Comparison of Control Structures for Visual Servoing

S. Chroust[†], J.P. Barreto[‡] **H. Araújo[‡], M. Vincze[†]**

[†]Vienna University of Technology
Institute of Flexible Automation
Gusshausstr. 27-20 /361
{sc,vm}@infa.tuwien.ac.at

[‡]Polo II - University of Coimbra
Institute of Systems and Robotics
3030 Coimbra
{jpbar,helder}@isr.uc.pt

Abstract

This paper deals with active tracking of 3D target motion. Visual servoing has the main problem of handling the latency produced by acquiring and processing the image produced by the CCD-camera. The performance of the visual control structure depends both on the visual process and the control structure. In this work different control structures are investigated. To characterise the system performance from the control point of view four standard types of target motions are used: ramp, step, parabola and sinusoid. The quality criteria of the different structures allow conclusions of an optimal system structure. The main result is that the usage of a higher sampling rate for the mechanism improves the transient behaviour and that a feedforward signal is only advantageous for smooth motions.

1 Introduction

Visual control of motion is a main task in active vision that includes complex topics of visual processing and control. It is a powerful technique to control the motion of a mechanism from images, as demonstrated in recent workshops and tutorials ([5, 8]). Visual servoing can either use cameras that are fixed in the work space or mounted on a robot or an active head. Applications range from following a person with an active head over steering cars to grasping moving parts with a robot. Besides the reliability of image processing, it is of high interest to optimise the entire visual control loop to obtain good dynamic performance.

The latency introduced by the visual feedback is one of the reasons that make vision-based control so difficult. Latency is introduced by the time necessary to acquire the image and by processing the image (or parts of the image) to obtain the target information needed at the controller. The main problem is that the target motion as the "real" reference for controller is delayed due to the visual process. This subject is exhaustively discussed in [6].

One way to cope with the latency is prediction of the target state. A common approach is the usage of a Kalman Filter [10] or Model Predictive Control [4]. Besides the predictive control strategy Corke shows in [1] that a feedforward structure is necessary to achieve high performance in visual servoing. This will be a further important part of the control architectures investigated.

The goal of this work is to investigate different control architectures to find and to analyse the characteristics to find a general rule to build up a control loop for visual servoing. Therefore a comparison of these different structures using the same control algorithm and mechanical setup is done to achieve guidelines for an optimal control structure.

In section 2 this work starts with the description of the hardware used and the visual process to track the target in 3D space. In section 3 the different control architectures are explained and an overview is given of their properties. Section 4 illustrates these properties with measurements and points out the differences in more detail. A conclusion summarizes the results of the experiments and gives an outlook over future work.

1.1 Previous Work

Vision-based control of motion using an active head or a robot has been demonstrated in many systems (see overviews in [5, 7]). Each system gives an example of an image processing and control architecture optimised regarding dynamic performance. The complexity of control architectures varies from structures with a controller in the forward tree added with a feedforward signal ([1, 3]) to systems with complex cascade control structures [2]. The results achieved are dependent on the vision system used and of course on the mechanism and the controller.

Independent of this variety of systems implemented, prediction of the target state, e.g. after Kalman [9] and Model Predictive Control (MPC) are a common approach to cope with the delay. A systematic investigation of the dynamic behaviour of vision-based control of motion was started in [1] using both a simple feedback controller and a feedforward structure. The performance of MPC is shown for example in [4]. Also the possibility of a combination of these two predictive control strategies is investigated. Comparing these single results is not so easy as they depend strong on the system.

2 System Architecture

The system used for vision-based control of motion is an active head. Tracking is performed in 3D space with two independent rotational axes: pan and tilt. The motion of the target in the image depends on the target motion and on the motion of the mechanism (egomotion). For visual control we are only interested on the motion of the target, thus egomotion has to be compensated. The system consists of four main parts. A block diagram is given in the following section as it depends on the control architecture.

The CCD-camera running at 25 Hz gathers information of the target. A standard PC processes the image using optical flow. The target velocity is calculated using two consecutive images. The difference image, obtained after egomotion compensation, contains the points where motion occurred. Position is estimated as the average location of the set of points with non-zero optical flow and brightness partial derivatives, with respect to X and Y. It is assumed, that all moving pixels in the image have the same velocity. The velocity vector is estimated considering the flow constraints and applying least-square estimation. The visual delay consists two parts, the time needed for acquiring (40 ms) and processing (6 ms) the image.

