

# Visual Servoing for Floppy Robots Using LWPR

Fredrik Larsson\*

Erik Jonsson \*

Michael Felsberg\*

## Abstract

We have combined inverse kinematics learned by LWPR with visual servoing to correct for inaccuracies in a low cost robotic arm. By low cost we mean weak inaccurate servos and no available joint-feedback. We show that from the trained LWPR model the Jacobian can be estimated. The Jacobian maps wanted changes in position to corresponding changes in control signals. Estimating the Jacobian for the first iteration of visual servoing is straightforward and we propose an approximative updating scheme for the following iterations when the Jacobian can not be estimated exactly. This results in a sufficient accuracy to be used in a shape sorting puzzle.

## 1 Introduction

Initially an analytical closed-form inverse kinematics solution for a 5 DOF robotic arm was developed and implemented. This analytical solution proved not to meet the accuracy required for a general assembly setup, e.g. a shape sorting puzzle like the one used in the COSPAL (*COgnitive Systems using Perception-Action Learning*) project [1, 4]. The correctness of the analytical model could be confirmed through a simulated ideal robot and the source of the problem was deemed to be nonlinearities introduced by weak servos unable to compensate for the effect of gravity. Instead of developing a new analytical model, which took the effect of gravity into account, a learning approach was selected.

As learning method we chose Locally Weighted Projection Regression (LWPR) [11]. This is an incremental supervised learning method and is considered a state-of-the-art method for function approximation in high dimensional spaces.

LWPR by itself was not able to give us the accuracy needed and we combined the trained LWPR model with the well known concept of *visual servoing* [8]. We show how to overcome

---

\*Computer Vision Laboratory. Dep. of EE, Linköping University, Sweden [larsson@isy.liu.se](mailto:larsson@isy.liu.se).

This work has been supported by EC Grant IST-2003-004176 COSPAL. This paper does not represent the opinion of the European Community, and the European Community is not responsible for any use which may be made of its contents.

the difficulties that arise from the merge of the two methods and present results showing a high level of accuracy.

## 2 LOCALLY WEIGHTED PROJECTION REGRESSION

LWPR is an incremental local learning algorithm for nonlinear function approximation in high dimensional spaces and has successfully been used in learning robot control [10, 9]. The key concept in LWPR is to approximate the underlying function by local linear models. The LWPR-model automatically updates the number of receptive fields (RFs), i.e. local models, as well as the location (which is decided by the RF center  $\mathbf{c}$ ) of each RF. The size and shape of the region of validity (decided by the distance metric  $\mathbf{D}$ ) of each RF is updated continuously based on the performance of each model. Within each local model an incremental version of weighted partial least-squares (PLS) regression is used.

LWPR uses a Gaussian weighting kernel to calculate the *activation* or *weight* of RF  $k$  (the subscript  $k$  will be used to denote that the particular variable or parameter belongs to RF  $k$ ) given query  $\mathbf{x}$  according to

$$w_k = \exp\left(-\frac{(\mathbf{c}_k - \mathbf{x})^T \mathbf{D}_k (\mathbf{c}_k - \mathbf{x})}{2}\right). \quad (1)$$

Note that (1) can be seen as a non-regular channel representation of Gaussian type if the distance metric  $\mathbf{D}_k$  is equal for all  $k$  [5].

The predicted output  $\hat{\mathbf{y}}$  is given as the weighted output of all RFs according to

$$\hat{\mathbf{y}} = \frac{\sum_{k=1}^K w_k \hat{\mathbf{y}}_k}{\sum_{k=1}^K w_k} \quad (2)$$

with  $K$  being the total number of RFs.

The output of each RF can be written as a linear mapping

$$\hat{\mathbf{y}}_k = \mathbf{A}_k \mathbf{x} + \beta_{k,0} \quad (3)$$

where  $\mathbf{A}_k$  and  $\beta_{k,0}$  are known parameters acquired through the incremental PLS. The incremental PLS bears a resemblance to incremental associative networks [7], one difference being the use of subspace projections in PLS.

