

$$\begin{aligned}
a_{12} &= -{}^wZ \sin \theta - {}^wX \cos \theta, \\
a_{22} &= \frac{r}{w_H} \left(1 - \frac{{}^wY}{w_H}\right), \\
a_{31} &= \frac{r}{f} (J_{31} \cos \theta + J_{33} \sin \theta), \\
a_{32} &= -J_{31} \sin \theta + J_{32} + J_{33} \cos \theta, \\
a_{33} &= \frac{w_H}{w_J},
\end{aligned}$$

where J_{31} , J_{32} and J_{33} are components of \mathbf{J}_{pers} and

$$\begin{aligned}
w_H &= -{}^wX \sin \theta + {}^wZ \cos \theta + r, \\
w_J &= ({}^w\alpha \sin \theta - {}^w\beta \cos \theta) {}^wY + w_H.
\end{aligned}$$

Consider now the geometric interpretation of w_H and w_J : w_H is the cZ coordinate of the point P , which means that it is positive when the moving rod is in front of the camera. w_J is related to the intersection P_J (see Fig. 12) of PP_0 and the ${}^cX{}^cZ$ plane, and is just the cZ coordinate of this point.

The image error \mathbf{E} exponentially decay to $\mathbf{0}$, if the eigenvalues a_{11} , a_{22} and a_{33} of the matrix $\mathbf{J}_{\text{pers}}\mathbf{J}_{\text{weak}}^-$ are all positive. That is, The error converges if $w_H > 0$, $w_H > {}^wY$ and $w_J > 0$. Thus, in the common situation when the rod stays in front of the camera and is neither very high nor too tilted, all the eigen values of $\mathbf{J}_{\text{pers}}\mathbf{J}_{\text{weak}}^-$ are positive and \mathbf{E} converges to $\mathbf{0}$. The two first condition are almost always satisfied as the viewing angle of the camera is commonly less than 90 degrees. Note that at the final stage of the convergence, when the moving rod is close to the stationary rod (${}^wX, {}^wY, {}^wZ \simeq 0$) (meaning $w_H \approx r$, ${}^wY \approx 0$) and almost aligns with it (meaning $w_J \approx r$) all the eigen values become 1 and the components of \mathbf{E} decay uniformly to $\mathbf{0}$.

Thus we conclude that the control law assuming weak perspective approximation proposed in our paper ensures uniform decay of the error to $\mathbf{0}$ even when the imaging situation behaves according to the perspective projection model.

P is perspectively projected to the image point

$${}^i\mathbf{p} = ({}^ix, {}^iy)^T = \left(f \frac{{}^cX}{cZ}, f \frac{{}^cY}{cZ}\right)^T,$$

where f is the focal length of the camera. The slope of the rod ${}^i\alpha$ is obtained from the projected endpoints of the rods by (7). The image Jacobian, associated with the perspective projection, may be calculated as

$$\mathbf{J}_{\text{pers}} = (J_{ij}) = \frac{\partial({}^ix, {}^iy, {}^i\alpha)}{\partial({}^wX, {}^wY, {}^wZ, {}^w\alpha, {}^w\beta)} \quad (12)$$

(see [21] for the full derivation). It turns out, as expected that for $r \gg {}^wX, {}^wY, {}^wZ$, $\mathbf{J}_{\text{pers}} = \mathbf{J}_{\text{weak}}$, where \mathbf{J}_{weak} is the Jacobian calculated in (8), based on the weak perspective assumption. Yet, we show now that the proposed control law (9), which was based on the simplified weak perspective imaging model, gives the desired results even when the approximation is rough.

The image error is now modeled by:

$$\dot{\mathbf{E}} = \mathbf{J}_{\text{pers}} \dot{\mathbf{P}}. \quad (13)$$

Applying the control law (9), which is based on the weak perspective assumption on a system behaving according to the more realistic model (13), gives an error \mathbf{E} , which, locally, behaves as

$$\dot{\mathbf{E}} = -\lambda \mathbf{J}_{\text{pers}} \mathbf{J}_{\text{weak}}^{-1} \mathbf{E}. \quad (14)$$

This error behavior depends on the matrix

$$\mathbf{J}_{\text{pers}} \mathbf{J}_{\text{weak}}^{-1} = \begin{pmatrix} a_{11} & a_{12} & 0 \\ 0 & a_{22} & 0 \\ a_{31} & a_{32} & a_{33} \end{pmatrix}, \quad (15)$$

the components of which are

$$a_{11} = \frac{r}{w_H},$$

Appendix A: A control structure for the full perspective case

The aim of this appendix is to briefly discuss the control task considered above in the more general perspective projection model, and to show that the use of the weak perspective approximation is indeed sufficient and justified. The full detailed derivation is described in [21]. We start by deriving the Jacobian matrix under this more general imaging model.

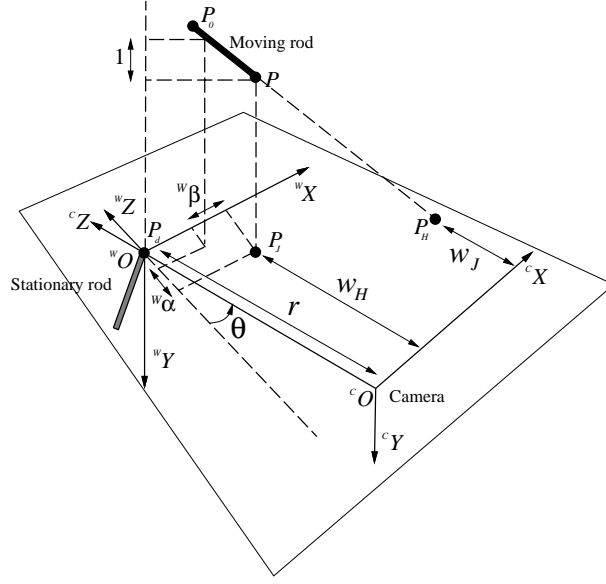


Figure 12: Point P_H is a projection of P onto ${}^wX{}^wZ$ plane. Point P_J is intersection of P_0P and ${}^wX{}^wZ$ plane. w_H and w_J correspond to the cZ coordinates of point P_H and P_J in the camera coordinate system. If they are both positive, our proposed control law obtained from weak perspective approximation will converge even in the perspective case.

Let the lower and upper endpoint of the moving rod be P and P_0 (Fig. 12). The upper endpoint of the stationary rod P_d is located at wO in our case. The coordinates of P in world and camera coordinate systems are,

$${}^w\mathbf{P} = ({}^wX, {}^wY, {}^wZ)^T, \quad (10)$$

$${}^c\mathbf{P} = ({}^wX \cos \theta + {}^wZ \sin \theta, {}^wY, -{}^wX \sin \theta + {}^wZ \cos \theta + r)^T. \quad (11)$$

- [11] Bradley J.Nelson and Pradeep K.Khosla. The resolvability ellipsoid for visual servoing. In *CVPR*, pages 829–832, 1994.
- [12] Y. Censor. Row action methods for huge and sparse systems and their applications. *Siam Review*, 23(4):444–466, 1981.
- [13] L.G. Gubin, B.T. Polyak, and E.V. Raik. The method of projections for finding the common points of convex sets. *USSR Computational Math. and Math, Physics*, 7(6):1–24, 1967.
- [14] Billibon H.Yoshimi and Peter K.Allen. Alignment using an uncalibrated camera system. *IEEE Trans. Robotics and Automation*, (4):516–521, 1995.
- [15] Seth Hutchinson, Gregory D.Hager, and Peter I.Corke. A tutorial on visual servo control. *IEEE Trans. Robotics and Automation*, (5):651–670, 1996.
- [16] L.E. Weiss, A.C. Sanderson, and C.P. Neuman. Dynamic sensor-based control of robots with visual feedback. *IEEE Journal Robotics and Automation*, 3(5):404–417, 1987.
- [17] P.I. Corke. Visual control of robot manipulators – a review. In Koichi Hashimoto, editor, *Visual Servoing*. World Scientific, 1993.
- [18] G. Strang. *Linear Algebra and Its Applications*. Academic Press, 1976.
- [19] K.Hashimoto and H.Kimura. LQ optimal and nonlinear approaches to visual servoing. In Koichi Hashimoto, editor, *Visual Servoing*. World Scientific, 1993.
- [20] P.L. Combettes. The foundations of set theoretic estimation. *Proceedings of the IEEE*, 81(2):182–208, 1993.
- [21] K. Kinoshita and M. Lindenbaum. Robotic control with partial visual information. Technical Report TR-H-215, ATR Human Information Processing Labs., 1997.