A Kalman Filter is used to estimate the targets angular parameters (image position $\Delta\Theta$, velocity ω and acceleration α) assuming a constant acceleration model between frames. Further tasks of the Kalman Filter are the prediction of the target state and the calculation of interpolated values between two frames.

The controller is used in different ways dependent on the control architecture. In this work an optimised GPC is always used to compare the control structures. The controller works either as main controller receiving the target information direct from the Kalman Filter or as an auxiliary controller responsible to hold a reference, which is generated from the Kalman Filter / Interpolator with the knowledge of the target and motor state. In the first case the whole loop has to be modelled including the visual process, in the second case the model for the controller only includes the motor.

DC-motors equipped with optical encoders for position feedback generate the motion of the active head. An independent module that implements an inner closed loop with a digital PID filter running at 1 kHz controls each axis. Each servo loop can be commanded in velocity by adding a profile generator that integrates the velocity sent by the superimposed process. Communication is synchronous at a frequency of 166 Hz. For this reason two communication rates are possible, either the maximum rate of 166 Hz or a rate of 25 Hz given by the visual process. Depending on the communication rate two different models of the motors are identified using standard identification techniques.

The higher sampling rate gives the possibility to build up a middle level loop running at 166 Hz embedded in the visual process running at 25 Hz. The separation between these two rates can be done either in an interpolator or in the controller. The higher communication ensuring a tight control of the motors. When using the middle level loop a reference for the auxiliary controller is used. In this case the Kalman Filter handles the dynamic behaviour of the visual system.

3 Control Strategies

In vision-based control of motion the target motion is not direct accessible and acts therefore as disturbance. To implement high performance tracking certain issues, such as robustness to sudden changes of the target trajectory and velocity must be taken into account. Evaluation of both vision and control strategies within a common framework is needed for optimising the global system performance. This framework has been studied in previous works [1, 2, 3]. The controller cannot react immediately to changes in the target motion as the visual process delays the information. A certain overshoot is therefore unavoidable. To minimise these overshoot an investigation of different control architecture is needed.

The different architectures can be divided in two groups. The first group works completely at 25 Hz frame rate, as the second group uses a middle level loop with the maximum communication rate at 166 Hz. This ensures a more tight control of the motor. The main limitation of visual servoing still comes from the visual system but the middle level loop improves the transient response.

3.1 Architecture without middle level loop

The basic visual control loop is displayed in fig. 1. The whole control loop works at 25 Hz (40 ms). The motor motion X_m superimposes the target motion X_t, which is the input for the vision system $V(z)$ that includes the CCD-camera and the visual processing. The output is the image plane error X and the target velocity in the image $\dot{X}$, which is the input to a Kalman Filter KF. The Kalman Filter calculates the angular position $\Delta\Theta$ and velocity α and predicts the target state in the future. The controller $C(z)$ calculates a velocity signal u_c to move the mechanism in the direction of the target. Velocity control is used to avoid path planning. The model of the motor includes therefore one integrator to convert the velocity command into a position. This structure is denoted as **40p**.

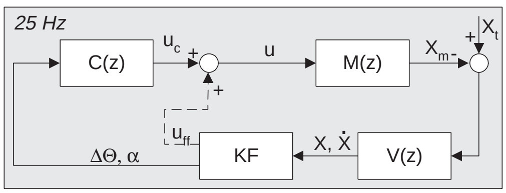

Figure 1: Control loop without middle level loop using only visual information.

To increase the performance a feedforward signal u_{ff} is added (**40pf**). The feedforward signal is the measured target velocity, which is added to the control signal calculated by the controller. For the GPC a feedforward signal is a disturbance, but as the disturbance works against the second disturbance in this loop (the target motion) the system behaviour is improved, dependent on the type of motion.

The advantage of these two structures is the easy implementation and the rather smooth behaviour, it is not necessary to observe the position of the motor, the information is delivered purely from the vision system. The feedforward signal is especially advantageous for a sinusoidal motion or a parabola, because these motions are smooth.