We have been using LWPR to learn the mapping between the configuration  $\mathbf{x}$  of the end-effector and the control signals  $\mathbf{y}$ . All training data was acquired through image processing since no joint-feedback was available from the robotic arm used. To reach a high level of accuracy we combined the moderately trained LWPR model with visual servoing.

### 3 VISUAL SERVOING BASED ON LWPR

We have been using position based visual servoing (categorized according to [8]) to minimize the norm of the deviation vector  $\Delta \mathbf{x} = \mathbf{x}_w - \mathbf{x}$ , where  $\mathbf{x}$  denotes the reached configuration and  $\mathbf{x}_w$  denotes the desired configuration of the end-effector.

If the current position with deviation  $\Delta \mathbf{x}$  originates from the control signal  $\mathbf{y}$ , the new control signal is given as  $\mathbf{y}_{\text{new}} = \mathbf{y} - \mathbf{J}\Delta \mathbf{x}$ , where the Jacobian  $\mathbf{J}$  is the linear mapping that maps changes  $\Delta \mathbf{x}$  in configuration space to changes  $\Delta \mathbf{y}$  in control signal space. When the Jacobian has been estimated the task of correcting for an erroneous control signal is in theory rather simple.

Using LWPR as a basis for visual servoing is a straightforward procedure for the first iteration. LWPR gives a number of local linear models from which the Jacobian can be estimated. However, problems arise when we need to update the Jacobian to use it for the following iterations.

Equation (1),(2) and (3) give  $\mathbf{J}$  as

$$\mathbf{J} = \frac{d\hat{\mathbf{y}}}{d\mathbf{x}} = \frac{\sum_{k=1}^K w_k (\mathbf{A}_k + (\hat{\mathbf{y}} - \hat{\mathbf{y}}_k)(\mathbf{x} - \mathbf{c}_k)^T \mathbf{D}_k)}{\sum_{k=1}^K w_k}. \quad (4)$$

The problem of updating  $\mathbf{J}$  after the first iteration is due to the fact that the current output was obtained by use of the old  $\mathbf{J}$  and not by the LWPR model. This means that we do not know which query  $\mathbf{x}$  that would give us the current  $\hat{\mathbf{y}}$  and as can be seen in (4) this is needed. The solution to this non-trivial problem is the main contribution of this paper. We propose a static approach and an approximative updating approach.

*Static approach:* The simplest solution is the static approach. The Jacobian is simply not updated and the Jacobian used in the first step is (still) used in the following steps.

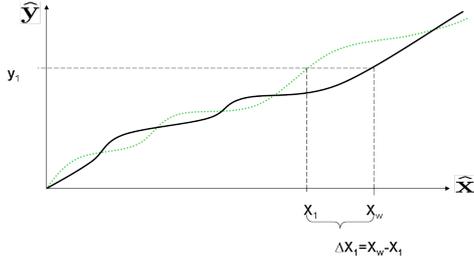
*Approximative updating approach:* The somewhat more complex solution treats the LWPR model as if it was exact. This means that we use the reached position as query and estimate the Jacobian for this configuration. The procedure is explained in Figure 1.

### 4 RESULTS

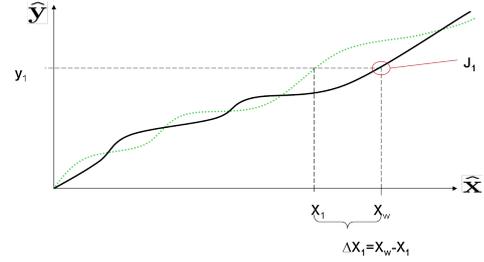
The real world experimental setup consisted of a low cost robotic arm of Lynx-6 type [2] (see figure 2) and a calibrated stereo rig. The end-effector of the robotic arm was equipped with three spherical markers in distinct colors. By stereo triangulation, the 3D position of the spherical markers were obtained relative one of the cameras. The positions were then

transformed into the robot frame. For the results presented below, we only deal with the 3D-position of the end-effector, neglecting the rotation and approach angle.