References

- [1] Keisuke Kinoshita and Michael Lindenbaum. Robotic control with partial visual information. *ICCV*, pages 883–888, 1998.
- [2] E.D. Dickmanns and V. Graefe. Dynamic monocular machine vision. *Machine Vision and Applications*, 1:223–240, 1988.
- [3] E.D. Dickmanns and V. Graefe. Applications of dynamic monocular machine vision. *Machine Vision and Applications*, 1:241–261, 1988.
- [4] Y. Shirai and H. Inoue. Guiding a robot by visual feedback in assembling tasks. *Pattern Recognition*, 5:99–108, 1973.
- [5] Koichi Hashimoto (editor). *Visual Servoing*. World Scientific, 1993.
- [6] B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. *IEEE Trans. Robotics and Automation*, 8(3):313–326, 1992.
- [7] G.D. Hager. A modular system for robust positioning using feedback from stereo vision. *IEEE Trans. Robotics and Automation*, 13(4):582–595, 1997.
- [8] N.P.Papanikolopoulos, P.K.Khosla, and T.Kanade. Visual tracking of a moving target by a camera mounted on a robot: A combination of control and vision. *IEEE Trans. Robotics and Automation*, (1):14–35, 1993.
- [9] Peter K.Allen, Aleksandar Timcenko, Billibon Yoshimi, and Paul Michelman. Automated tracking and grasping of a moving object with a robotic hand-eye system. *IEEE Trans. Robotics and Automation*, (2):152–165, 1993.
- [10] Kevin Nickels and Seth Hutchinson. Characterizing the uncertainties in point feature motion for model-based object tracking. In *Workshop on New Trends in Image-Based Robot Servoing*, pages 53–63, 1997.

task. We characterize the conditions required for a guaranteed convergence of such control tasks using tools from optimization.

The second issue we consider is the design of control algorithms for a particular task belonging to the class of the above characterization, where a moving camera provides the visual feedback. The successful convergence was proved using the tools developed in the first part. The Jacobian based control rule was optimized so that the task is carried out in a relatively small number of steps.

The control scheme proposed here are relatively slow. Therefore, in practice, combining it with other methods may give better performance. We already mentioned that combining our method with stereo based control leads to enhanced robustness to occlusion: In the event that one camera is occluded, the proposed method can partially close the loop, and can prevent excessive error due to drift or motion. Another useful combination is in the context of one moving camera: use several image and a reconstruction based method for roughly correcting a large error and then, use the proposed scheme for the finer positioning. Such combined schemes as well as the design of optimal camera motion paths are currently under investigation.

Acknowledgements

The authors wish to thank Dr. Shigeru Akamatsu for his useful suggestions and comments. We also thank the anonymous reviewers of IJCV for their helpful advice.

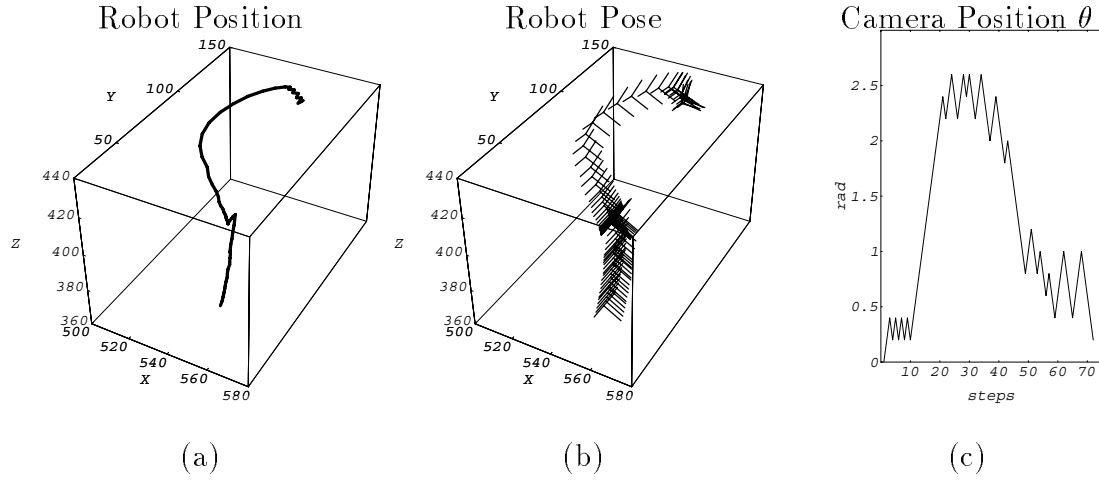


Figure 10: (a) Trajectory of the robot control point in robot coordinate system. The robot moved from the back to the front and then in the vertical direction. (b) A more detailed description of the robot trajectory. The plotted coordinate system provides the orientation of the robot end effector too. (c) The camera pose θ , changing by constant step, in a direction determined by the time sharing strategy (Every 10 steps, the criterion for ideal camera pose, which determines the camera direction, alternates between two options (see text)).

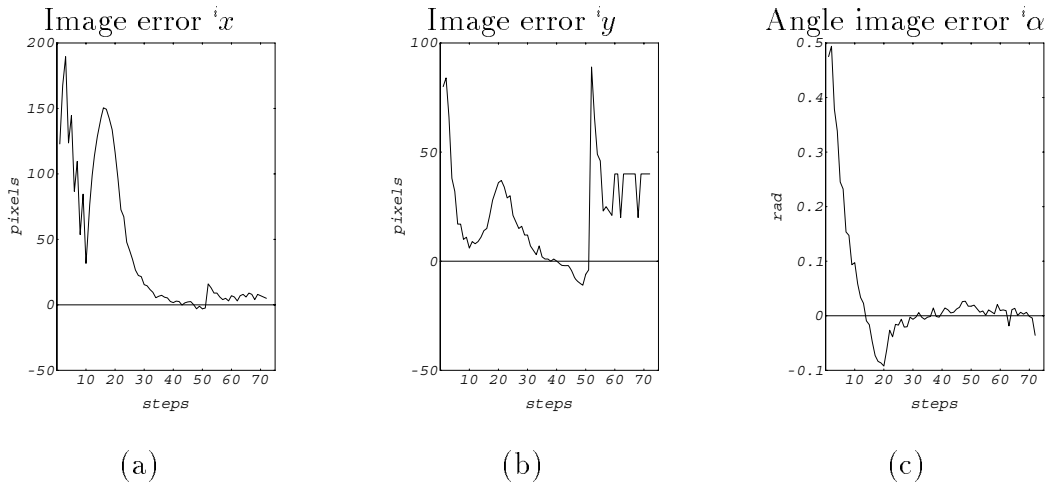
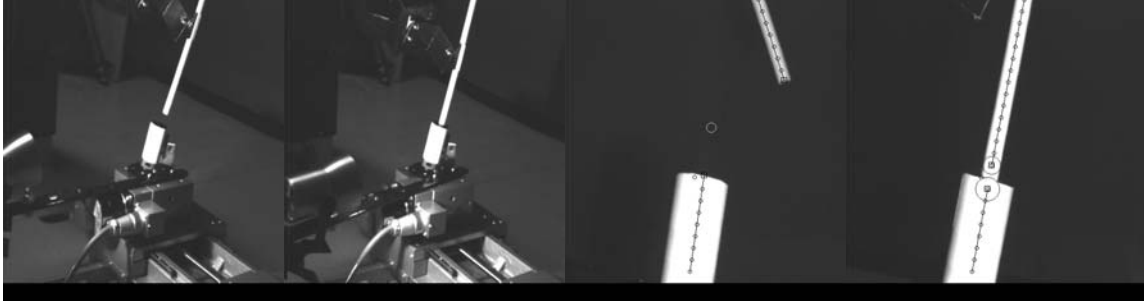


Figure 11: A plot of the three image errors. (a) Initially, the horizontal image error $^i x$ does not decay smoothly, but starting from step 20, the error decay exponentially to 0. (b) The jump observed at step 52 of the vertical image error $^i y$ corresponds to the change in the goal point indicating the beginning of the insertion process. (c) The plot of angular error $^i \alpha$, between the moving rod and the stationary rod demonstrates that the error converges to zero with one degree accuracy or better.

To conclude, the proposed control scheme is able to carry the alignment and insertion tasks successfully. The task is non-trivial and implicit 3D information was needed to carry it. Yet, one camera was sufficient to provide this information and to close the loop successfully.



(a) (b) (c) (d)

Figure 9: (a) An initial image of the two rods as seen from a side camera (which is not the rotating camera providing the feedback). (b) An image of the two rods after the insertion process (as seen from a side camera). (c) An initial image of the two rods as seen by the rotating camera. The two segments are extracted by a simple process which relies only on 1/20 of the lines, as explained in the text. A imaginary goal is placed vertically 80 pixels above the upper endpoint of the stationary rod. During alignment stage, the robot tries to move the lower end of the moving rod to this imaginary goal. (d) A final image of the insertion process as seen by the rotating camera.

6 Conclusions

This paper considers control tasks which rely on visual feedback. Such control tasks can be characterized by a Jacobian matrix relating the observation to the pose, locally. Unlike previous work, we consider the case where this Jacobian matrix is of non-full rank.