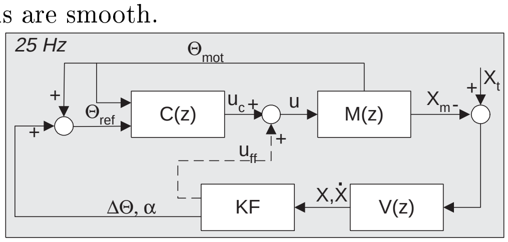

Figure 2: Control loop without middle level loop using visual and motor information.

The next architecture is shown in fig.2. In this case a reference trajectory is generated with the usage of the position information of the motor. The generation of the reference makes only sense, because the GPC does not use the control error $e(t) = \Theta_{ref}(t) - \Theta_m(t)$ but a filtered control error $e^f(t) = \Theta^f_{ref}(t) - \Theta^f_m(t)$. For the design of the controller the difference arise from the fact that the model for the GPC does not include the transfer behaviour of the vision system. In principle the GPC acts only on the motor, while the motion of the target is handled via the generation of the reference with the Kalman Filter and the motor position.

Once again a feedforward signal can be added. The disturbance is stronger in this case compared to the first architecture, as the GPC acts direct on a reference and not on a disturbance as in the first case. Due to the fact, that the reference is rectified every step the response is improved dependent on the type of the motion. The two structures are denoted in section 4 by **40pr** and **40prf**.

The problem of the generation of the reference is a time shift between the measured values of the vision system and the position of the motor. This effect can lead to oscillation. An advantage compared to the first two structures is the possibility to tune the transfer and noise behaviour separately.

3.2 Architecture with middle level loop

A middle level loop is used to increase the performance of the visual loop, see fig. 3. The difference to the structures in fig. 1 and fig. 2 is that the blocks $C(z)$ and $M(z)$ work at 166 Hz sampling rate. The new sampling rate results in a new motor model. The reference is generated in the Kalman Filter including visual information (velocity and position) and the motor position at time instant of the grabbed image. The reference produced is a position signal therefore the controller can directly use the position values of the encoder to calculate the filtered control error.

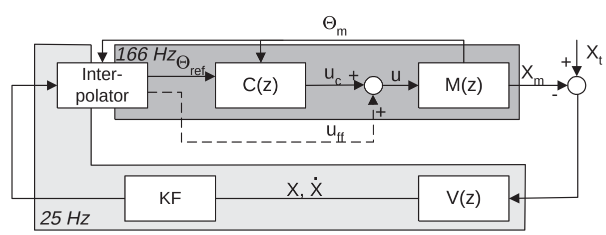

Figure 3: Control loop with middle level loop using position information in the middle level loop.

Four possibilities exist to use this structure, with and without interpolator and in each case with an additional feedforward signal. The easiest way is a stepwise reference, where the separation between the subsystems working at 25 Hz ($V(z)$ and KF) and at 166 Hz ($C(z)$ and $M(z)$) is done with the controller. The interpolator described in section 2 is not used in this case. The feedforward signal introduced is no improvement because the controller tries to hold the reference during the whole period of 40 ms. The feedforward signal tries to move the motor in the direction of the target motion but as the controller does not have the information, that this movement corresponds better to the target motion than the reference, the controller works against the feedforward signal. These two control architectures are shortened with **6po** and **6pof**.

The next step is the introduction of an interpolator seen in fig. 3 (**6p** and **6pf**). This interpolator generates with the velocity and position information a reference trajectory, which smoothens the steps between two images. The tuning of the controller can be done faster, as the reference is smoother.

This control structure shows an improved behaviour compared to the structure without interpolator. The Interpolator ensures a good reaction and the introduction of the velocity measured in the Kalman Filter is straightforward. The position measured is very important for the generation of the reference, therefore the position signal must be reliable not robust.