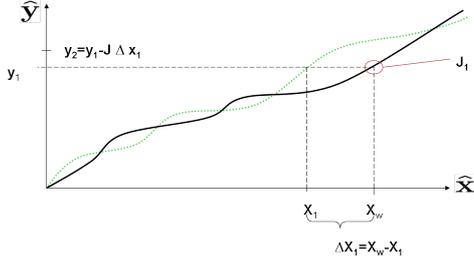
The test scenario used is a reduced 3D scenario. The end-effector can be positioned in two different planes (the grip- or the movement-plane) and the approach vector is to be perpendicular to the ground plane. The task space of the robotic arm is restricted (by physical and practical constraints) to a half circle with radius of 240 mm. Training points were acquired by using the inaccurate analytical model.



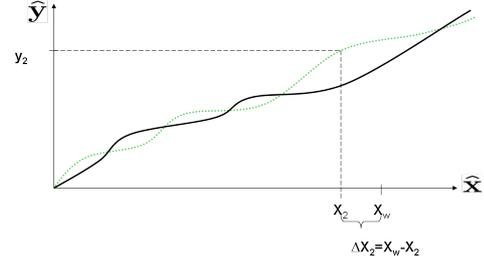
**A** : Given the wanted configuration  $\mathbf{x}_w$  we obtain the first prediction  $\mathbf{y}_1$ . Which results in deviation  $\Delta \mathbf{x}_1$ .



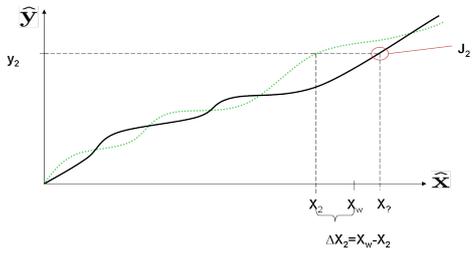
**B** : The true Jacobian  $\mathbf{J}_1$  is estimated.



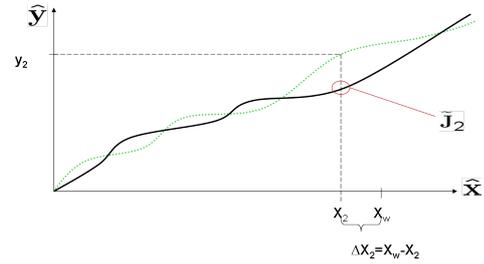
**C** : The prediction is updated, giving  $\mathbf{y}_2$ .



**D** :  $\mathbf{y}_2$  results in  $\mathbf{x}_2$  with deviation  $\Delta \mathbf{x}_2$ .



**E** : The true Jacobian  $\mathbf{J}_2$  can not be estimated due to the unknown  $\mathbf{x}_3$ .



**F** : The approximative Jacobian  $\tilde{\mathbf{J}}_2$  is estimated.

Figure 1: The approximative updating approach explained. The dotted line represents the true function and the solid line the LWPR approximation.

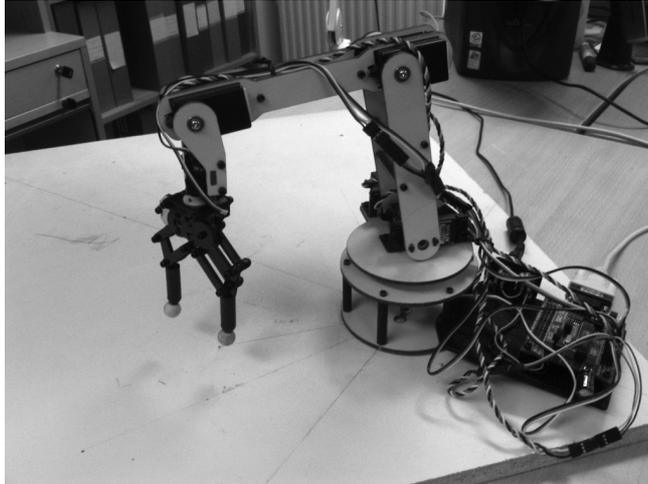


Figure 2: Our Lynx-6 robotic arm positioned in the movement-plane. The spherical markers can be seen at the end of the end-effector