The paper focuses on two issues. The first is the ability to succeed in such control tasks. We show that if the associated Jacobian changes in time, then the partial visual information may be sufficient as feedback, and leads to a successful

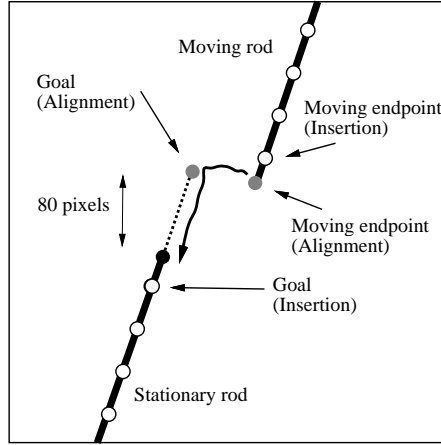


Figure 8: The intersection of the rods image and the horizontal line of every 20 pixels are detected and called crossing points. Connecting the crossing points, the rods are extracted. The endpoints as well as the orientation of the rods are used for the task. The goal point and the moving endpoint are switched to the endpoints of the rods or to the crossing points depending on the two stages, alignment and insertion.

also the orientation of the robot hand, is given in Fig. 10(b), where the coordinate axes represents the orientation of the robot.

The camera pose θ , as implied from the adopted time sharing strategy, is shown in Fig. 10(c). Two θ values corresponding to the ideal poses for best angle convergence and best endpoint position convergence were calculated, and alternatively (every 10 steps) served as targets to the camera pose. Note that the camera pose changes by constant steps of size $\Delta\theta = 0.2(\text{rad})$. Only the direction of changes is determined from the changing moving rod pose.

The horizontal image error ${}^i x$ is shown in Fig. 11(a). In the beginning (first 20 steps), the error does not decay smoothly because of the non-linearity of the system and the calibration error. Later, however, the error decays exponentially to 0, justifying the approximation. The vertical image error ${}^i y$ decreases monotonically as shown in Fig. 11(b) after 20 steps. At step 52, both ${}^i x$ and ${}^i y$ “jump”, indicating the beginning of the insertion process. The angular error ${}^i \alpha$ smoothly converges to 0, as shown in Fig. 11(c).

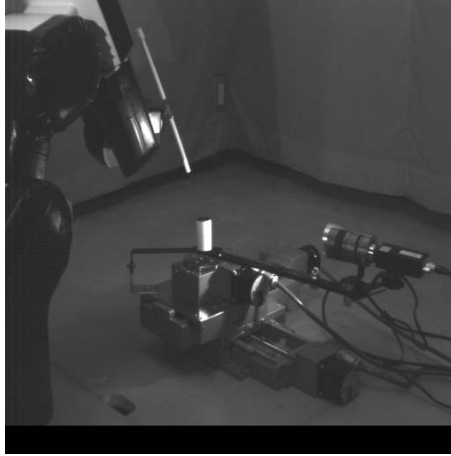


Figure 7: The full experimental setup. A rod is fixed to a robot with a paper clip. A hollow rod is fixed near the center of a rotating table. A camera, together with a lighting source, is fixed to the rotating table with a metal bar support. The robot tries to insert the rod into the stationary hollow rod and the camera takes a suitable pose to accomplish the task.

stage starts. The goal point is lowered to the uppermost crossing point of the stationary rod, and the moving endpoint is raised to the lower-most crossing point of the moving rod. (A “jump” occurs in the horizontal and vertical image error (${}^i x, {}^i y$)). Note that, during the insertion stage, the error in the ${}^i y$ coordinate does not converge to zero but stays constant (and equal to the vertical distance between the two crossing points). Consequently, the same algorithm used in the first stage will now continue to push the moving rod into the stationary rod, without any further change.

Typical results, associated with the more difficult situation involving a heavily tilted stationary rod, are now described. The initial and the final poses of the two rods in the first experiment are shown in Fig. 9(a) and (b), respectively. The corresponding images, which were used as feedback are given in Fig. 9(c) and (d), with the extracted segments and the imaginary goals superimposed.

The trajectory of the robot in robot base coordinate system is shown in Fig. 10(a). Note that the robot moved from the back to the front, and then moved downward for the insertion. A richer description of the path, including

this scheme and to follow a “time sharing” procedure in which the camera position is shifted between being optimized for one subtask and being optimized for the other subtask every several dozens of moves. Note that all the pose parameters may be controlled all the time, even when the camera is not positioned optimally for controlling them. In these moves, however, their convergence is relatively slow. We also observed that choosing camera position which is in the middle between the optimal positions corresponding to the two subtasks, is bad for both subtasks: both processes converge very slowly. Therefore, the “time sharing” strategy was adopted in our experiments.

5 Experimenting in real robotic environment

To test the control strategy proposed in the former section, experiments were conducted. A six DOF robot, a rotating table and a camera were used to build the system described in Fig. 7. The task is, as mentioned earlier, to insert the rod to the stationary hollow rod. No calibration was done except a rough measurement of some distances such as the radius of rotating circle of the camera (about 350mm) and the distance between the robot base and the center of rotating table (about 650mm).

The only measurements extracted from the images were the relevant end points and some *crossing points*, specified as the intersection between the rod images and some horizontal lines (spaced every 20 pixels) (Fig. 8). The location and slope of the rods were inferred from this partial information.

The task is separated into two stages, *alignment* and *insertion*. The same process is used for both stages, and switching between them is done by only different specification of the end points and the target: During the alignment stage, the goal point is specified slightly above the uppermost endpoint of the stationary rod (on the rod axis), while the moving rod endpoint is the lowest endpoint of the moving rod (Fig. 8). After the rods are aligned, the insertion

rotations around the visual axis, and they lead to convergence similarly, as if the ${}^w\alpha$ and ${}^w\beta$ parameters, which describe the angle, were coordinates of a point on a plane. The three processes are independent and the recommendations for choosing λ , described above, apply also here. Convergence is also guaranteed as before.

4.2.3 A camera motion strategy

The camera pose should be set so that the error, observed by the camera, is useful in correcting the 3D discrepancy. This was simple in the 2D case, where only one direction (pose subspace of dimension one) was effective. Now, however, one camera pose may be effective for closing the gap between the endpoints and another camera pose, which may be completely different, may be effective for reducing the difference between the (3D) slope of the rods. Consider, for example, a moving rod lying on the ${}^wX = 10$ plane, with its lower endpoint in $(10, 0, 0)^T$ and tilted so that it coincides with the ${}^wY = {}^wZ$ line on this plane. There, the length ratio parameters are $\alpha = 0$ and $\beta = 1$. For moving the endpoint, it is best to set the camera so that its visual axis coincides with the wZ direction, but in this position, the slope of the rod is not changed at all. For changing the slope, it is best to set the camera so that its visual axis coincides with the wX direction, but in this position, the endpoint position does not change.

Therefore, the camera position cannot be optimal for regulating both the endpoint pose and the rod angle. One straightforward solution is to start with a subset of the pose parameters (e.g. wX , wY and wZ), control it so that it reach the desired value with the camera position optimized for this task. Then, switch to the other subset (e.g. ${}^w\alpha$, ${}^w\beta$), and control it with the camera position changed so that it matches the second subtask. This method is problematic because controlling the different pose parameters is coupled, due to the inaccuracy in the modeling of the manipulating robotic arm. Therefore we have chosen to modify

The image error \mathbf{E} is related to the pose error \mathbf{P} by the Jacobian.

$$\mathbf{E} = \begin{pmatrix} {}^i x \\ {}^i y \\ {}^i \alpha - {}^i \alpha_d \end{pmatrix} = \mathbf{J} \mathbf{P} = \begin{pmatrix} \frac{f}{r} \cos \theta & 0 & \frac{f}{r} \sin \theta & 0 & 0 \\ 0 & \frac{f}{r} & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos \theta & \sin \theta \end{pmatrix} \begin{pmatrix} {}^w X \\ {}^w Y \\ {}^w Z \\ {}^w \alpha - {}^w \alpha_d \\ {}^w \beta - {}^w \beta_d \end{pmatrix} \quad (8)$$

(Trivially, $\dot{\mathbf{E}} = \mathbf{J} \dot{\mathbf{P}}$.) Note that the 5D pose space contains a 2D subspace for which all points in this subspace have the same image and are therefore indistinguishable.