The last structure in fig. 4 uses velocity information in the middle level loop (**6v**). To ensure an offset free control it is necessary to introduce the position error in the reference. Therefore the usage of the interpolator is important. The generation of the reference is not so straight forward as in the structures using position information in the middle level loop, because the position error must be converted into a velocity signal. This introduction may cause problems, because small position errors can lead to oscillations and have to be damped. Despite this problem the usage of the velocity signal might be necessary when the velocity signal measured is more reliable than the position information. Another advantage is that the interpolator increases the type of the system without adding an explicit interpolator. A disadvantage is that the generation of the reference must be investigated carefully as a bad tuning can lead to an oscillating behaviour. The motor velocity for the controller is calculated with a discrete differentiation out of the encoder signal.

Without a feedforward signal the GPC yields good results for vision-based control of motion. The problem with combining the GPC with a feedforward structure is, that the GPC calculates an optimal sequence of control increments taken into account the past increments. The feedforward signal acts as a disturbance at the input of the plant and this is a rather strong perturbation for the controller. This affects the controller especially when the feedforward signal changes abrupt like for a ramp or even a step.

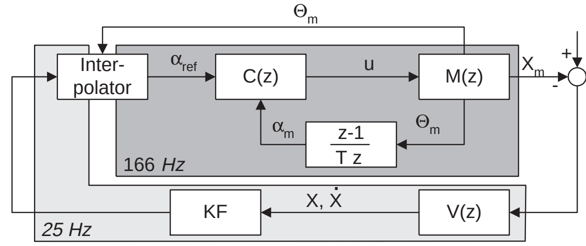

Figure 4: Control loop with middle level loop using velocity information in the middle level loop.

4 Experimental Results

The experiments are done with a stereo head using "synthetic images". The trajectories applied are a ramp, a sinusoidal and step with a ramp. The target motion is generated with synthetic images to receive comparable trajectories with exact motion parameters. The usage of this technique instead of real images has several advantageous. First the mathematical functions corresponding to the target motion could be generated accurately and repeatable. Second the comparison of the control architecture, which is the main goal of this work, is not so dependent on the visual process. The images have to be generated online, as the captured frames depend not only on the predefined motion but also on the camera orientation.

Fig. 5 and fig. 6 show the image plane error ΔX in $Pixel$ of the control architectures for a sinusoidal motion. Both images use the same scaling factor of the axes for an easier comparison.

When comparing the structures without middle level loop in fig. 5 it is clear, that the feedforward signal gives an improvement, because the sinusoidal motion is smooth and well predictable. Therefore the maximum image plane error is reduced nearly by a factor two. The important thing during the sinusoidal motion is, that the motion can be divided in two parts. During one part the position error is small and the velocity is high, in this stage the main control effort comes from the feedforward signal. During the other part the position error is large and the velocity is high and therefore the GPC reacts more active. With this separation the main control effort changes from the GPC to the feedforward signal and back.

In fig.6 the structures show a little bit different behaviour. The feedforward structure gives now not that improvement. For the structure without interpolator (**6po, 6pof**) this comes from the fact that the GPC works against the feedforward signal as explained in section 2. For the structure with interpolator (**6po, 6pof**) the inclusion of the velocity signal in the reference is a type of feedforward signal, so there is nearly no improvement due to the introduction of the feedforward signal at the input of the plant.

The the structures **6p**, **6pf** and **6v** shows the best behaviour. Structure **6v** has a little bit larger error, because the information of the position error is not taken full into account. The larger error occurs at the peaks in the position. In this moment the position error represents the main influence to the reference trajectory and, because the weight to this position error is decreased, it is clear that the reaction becomes slower. Comparing the results of this structures with the results achieved with the structures **40pf** and **40prf** at can be seen, that the maximum error is nearly the same (5 Pixel for **40pf** and **6p**). So the middle level loop does not give a large improvement during the steady state. An improvement can be seen on the end of the sinusoidal motion, where the structures with middle level loop show a smaller overshoot.

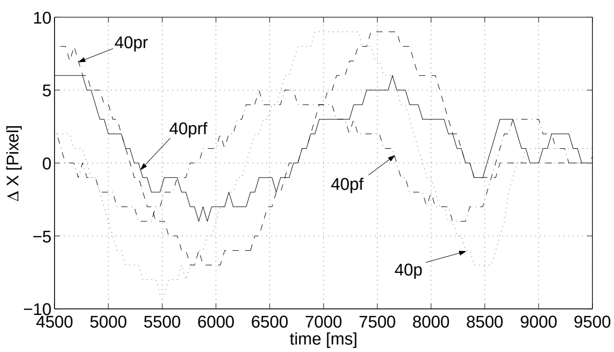

Figure 5: Image plane error ΔX as response to a sinusoidal target motion for the structures without middle level loop.

