of transmission components," *Proc. 1999 IEEE Int. Conf. of Robotics and Automation*, pp. 2096–2102, 1999.
- [9] H. Bruyninckx, S. Dutré, and J. D. Schutter, "Peg-on-hole: a model based solution to peg and hole alignment," *Proc. 1995 IEEE Int. Conf. of Robotics and Automation*, pp. 1919–1924, 1995.
- [10] J. Xiao, "Goal-contact relaxation graphs for contact-based fine motion planning," *Proc. 1997 IEEE Int. Symp. on Assembly and Task Planning*, pp. 25–30, 1997.
- [11] T. Lozano-Pérez, M. T. Mason, and R. H. Taylor, "Automatic synthesis of fine-motion strategies for robots," *The Int. J. of Robotics Research*, vol. 3, no. 1, pp. 3–24, 1984.
- [12] A. Stemmer, A. Albu-Schäffer, and G. Hirzinger, "An analytical method for the planning of robust assembly tasks of complex shaped planar parts," *Proc. 2007 IEEE Int. Conf. of Robotics and Automation*, pp. 317–323, 2007.
- [13] K. Arbter, "Affine-invariant Fourier descriptors," in *From Pixels to Feature*, J. C. Simon, Ed. Elsevier Science Publishers B.V., 1989.
- [14] —, "Affinvariante Fourierdeskriptoren ebener Kurven," Dissertation, Techn. Univ. Hamburg-Harburg, 1990.
- [15] R. O. Duda and P. H. Hart, *Pattern Classification and Scene Analysis*. John Wiley & Sons, Inc., 1973.
- [16] M. Hu, "Visual pattern recognition by moment invariants," *IRE Trans. on Information Theory*, vol. 8, no. 2, pp. 179–187, 1962.
- [17] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Trans. on Robotics and Automation*, vol. 8, no. 3, pp. 313–326, 1992.
- [18] F. Chaumette, "Image moments: A general and useful set of features for visual servoing," *IEEE Trans. on Robotics and Automation*, vol. 20, no. 4, pp. 713–723, 2004.
- [19] L. Sciavicco and B. Siciliano, *Modelling and Control of Robot Manipulators*. Springer, 2000.