4.2.2 A Jacobian based control algorithm

The Jacobian based controller is, trivially,

$$\Delta \mathbf{P} = -\lambda \mathbf{J}^- \mathbf{E} = -\lambda \begin{pmatrix} \frac{r}{f} \cos \theta & -\frac{r}{f} \sin \theta & 0 \\ 0 & \frac{r}{f} & 0 \\ \frac{r}{f} \sin \theta & \frac{r}{f} \cos \theta & 0 \\ 0 & 0 & \cos \theta \\ 0 & 0 & \sin \theta \end{pmatrix} \begin{pmatrix} {}^i x \\ {}^i y \\ {}^i \alpha - {}^i \alpha_d \end{pmatrix} \quad (9)$$

Note that the vectors spanning the subspace of indistinguishable poses are orthogonal to the columns of the \mathbf{J}^- matrix, implying that the motion is perpendicular to this subspace. Note also that if θ changes then the columns of the two different \mathbf{J}^- matrices span the whole space.

The intuitive meaning of the control steps is as follows: Every step moves the $({}^w X, {}^w Z)$ coordinates of the lower endpoint of the moving rod in the direction perpendicular to the visual axis ${}^c Z$ and eventually this endpoint converges to the upper endpoint of the stationary rod exactly in the same way that the point object converged to its destination in the 2D control task described above.

The ${}^w Y$ coordinate converges even smoother and faster because full information is available on this coordinate from the image. The rod angle steps are

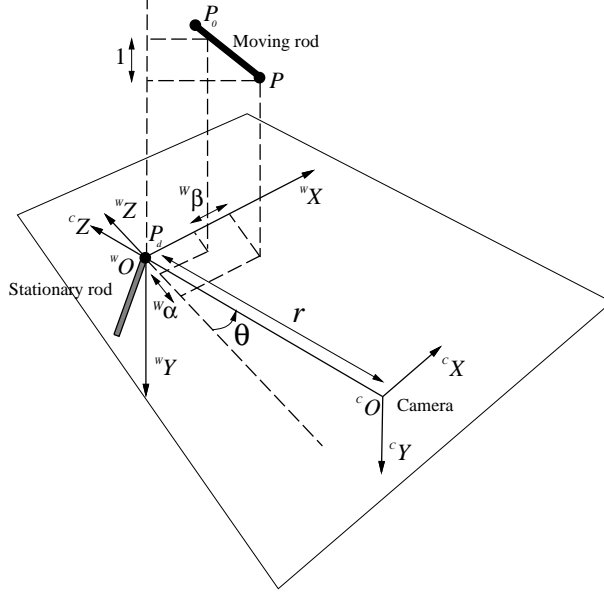


Figure 6: The moving rod is represented with $P_0\vec{P}$. Its orientation is $\alpha = ({}^w\alpha, 1, {}^w\beta)^T$. The stationary rod is P_d . The Camera cO rotates around P_d with radius r .

normalized by the projection on the wY axis, to specify the orientation of the rod (Fig. 6). Similarly, the stationary rod is characterized by its upper endpoint coordinates, specified as the origin, $({}^wX_d, {}^wY_d, {}^wZ_d)^T = (0, 0, 0)^T$ and by the ${}^w\alpha_d, {}^w\beta_d$ parameters.

The 3D endpoint $({}^wX, {}^wY, {}^wZ)^T$ is observed by its projection on the image, $({}^ix, {}^iy)^T$. The rod P_0P orientation is observed by the slope of its projection

$${}^i\alpha = \frac{{}^ix - {}^ix_0}{{}^iy - {}^iy_0}. \quad (7)$$

The slope of the stationary rod image projection ${}^i\alpha_d$ is similarly defined.

Let $\mathbf{P} = ({}^wX, {}^wY, {}^wZ, {}^w\alpha - {}^w\alpha_d, {}^w\beta - {}^w\beta_d)^T$, be the pose vector characterizing the moving rod. Note that every pose vector corresponds to a valid state and thus the space does not have any “holes”.

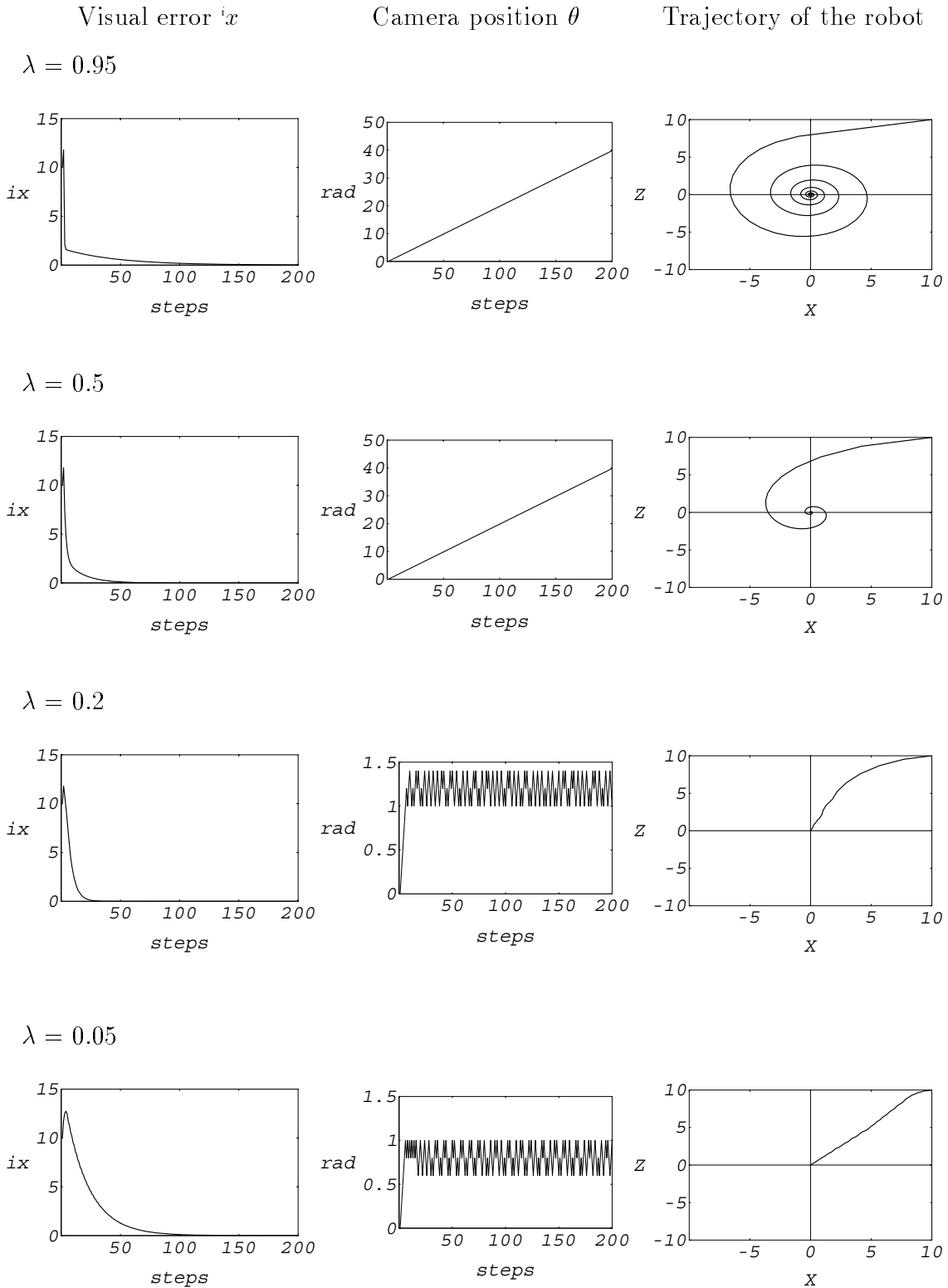


Figure 5: A simulation of the control system behavior: every row contains three plots. The left one is the convergence of the visual error $\|x$ as seen from the camera, the middle one is the camera position, and the rightmost one is the path of P from the starting point to the goal. The different rows correspond to different λ values, starting from $\lambda = 0.95$ (at the top) and continuing with $\lambda = 0.5, 0.2, 0.05$. Note the oscillatory behavior corresponding to high λ values and the relative slow convergence corresponding to extremal values.

4.1.5 Simulations

Simulating the control process according to the policies given above, and assuming that every step takes a constant time, confirms these predictions. Making small steps (a small λ value) leads to relatively small movement until the camera stabilizes in a position putting its visual axis perpendicular to the optimal motion direction. Convergence is slow however, due to the small steps. Making large steps (a large, close to 1, λ value) causes undesirable oscillatory behavior, and slow convergence, in spite of the large steps, due to the large distance traveled. The optimal step seems to be an intermediate value, which matches the optimal value prediction (see Fig. 5).

Note that these parameter choices is not in agreement with the usual visual servoing, associated with a stationary camera. There, in order to achieve a fast convergence, one tries to move the robot, so that the image error is maximally reduced. Here on the other hand, some fraction of the error should be maintained.

4.2 A full 3D rod alignment task

Now, after gaining some insight into the peculiarities of control with a moving camera, we turn to the realistic nontrivial task of inserting one rod into another.