Fig. 7, shows for a better comparison, three quality criterions of the sinusoidal motion (the criterions are always normalized with the mean value). The quality criteria calculated are the random least square (RLS) value of the image plane error ($RLS(\Delta\Theta)$, of the velocity error ($RLS(\alpha)$) and of the control signal defined by

$$RLS(\Delta u) = \sqrt{\frac{1}{\Delta t}\int_{t_0}^{t_0+\Delta t}[\Delta u(t)]^2} \qquad (1)$$

to assess the control effort. The structures **6po** and **6pof** have a much higher control effort with a large

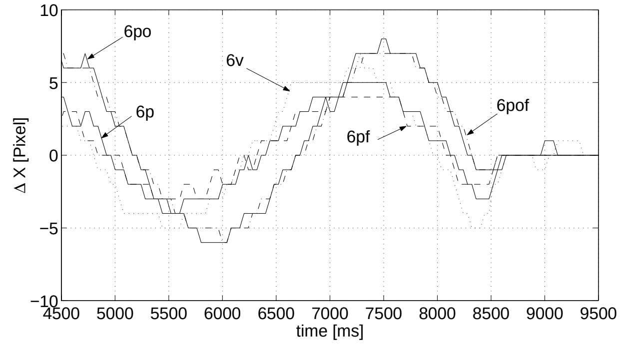

Figure 6: Image plane error ΔX as response to a sinusoidal target motion for the structures with middle level loop.

velocity error and larger position error. Also it is obviously, that the structures with feedforward signal have a lower position error without a much higher control effort. Comparing the RLS-values of the velocity error and control increment it must be mentioned, that this values contain more data for the structures with middle level loop, as the sampling rate is higher. Only the structure **6v** shows low values for all three criterions.

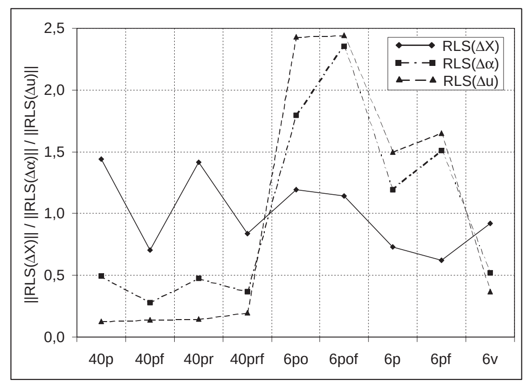

Figure 7: Quality criteria of the nine control structures for a sinusoidal motion.

Fig. 8 shows the quality criterions for a ramp like motion. The RLS-value of the velocity is now replaced by the standard deviation of the velocity during the steady state of the ramp. Once again the structure **6v** shows for all criterions very good values. The structures **6p** and **6pf** have a lower value of the position error for the cost of a higher velocity value and a much higher control effort.

The last experiment is a step in the position with parabola starting immediately after the step. Once again the quality criterions are displayed in fig. 9. In this case the velocity is not the critical variable and therefore the maximum image plane error Δx_{max} is displayed. In this case the feedforward signal is no improvement compared to the structures without feed-

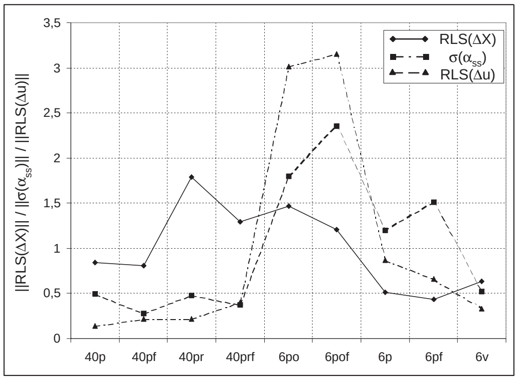

Figure 8: Quality criteria for the nine control structures for a ramp with about 37 $Pixel/s$.

forward signal because the values for the position error is larger. The structure **6v** shows once again a good compromise of good error values with a decent value of the control effort. Conspicuous are the values of **40pr** with the lowest values of ΔX_{max} and $RLS(\Delta X)$. In the case of the step the assumption made for the generation of the reference are quite good (also for **6po, 6pof**) and therefore the reaction of the controller is very well.