Table 1 contain the results from real world experiments. LWPR denotes the mean deviation with just the trained LWPR model. J Static/Update denotes whether the static or the updating approach has been used for visual servoing. It is worth noticing that perfect positioning with just the estimated noise added would correspond to a mean error of 2.05 mm.

Real World Evaluation				
Training points:	100	500	1000	5000
LWPR	16.89	12.83	7.53	8.78
J Static	9.83	5.41	1.79	1.64
J Update	9.07	4.32	1.65	1.65
Analytical solution:	15.87			

Table 1: Mean deviation in mm from desired position. 50 test points were used for evaluation except from in the analytical case were 100 test positions were used. Stopping criteria for the visual servoing was 10 iterations or a deviation less than 1 mm.

## 5 DISCUSSION

By combining LWPR with visual servoing we have reached an accuracy sufficient for a shape sorting puzzle. The main novelty of this paper is to present two methods to overcome the difficulties that arise from the merge of LWPR and visual servoing. The results show that the approximative updating approach is favorable. To further improve the accuracy we would need to reduce the noise in the estimated positions since it is currently the limiting factor.

The restrictions imposed by the test scenarios mean that we are avoiding the problems with the solution to the inverse kinematic problem not being one to one. However, if the training data shown to the LWPR model would have some ambiguities this may cause problems. In fact, if all positions would be reachable with servo 1 set to e.g.  $+\pi$  or  $-\pi$ , the linear averaging of the LWPR model will predict the output for servo 1 to 0. Of course, this can be avoided with preprocessing of the signals, e.g. using the *channel representation* [6] which allows for robust estimation of multiple modes [3].

## References

- [1] COSPAL project. <http://www.cospal.org/>, January 2007.
- [2] Lynxmotion robot kits. <http://www.lynxmotion.com/>, January 2007.
- [3] M. Felsberg, P.-E. Forssén, and H. Scharr. *IEEE Transactions on Pattern Analysis and Machine*, 28(2):209–222, February 2006.
- [4] M. Felsberg, J. Wiklund, E. Jonsson, A. Moe, and G. Granlund. Exploratory learning structure in artificial cognitive systems. In *ICVW*, 2007.
- [5] P.-E. Forssén. *Low and Medium Level Vision using Channel Representations*. PhD thesis, Linköping University, Sweden, SE-581 83 Linköping, Sweden, March 2004. Dissertation No. 858, ISBN 91-7373-876-X.
- [6] G. H. Granlund. An associative perception-action structure using a localized space variant information representation. In *Proceedings of Algebraic Frames for the Perception-Action Cycle (AFPAC)*, Kiel, Germany, September 2000.
- [7] E. Jonsson, M. Felsberg, and G. Granlund. Incremental associative learning. Technical Report LiTH-ISY-R-2691, Dept. EE, Linköping University, Sept 2005.
- [8] D. Kragic and H. I. Christensen. Technical report, ISRN KTH/NA/P-02/01-SE, Jan. 2002., CVAP259.
- [9] S. Schaal, C.G. Atkeson, and S. Vijayakumar. Scalable techniques from nonparametric statistics for real time robot learning. *Applied Intelligence*, 17(1):49–60, 2002.
- [10] S. Vijayakumar, A. D’souza, T. Shibata, J. Conradt, and S. Schaal. Statistical learning for humanoid robots. *Auton. Robots*, 12(1):55–69, 2002.
- [11] S. Vijayakumar and S. Schaal. Locally weighted projection regression: An  $O(n)$  algorithm for incremental real time learning in high dimensional spaces. In *Proceedings ICML*, pages 288–293, 2000.