4.2.1 The 3D setting

As described above, one rod is hollow and stationary and the other rod is held by the robot. The task is to control the robot so that the moving rod first aligns with the stationary one and eventually gets inside it. This requires to align the axes of the two rods, so that the lower point of the moving rod is on the 3D axis of the stationary rod and just above its upper endpoint, and then, to lower the moving rod while keeping it on this 3D line, so that it gets inside.

We shall use the world coordinates $({}^wX, {}^wY, {}^wZ)$ to specify the position of the endpoint of the rods, and the rod projections on these coordinates $({}^w\alpha, 1, {}^w\beta)$,

good step size (λ value): long steps (with λ value close to one) are good for maximally decreasing the distance between the controlled object and the desired final location at that particular step. On the other hand, such long steps bring the controlled point closer to the visual axis, and imply that the next step is less effective in reducing this distance.

To see how performance indeed depends on λ , we use two performance measures: the total traveled distance and the number of steps until convergence. We simulated for a sequence starting from ${}^w\mathbf{P}_1 = (7, 7)^T$ and reaching a distance of $\epsilon (= 0.01)$ (Fig. 4). Choosing a small λ value gives a small traveled distance but requires many steps. Choosing a large, close to 1, λ value causes both excessive number of steps and large travel distance. Choosing an intermediate λ value, anywhere from 0.2 to 0.4, seems best for the common robotic motion which makes any step in approximately the same time, and depends relatively little on the distance traveled.

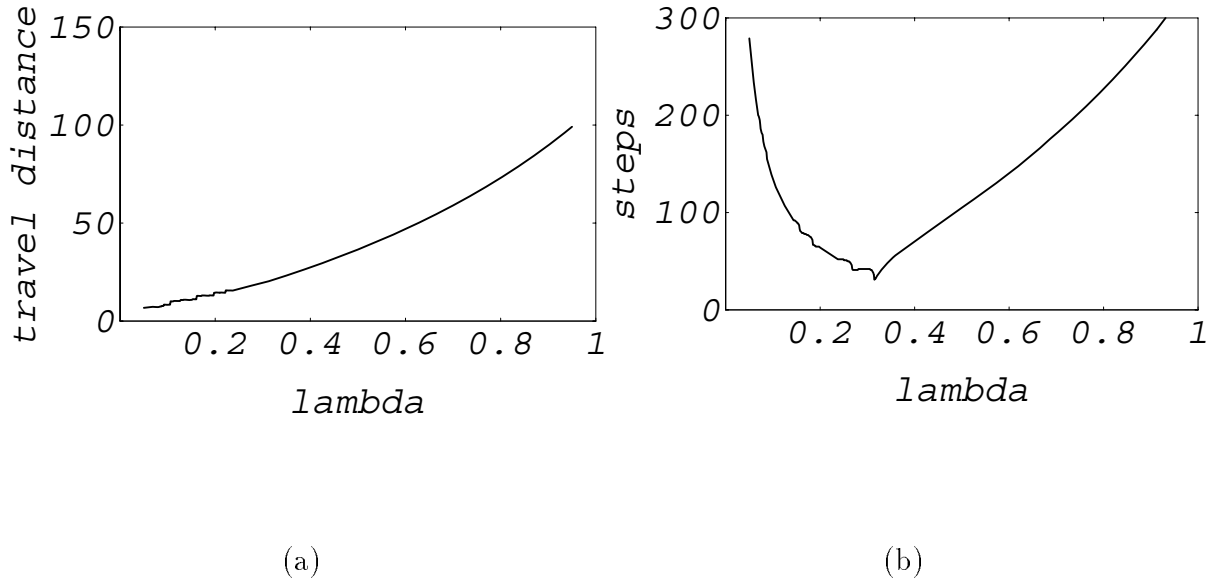


Figure 4: (a) The approximate travel distance and (b) the number of steps required for convergence. (${}^w\mathbf{P}_1 = (7, 7)$, $\Delta\theta = 0.2$).

every step and let $\Delta \mathbf{P}_{j-1} = (\Delta {}^{c'}X_{j-1}, 0)^T$ be the robot motion associated with the $(j-1)$ -th step (measured in the $(j-1)$ -th camera coordinate system). Then, the following equation relates the pose error vectors associated with the two steps.

$$\begin{pmatrix} {}^{c'}X_j \\ {}^{c'}Z_j \end{pmatrix} = \begin{pmatrix} \cos \Delta\theta & \sin \Delta\theta \\ -\sin \Delta\theta & \cos \Delta\theta \end{pmatrix} \cdot \left(\begin{pmatrix} {}^{c'}X_{j-1} \\ {}^{c'}Z_{j-1} \end{pmatrix} - \begin{pmatrix} \Delta {}^{c'}X_{j-1} \\ 0 \end{pmatrix} \right) \quad (5)$$

Inserting the estimate $\widehat{{}^{c'}X} = \frac{r}{f} {}^i x$ for both $\widehat{{}^{c'}X}_j$ and $\widehat{{}^{c'}X}_{j-1}$ in the first line of (5), we find $\widehat{{}^{c'}Z}_{j-1}$. Inserting it in the second line yields the estimate

$$\widehat{{}^{c'}Z}_j = \frac{1}{\sin \Delta\theta} \frac{r}{f} (\Delta {}^{c'}X_{j-1} - {}^i x_{j-1}) + \frac{\cos \Delta\theta}{\sin \Delta\theta} \frac{r}{f} {}^i x_j \quad (6)$$

Note that this pose estimate is used only for determining the camera movement and is not used for the control of the robot itself. Therefore we continue to benefit from the advantages of Jacobian based servoing as compared with reconstruction based approaches. The estimate improves when the camera step size increases. Yet, it does not need to be accurate, as we need only its sign. Note also that in places where this sign changes with small changes of pose, the ${}^{c'}Z$ component of the pose error \mathbf{P}_j is close to zero and therefore the camera position is nearly optimal anyway and its motion direction is not critical. Recall that the convergence is guaranteed anyway, even if the camera position is not what we asked for.

4.1.4 Some guidelines for choosing λ

Setting λ to 1 maximally decreases the distance between the current pose and the destination and thus seems to yield the fastest convergence. This intuition is also in agreement with previous image based control approaches, which try to make a move which maximally reduces the image error.

It turns out however, that when the camera moves, other considerations must be taken into account and that a λ value which is close to 1 leads to bad convergence properties. Two conflicting demands arise when coming to choose a

mated pose error vector lies, determines the preferred camera motion direction (see Fig. 3) and leads to the following strategy for increasing the angle between \mathbf{P}_j and the visual axis.

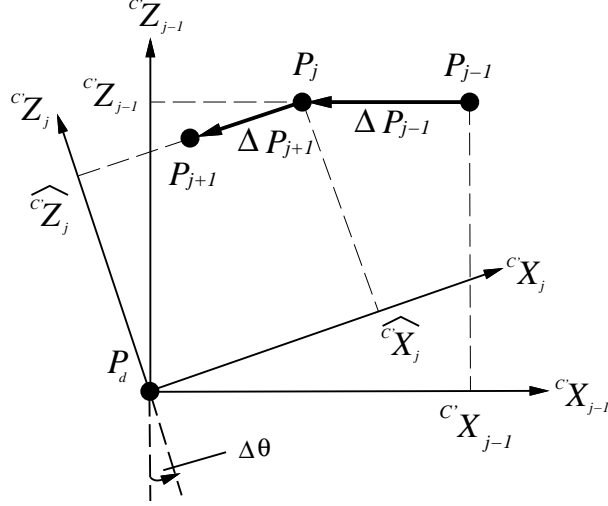


Figure 3: The camera motion direction is implied by the quadrant in which the estimated pose error vector lies. If, for example, $c^i \widehat{X}_{j-1} > 0$ and $c^i \widehat{Z}_{j-1} > 0$ then moving the camera counterclockwise ($\Delta\theta < 0$) decreases the angle between \mathbf{P}_j and the cX axis.

Camera Moving Policy : Let $\Delta\theta$ be some constant angle.

1. For the first step, arbitrarily move camera either by $+\Delta\theta$ or by $-\Delta\theta$.
2. At the j -th ($j > 1$) step
 - (a) Let $c^i \widehat{X}_j = \frac{r}{f} x$
 - (b) Estimate $c^i \widehat{Z}_j$ by (6) (below).
 - (c) Move camera with angular step of $\Delta\theta \cdot \text{Sign}(c^i \widehat{X}_j \cdot c^i \widehat{Z}_j)$.