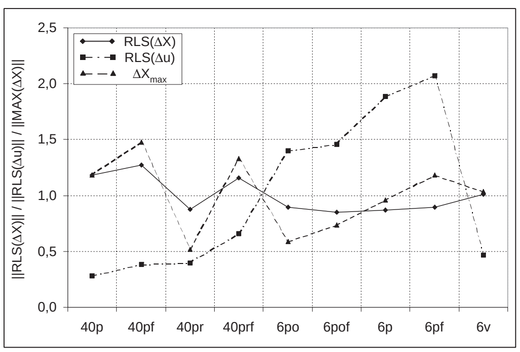

Figure 9: Quality criteria for the nine control structures for a ramp with about 37 $Pixel/s$.

5 Summary and Conclusions

The main objective in this work was to find the optimal control architecture for visual-based control of motion. The experiments show that the introduction of a middle level loop, sampled at a higher rate, gives an improved transient response of a higher control effort. Also the interpolator is very important when using the middle level loop. This interpolator is very critical when using the velocity information in the middle level loop. The decision of using either velocity or position information in the middle level loop is mainly answered what signal is more reliable.

It seems that the generation of the reference for a structure without middle level loop does not have a big advantage. The time of the visual and motor information must fit well otherwise an oscillating behaviour occurs. The feedforward signal is for structures without middle level loop only an advantage if the motion is predictable (sinus or parabola). This is an issue of the GPC as the controller calculates a sequence of optimised control increments.

Future works investigate the connection between GPC and the feedforward structure and will make a comparison with real images. A further investigation will be a state space controller, as this would solve the problems of introducing the position error in the velocity trajectory.

Acknowledgments

This work is partly supported by the Austrian Science Foundation (FWF) under grant P13167-MAT and Project GRD1- 1999-10693 FlexPaint.

References

[1] P.I. Corke, *Visual Control of Robots: High Performance Visual Servoing*, Research tSudies Press (John Wiley), 1996.

[2] J.P. Barreto, J. Batista, H Auraújo, "Model Predictive Control to Improve Visual Control of Motion: Applications in Active Tracking of Moving Targets", *ICPR2000 Int. Conf. on Pattern Recognition*, 2000.

[3] S. Chroust, M. Vincze, R. Traxl, P. Krautgartner, "Evaluation of Processing Architecture and Control Law on the Performance of Vision-Based Control Systems", *AMC200 6th Int. Workshop on Advanced Motion Control*, 2000.

[4] J.A. Gangloff, M.d. Mathelin, G. Abba, "Visual servoing of a 6 DOF manipulator for unknown 3D profile following," *IEEE ICRA*, 1999.

[5] S. Hutchinson, G.D. Hager, P. Corke, "Visual Servoing: A Tutorial," *IEEE Trans. RA* Vol.12(5), 1996.

[6] P.M. Sharkly, D.W. Murray, "Delay versus performance of visually guided systems," *IEE Proc.-Control Theory Appl.* Vol.143(5), 1996.

[7] M. Tonko, K. Schäfer, F. Heimes, H.H. Nagel, "Towards Visually Servoed Manipulation of Car Engine Parts," *IEEE ICRA*, 1997.

[8] M. Vincze, G.D. Hager, Eds., *Robust Vision for Vision-Based Control of Motion,* IEEE Press, 2000.

[9] W.J. Wilson, C.C. Williams Hulls, G.S. Bell, "Relative End-Effector Control Using Cartesian Position Based Visual Servoing," *IEEE Trans. RA* Vol.12(5), pp. 684-696, 1996.

[10] P. Wunsch, G. Hirzinger, "Real-Time Visual Tracking of 3D-Objects with Dynamic Handling of Occlusion," *IEEE ICRA*, 1997.