The estimate $c^i \widehat{X}_j = \frac{r}{f} x$ follows directly from the definition of the coordinate systems above. The second component, $c^i \widehat{Z}_j$ of the pose vector is estimated from two consecutive images as follows: Let $\Delta\theta$ be the camera motion associated with

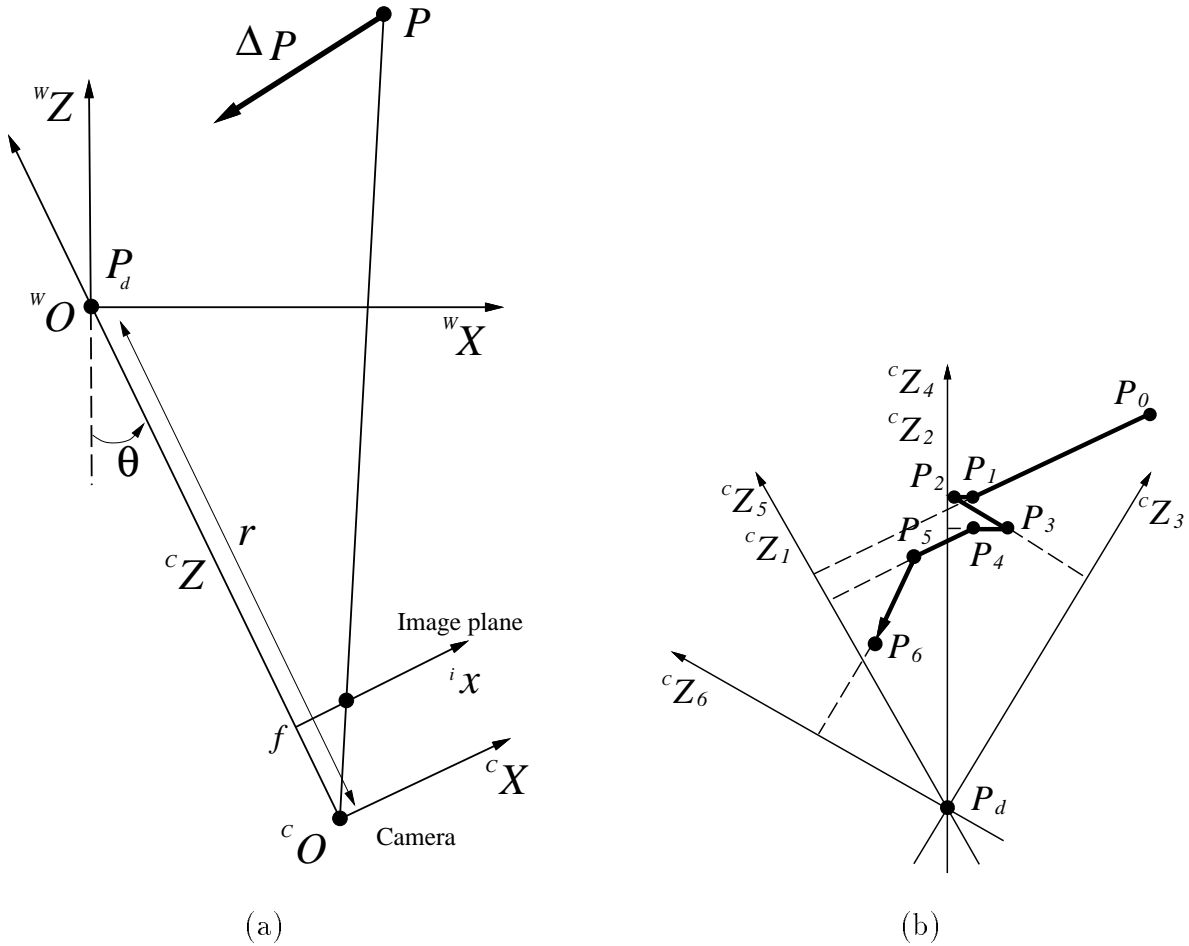


Figure 2: A diagram of the simplified 2D setting. The diagram (a) reveals the relation between the three coordinate systems. The path of the object depends on the sequence of camera positions and is illustrated in (b).

camera coordinate system. Therefore, intuitively, making the pose error vector $\Delta \mathbf{P}_j$ as parallel as possible to the ${}^c X$ axis (and perpendicular to the visual axis ${}^c Z$) should yield faster convergence. A simple camera moving strategy, which follows this intuition is now described.

It is more convenient to phrase the control rule in the coordinates $({}^c X, {}^c Y, {}^c Z)$ of an imaginary “shifted camera” which are parallel to those of the real camera but are centered at the target point P_d (and the world coordinate origin). The proposed strategy uses an estimate $\widehat{\mathbf{P}}_j = (\widehat{{}^c X}_j, \widehat{{}^c Z}_j)$ of the current error vector \mathbf{P}_j in the $({}^c X, {}^c Y, {}^c Z)$ coordinate system. The quadrant in which the esti-

4.1.1 The setting

We refer to three coordinate systems: world coordinate system $({}^wX, {}^wZ)$, camera coordinate system $({}^cX, {}^cZ)$ and image coordinate system ix . The goal P_d is located at $(0,0)$ in the world coordinate system. The camera rotates about a fixed axis close to the origin, but it does not need to coincide with it. The camera position is specified by the distance r from P_d and by the angle θ between the optical axis and the $-{}^wZ$ axis (Fig. 2(a)). The image of the observable target point serves both as the origin of the image coordinate system and as a reference for orienting the camera coordinate system.

4.1.2 A Jacobian-based control algorithm

The image error is 1D and is just the ix -coordinate value, $\mathbf{E} = {}^ix = \frac{f}{r} {}^cX$. The local description is

$$\dot{\mathbf{E}} = \mathbf{J}\dot{\mathbf{P}} = \frac{f}{r} (\cos \theta, \sin \theta) ({}^w\dot{X}, {}^w\dot{Z})^T, \quad (3)$$

where $\mathbf{J} = \frac{f}{r} (\cos \theta, \sin \theta)$ is the Jacobian. The Jacobian based controller (2), relying on this Jacobian, moves the robot by

$$\Delta \mathbf{P}_j = -\lambda \mathbf{J}^- \mathbf{E}_j = -\lambda \frac{r}{f} (\cos \theta_j, \sin \theta_j)^T \mathbf{E}_j \quad (4)$$

at the j -th step. (θ_j is the angle between the camera visual axis and the $-{}^wZ$ axis, in the time of taking the j -th image, before making the j -th move.) The intuitive meaning of this control scheme is simple: at every step the controller decreases the distance between the current pose and the desired pose in the only direction on which information is available: the cX axis (Fig. 2(b)).

4.1.3 A simple camera motion strategy

Let $\mathbf{P}_j - \mathbf{P}_d = \mathbf{P}_j$ be the current pose error of the controlled object. The Jacobian based control process corrects only the error component along the cX axis of the

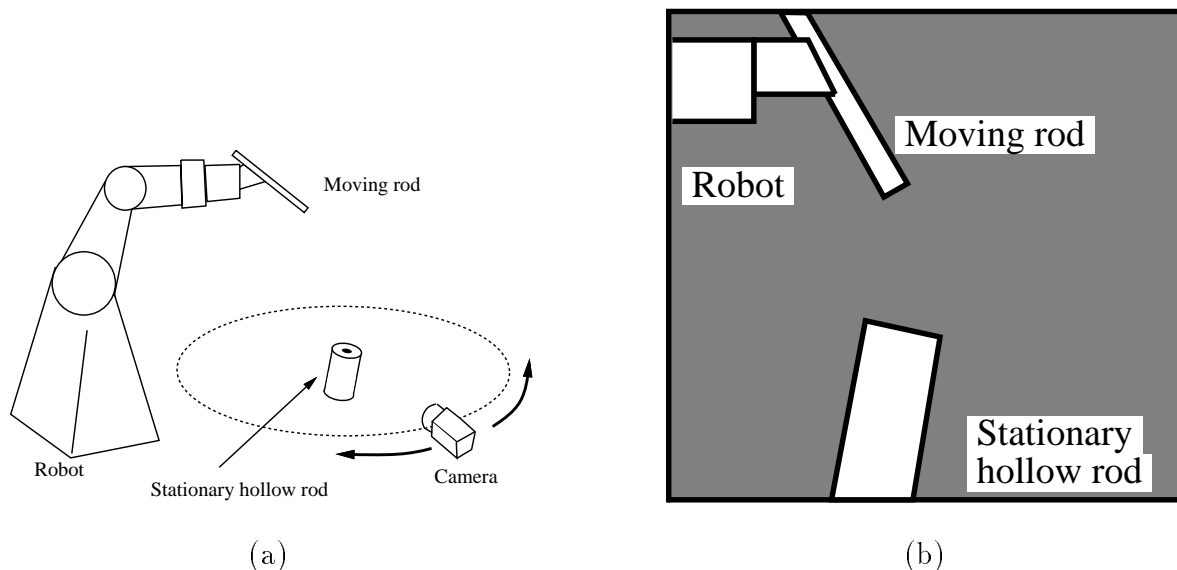


Figure 1: (a)A diagram of the 3D control setting and (b)a schematic image similar to those seen by the rotating camera. Note that only the two rods are observed.

4.1 A reduced dimension task

The simplified task we start with is controlling a “point object” on a plane so that it reaches some pre-specified destination point P_d from any initial position P_0 . The control algorithm uses only the information from a camera lying on the same plane and looking at the origin. All the motion space is a plane and it is mapped into one line in the image plane, which may be identified with the x axis of the camera coordinate system. The image of the “point object” is a point lying on this line. Therefore, only one scalar measurement, the difference in this x coordinate between the current object position and its destination, is available. The Jacobian based controller, relying on this feedback source, corresponds to a simple, intuitive control strategy, and, by the results of the previous section, results in successful convergence.

of the moving rod is a 3D space and 2D rotation. (The rotation of the moving rod on its own axis is both irrelevant for the insertion process and nearly unobservable in the image). A stereo camera pair, if available, seems to be the most adequate solution and will probably give better performance. Still, we show here that this task is feasible also with one camera and use this simple setting to explore the special properties of visual control processes that rely on a moving cameras.

A stereo-like approach, of using two consecutive image, is possible if the camera and the robot do not move together, and was used by Yoshimi and Allen [14]. There, they used the information from a camera mounted on a gripper, to insert a peg in hole, in a somewhat simplified scenario, where neither peg orientation nor the peg's distance from the hole were considered. Their approach is, in a sense, static, as it is based on collecting several images for every position of the peg, and inferring the required movement after every such round. Their work is classified to conventional feedback control with full-rank Jacobian. Our algorithm, on the other hand, uses only one image for every peg position, and gets only partial information from every such image. Thus, the Jacobian matrix becomes non-full rank. The integration of these pieces of information, via the combined peg and camera movements, yield a more dynamic approach.

We start with two simplifications. First, we considered a related simpler task, the analysis of which is more intuitive than the analysis of the full task, and the results may be fully illustrated by 2D curves. The control principles remain similar when moving to the more general case, which we discuss afterwards. A weak perspective imaging model is used for modeling and design in the paper, and is rigorously justified in appendix A, where the controller is derived using the exact perspective model.

of \mathbf{P}_1 in this coordinate system and show that their absolute value decreases at least in one of the first N pose changes. The absolute value of the components along the eigenvectors spanning \mathcal{P}^1 decreases to $|1 - \lambda|$ of their original value in the first step. Consider now a component of \mathbf{P}_1 along one of the other axes \mathbf{v} (\mathbf{v} is an eigenvector of the matrix $\mathbf{J}_1^- \mathbf{J}_1$ associated with null eigenvalues). Let \mathcal{P}_j^1 be the first subspace associated with an eigenvector \mathbf{e} which is not normal to \mathbf{v} . Such an eigenvector \mathbf{e} is associated with one of the subspaces in $\{\mathcal{P}_1^1, \mathcal{P}_2^1, \dots, \mathcal{P}_N^1\}$ because every vector in \mathcal{P} is spanned by the subspaces included in the subsequence. Then, at the j -th step, the size of this component decreases to $|1 - \lambda|\mathbf{v} \cdot \mathbf{e}|$ of its value or lower. \square

The theorem is generic and is not committed to a particular sequence of Jacobians and null spaces. The condition required is rather weak and does not put any constraint neither on the size N of the subsequences nor on the uniformity of the space covering by the subspaces. Note that for any set of camera poses, even for random poses, the theory guarantees the convergence as far as the set spans the full pose space. However, even the system converges, the convergence speed may be slow.

4 Robotic control using a single moving camera

After providing conditions for convergence in a very general setting, we now focus on a particular control scenario, which allows a limited motion of a single camera.

We shall use a 6 degrees of freedom robotic arm, to align a short cylindrical rod with another fixed, hollow cylindrical rod, and then to insert the first rod into the second. We shall use a “look and move” [4, 7] approach with Jacobian based controller. The visual information, used for the feedback is obtained from a camera moving on a circle in the $({}^wX, {}^wY)$ plane, with its optical axis roughly pointing towards the circle’s center $(0,0)$, where the stationary rod is located. (see Fig. 1) This particular control task requires 5 degree of motion: translation

particular, it is common to use relaxation and to update the temporary solution by changing it to some average between itself and its projection on the subspace. $\mathbf{x}_{j+1} = \mathbf{x}_j + \lambda(\text{Proj}_k(\mathbf{x}_j) - \mathbf{x}_j)$. Using λ values in $(0, 2)$ lead to correct convergence, and the exact value to use is a matter of taste and intuition.

We argue that the pose changes implied by the Jacobian based control scheme described above, may be considered to be relaxed projections of a subspace. The variables \mathbf{x}_j and $\text{Proj}_k(\mathbf{x}_j)$ correspond to the pose \mathbf{P}_j and to its projection on the subspace spanned by the eigenvectors of $\mathbf{J}^{-1}\mathbf{J}$ associated with null eigenvalues, at k -th step, respectively. Also, \mathbf{P}_d corresponds to the correct solution.

Every step of this control process decreases the distance between the pose and the desired pose \mathbf{P}_d , unless the pose is already in the null space \mathcal{P}_j^0 , associated with the Jacobian. If the Jacobian changes “enough” then, the corresponding null space also changes, the distance between the pose \mathbf{P}_j and the desired pose \mathbf{P}_d decreases, and eventually the pose converges to the desired pose. To formulate this condition more precisely, we shall say, with some abuse of notation, that a vector is spanned by a set of subspaces if it can be written as a linear combination of vectors from these subspaces. Then,

theorem 1 *Let $\mathbf{J}_1, \mathbf{J}_2, \dots$ be the sequence of Jacobians associated with the control process and let \mathcal{P}_j^1 $j = 1, 2, \dots$ be the corresponding pose subspaces in which the pose changes take place. Let an N -partition be a partition of these sequences into subsequences $\{\mathcal{P}_1^1, \mathcal{P}_2^1, \dots, \mathcal{P}_N^1\}$, $\{\mathcal{P}_{N+1}^1, \mathcal{P}_{N+2}^1, \dots, \mathcal{P}_{2N}^1\}, \dots$, each of length N . Suppose that there is an integer N such that every vector in the entire pose space \mathcal{P} is spanned by every subsequence of the N -partition. Furthermore, let $\lambda \in (0, 2)$. Then, the sequence of pose changes done according to Jacobian based controller (2) leads to convergence of the pose to the desired pose, $\mathbf{P}_n \rightarrow \mathbf{P}_d$.*

Proof: Consider, without loss of generality, the first subsequence and the coordinate system defined by the eigenvectors of the matrix $\mathbf{J}_1^{-1}\mathbf{J}_1$, associated with the first Jacobian and the first control step. We shall consider all the components

r -dimensional pose subspace, spanned by the $r = \text{rank}(\mathbf{J}^T \mathbf{J})$ eigenvectors of the $\mathbf{J}^T \mathbf{J}$ matrix, which are associated with non-zero eigenvalues. If, however, the Jacobian changes, then the corresponding subspace changes too, the sequence of pose changes is not limited to one subspace, and thus may reach any desired pose. A crucial question is whether such a control process indeed converges.

3.1 Conditions for convergence

To answer this question, we would like to start by observing the similarity between the control scenario and a method from numerical analysis: solving linear equations systems using row-action operations [12] which is a particular case of the known Projection On Convex Sets (POCS) method [20]. The task considered is the common task of finding a vector which satisfies a given set of linear constraints, or, in other terms, solving a (possibly over-constrained) system of linear equations. This task may be solved directly by using pseudo-inverse. The row-action solution is preferred when both the size of the sought for vector and the number of constraints are very large, so that a direct solution is not realistic and even the storage of all the constraints may be prohibitive [12].

The method is simple: every constraint specifies some subspace in the solution space and any point \mathbf{x} in the space is associated with a unique projection $\text{Proj}_k(\mathbf{x})$ on this k -th subspace. According to the row-action method, one starts with some arbitrary guess and, at every step, updates it by projecting it onto the hyper-plane specified by one of the constraints, $\mathbf{x}_{j+1} = \text{Proj}_k(\mathbf{x}_j)$ (k is a function of j). This sequential projections process provably converges to the correct solution provided that the constraints are consistent and that every constraint is used “many” times. The solution method has many variations related to the particular constraints involved (equalities and inequalities), the particular task (just satisfying the constraints or minimizing some cost function as well), the sequencing of the constraints, and the particular update done at every stage. In

is not of full rank, $r = \text{rank}(\mathbf{J}) < p$, and \mathbf{J} is associated with a null space of dimension $p - r$ corresponding to all local pose changes which are not observable. Moreover, the controller $\dot{\mathbf{P}} = -\lambda\mathbf{J}^{-}\mathbf{E}$ changes the pose on an r -dimensional subspace in \mathbb{R}^p and not in the full pose space.

One may avoid this problem by making the rank full, either by integrating more image information (e.g. information from two cameras [7] or more [14]), or by using geometrical information on the observed object (e.g. its dimensions [6]). Yet another approach for dealing with a non-full rank is to reduce the dimension of the pose space controlled using the image born information so that the rank stays the same but becomes full. The control of the rest of the pose components is defined as a secondary task, which does not depend on the image based data, and is done using either another source of feedback or even in open loop (see [6] for a rigorous derivation of such control methodology).

3 Jacobian based visual servoing with non-full rank

The control scenario we have in mind is the following: one or more cameras, either stationary or moving, provide the visual feedback, but not all of them are available all the time, and both the information available and the Jacobian which characterizes its meaning, may change at every step. The control process should make the maximum use of the information at every step: if the current Jacobian is of full rank, then the control is simple and is done as described in section 2. If the rank is not full however, then a different technique is required. Occluded stereo and moving camera are two practical situations which are instances of this general scenario.

The main idea underlying the results of this paper is intuitively simple: the pose changes carried out according to a particular Jacobian are limited to an

In visual control tasks, we would like to use the information available from the image to direct the robot motion so that the object arrives at some predefined place which corresponds to the pose \mathbf{P}_d .

To achieve this goal, one can reconstruct the changing 3D pose and use it as a feedback, but one may also act in a more direct way, called *visual servoing* [6, 7, 15, 16, 17]: Let $\mathbf{M}_d = I(\mathbf{P}_d)$ be the image measurements corresponding to the required position(goal) and let $\mathbf{E} = \mathbf{M} - \mathbf{M}_d$ be the measurement error. Then, choosing the local control law

$$\dot{\mathbf{P}} = -\lambda \mathbf{J}^- \mathbf{E} \quad (1)$$

where \mathbf{J}^- is the pseudo-inverse of \mathbf{J} and $\lambda > 0$, leads to $\dot{\mathbf{E}} = \dot{\mathbf{M}} - \dot{\mathbf{M}}_d = \mathbf{J} \dot{\mathbf{P}} = -\lambda(\mathbf{J}\mathbf{J}^-) \mathbf{E}$. The matrix $\mathbf{J}\mathbf{J}^-$ is positive definite ([18]), and all components of the error vector \mathbf{E} converge uniformly and geometrically to zero. In practice, the controller is usually discrete and at every step, the robot is moved by

$$\Delta \mathbf{P} = -\lambda \mathbf{J}^- \mathbf{E}. \quad (2)$$

The advantage of such direct approach over reconstruction based control is that it does not need neither massive calculation nor accurate calibration and that it is more robust to changes in the scene. The implementations of such direct approach by the Jacobian based controllers (1), (2) are just one option of designing such controllers and other methods (e.g. optimal control, Kalman filter) were used as well [19, 8].

The convergence of the image error does not imply however that the pose reaches the required destination. The measurements may not be sensitive to every pose change, implying that some deviations from the required position are simply not observable in the image. The simplest example of this unfortunate phenomenon is the observation that a point in the image corresponds to many 3D points lying on a ray. Therefore, motion (or pose change) of the 3D point along this ray does not create any change in the image. In such cases the $m \times p$ Jacobian

issue both analytically and with simulations and suggest an alternative control principle.

The main contributions of this paper is the observation and the proof that Jacobian-based controllers are useful even if the space of uncontrollable pose is not zero (Jacobian matrix rank is not full). This observation allows to use visual servoing in some practical situations that were not considered before. A second contribution is the actual design and analysis of a non-trivial control task which relies on a moving camera.

The paper continues as follows: First we review some known facts on visual servoing and place our work in the correct context (section 2). Then, we turn to our subject here: robotic control using a sequence of images taken from a sequence of camera(s) locations. We provide generic conditions for convergence in section 3. The next part (section 4) introduces the particular control task that we study, controlling with a moving single camera, analyses a simplified case and the full 3D case, and ends with implementation details and some experiments (section 5). Finally, we draw some conclusions and set some open problem (section 6).

2 Jacobian-based visual servoing

Obviously, one image cannot reveal the entire 3D structure of the observed scene, and usually provides only some constraints on the 3D coordinates. The relation between the (p dimensional) vector, \mathbf{P} , of 3D pose parameters to the (m dimensional) vector of (image-based) measurements, $\mathbf{M} = I(\mathbf{P})$, is, in general, nonlinear and is therefore difficult to analyze and to use. Treating both \mathbf{P} and \mathbf{M} as time functions, however, we can write the time derivative of the image measurement as a linear function of a time derivative of the pose, $\dot{\mathbf{M}} = \mathbf{J}\dot{\mathbf{P}}$, where the components of the Jacobian matrix \mathbf{J} are the partial derivatives $\partial M_i / \partial P_j$. Thus, the Jacobian describes the relation between the pose and the image measurements locally.

controlled scene. We show here, that even in such cases, where at every moment the information available does not allow the controller to move the controlled object in the optimal direction, it is still beneficial to move it in a direction included in the reduced motion subspace. We suggest a generic control principle to be used in such situations and prove, under very mild conditions, that using this principle leads to the desired convergence.

This convergence is proved by observing that the sequence of object poses, implied by the proposed control principle, is similar to the sequence of values taken by a variable in common optimization techniques known as row-action methods [12] (or more generally, projections on convex sets (POCS) algorithms [13]).

To test the proposed framework empirically, we focus on a particular task: insertion of a rod into a pipe using feedback provided by a single camera able to move around the scene. A similar peg-in-hole task was studied by Yoshimi and Allen [14] and a rotating camera was used in their solution as well. Yet, there are crucial difference between the tasks and the solutions: first we consider a less constrained setting where the camera rotation center do not need to be the hole nor the rod itself. Also, we challenge a full 3-D alignment task where not only the position but also the orientation of the rod needs to be controlled. The most substantial difference is that they collect several images before every move, yielding a stereo-like situation where all the information needed is available before every move. We, on the other hand, use only one image and work with the partial information it contains.

We derive control laws, both for the robot holding the rod and for the camera position. While convergence is guaranteed for nearly any camera motion rule, some rules are better than others. We suggest such a rule and discuss some new considerations which arise in the moving camera scenario. In particular, the rule “move the robot so that the image error is minimized” leads to a slow convergence rate and other disadvantages and thus should be avoided. We investigate this

that this task may be carried out, in principle, for any camera movement, and suggest efficient control and camera moving strategies. Interestingly, it seems that the advisable control law is not consistent with simple-minded intuition. We substantiate our claims by simulations and experiments.

1 Introduction

Integrating control with visual feedback holds much promise for useful applications such as vehicle guidance[2], [3], treating sensitive materials, etc., and was shown to be feasible in several demonstrations (see some examples in [4, 5, 6, 7, 8, 9] and section 2, where we review some of these approaches).

The use of visual feedback is challenging because, due to the information loss inherent to the imaging process, the visual sensor provides only partial information about the object pose. In particular, at every given pose, there are several object motion (pose change) directions which do not change the location of its projection on the image, implying that pose control along these directions cannot rely on the visual feedback. Parameterizing the motion, we may say that there is an uncontrollable motion subspace. This phenomena is related to the variable sensitivity of control variables to the image measurements, analyzed in [10] and [11]. The common approach for handling this difficulty, is either to rely on richer visual feedback or to reduce the pose space, thus eliminating the uncontrollable motion subspace [6, 7].

This paper addresses situations in which this approach cannot be applied. We consider scenarios where the sequence of images available to the controller is taken from cameras in different locations and corresponds to different poses of the controlled robot. Such scenarios occur, for example, when a camera stereo pair is used but then, occasionally (or frequently) one of the cameras' fields of view is obscured by the robot manipulators. A similar yet different situation occurs when the information available comes from a single camera which is moving around the

Robotic Control with Partial Visual Information ^{*}

Keisuke Kinoshita [†] and Michael Lindenbaum [‡]

Abstract

We consider a robotic setting and a class of control tasks, which rely on partial visual information. These tasks are hard in the sense that at every given moment, the available information is insufficient for the control task. This implies that the image Jacobian, which relates the image space and the control space, is not of full rank anymore. However, still, the amount of information collected throughout the control process is large and thus seems sufficient for carrying out the task. Such situations commonly arise when the object is frequently occluded from one of the cameras in a stereo pair or when only one moving camera is available.

We propose a generic control rule for such tasks and characterize the conditions required for the success of the task. The analysis is based on the observation that mathematically, the behavior of such systems is related to a class of row-action optimization algorithms also known as POCS (Projection On Convex Sets) algorithms.

In the second part of the paper we focus on one particular task from this class: position and orientation control with a single rotating camera. We show

^{*}Submitted to International Journal of Computer Vision, Special Issue on Image-based Robot Servoing. A preliminary version of this work was described in [1].

[†]Human Information Processing Lab., ATR, Kyoto 619-02 Japan

[‡]Computer Science Dept., The Technion, Haifa 32000 Israel. The work was done while M.L. was staying in ATR