Proceedings of the 8th IEEE Mediterranean Conference
on Control and Automation (MED 2000)
Rio Patras, GREECE 17-19 July 2000

SOLUTIONS FOR VISUAL CONTROL OF MOTION: ACTIVE TRACKING APPLICATIONS

JOÃO P. BARRETO, JORGE BATISTA, HELDER ARAÚJO

Institute of Systems and Robotics, Department of Electrical Engineering—Polo 2, University of Coimbra, 3030 Coimbra, Portugal, jpbar@isr.uc.pt, batista@isr.uc.pt, helder@isr.uc.pt

Abstract. This paper deals with active tracking of 3D moving targets. The performance and robustness in visual control of motion depend on the vision algorithms and the control structure where dynamical aspects can not be neglected. Visual tracking is presented as a regulation control problem. Both system architecture and controller design are discussed. The performance of visually guided systems is substantially deteriorated by delays in the control loop. Interpolation is used to cope with visual processing delay. Model predictive control strategies are proposed to compensate for the mechanical latency and improve the global system performance.

Key Words. visual tracking, visual control of motion, active vision, robotic heads.

1.INTRODUCTION

Visual control of motion is a major issue in active vision, involving complex topics of both visual processing and control [3][5][7]. This work discusses the problem of tracking moving targets using visual information to control camera motion. An architecture to achieve this goal is presented.

Visually guided systems are systems whose actions are derived directly from image information. In a monocular tracking application the camera is mounted on an active plataform with two independent degrees of freedom: pan and tilt. For real-time visual control of motion three distinct concurrent processes can be identified: the visual processing of images, low-level servo control and high-level /gaze control.

Several strategies to extract visual information for motion control have been proposed [6][9]. The visual processing must be fast, accurate and robust to achieve high performance behaviors. Mounting the camera on an active platform creates additional difficulties due to the self induced image motion (egomotion). Kalman filtering can be used to estimate the 3D parameters of motion of the target and limit the effects of measurement errors in the image, allowing smooth tracking behaviors.

The low-level servo controller commands the active plataform actuators. The choice of these actuators and the corresponding control strategie are discussed in [11][12]. A local-servo loop with a PID controller is implemented to achieve high-performance motor control.

The gaze controller establishes the link between the visual processing and the active plataform actuators. It uses the information extracted from the images to compute the commands to be sent to the mechanical actuators. Delays in both feedforward and feedback paths of a dynamic system affect substantially the overall performance. This subject is exhaustively discussed in [3][8][10]. The latency introduced by visual feedback is one of the reasons that make vision-based control so difficult. Mechanical/ communication delays also decrease the global performance. The gaze controller is designed to compensate these delays and increase system bandwidth. Interpolation assuming a constant acceleration model of motion in

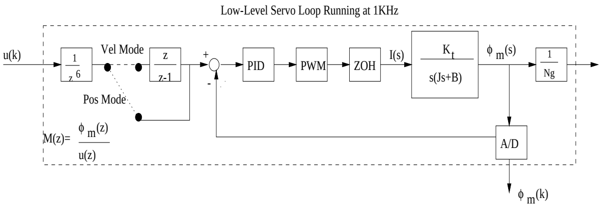

Figure 1: Servo Control Loop. The servo loop runs at 1KHz and can operate both on position and velocity mode. The transfer function corresponds to the DC direct drive. Ng is the reduction ratio due to the gearhead.

3D space is used to cope with visual processing delays. Model predictive control techniques are proposed to compensate process delay in vision-based control. In our case the plant is a robot-head. Plant models are obtained with standard system identification techniques and a dynamic matrix controller (DMC) is used in our tracking application. The performance of the DMC controller is discussed and compared with other possible control strategies.

2. LOW-LEVEL SERVO CONTROL

Stepper motor drives have been used in several active vision systems. In general it is easier to interface and control stepper motors than DC motors. However stepper motors have poor characteristics in terms of acceleration and smooth velocity tracking, which are major requirements for real-time active tracking. Therefore DC direct drives or DC geared drives are well suited to this kind of applications. The geared drives have the advantage that small units can provide high accelerations and attenuate disturbance torques. In our system, the plataform motion is generated by DC drives equipped with very low backlash harmonic gearheads. Position feedback is given by optical digital encoders mounted on the back shaft of the DC motor.

The motor position can be controlled directly using visual feedback. In this case the sampling rate is equal to the frame rate (25Hz). Fig.2 shows the motor frequency response. The extra phase-lag introduced by the visual processing will tend to worsen the stability of the system. One solution is to use small loop gains to keep the closed loop stability. The low sample rate associated to the small gains will lead to sluggish system responses.

The optical encoders provide high fidelity platform position information. Encoder information could be used to implement a local servo-loop running at a high sample rate. This would improve the motor frequency response. This system comprises a commercial multi-axis PC controller card with a dedicated servo control module for each degree of freedom of the platform. Each axis is controlled in position by a local closed loop with a digital PID filter

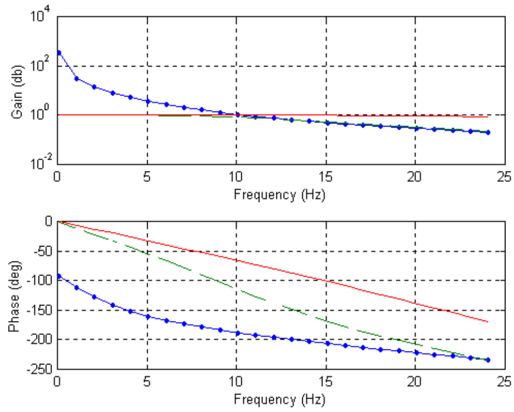

Figure 2: Motor Frequency Response. Open loop frequency response (.-). Closed loop frequency response in position mode (--) and on velocity mode (-). In this last case it is assumed that system output is motor velocity instead of position.

running at 1KHz. The use of commercial multi-axis control boards frees host PC processing time for running the high level visual algorithms. This is a cost effective and well balanced solution to build real time active vision systems. Communication between the host PC and the multi-axis board is synchronous at a frequency of 166Hz. This means that the user process can only send commands to the servo-loop and read the encoders in every 6ms interval. The communication delay introduces undesired phase-lag in the loop. Despite of that, Fig. 2 shows that by using a high gain, high sample rate local position controller, the motor frequency response improves. Additionally each servo loop can be commanded in velocity by adding a profile generator that integrates the velocities sent by the user process. This feature is useful on an active tracking vision system. In this type of system position control is used to implement saccadic behaviors, whereas velocity control is more suitable to perform smooth pursuit. Fig. 2 displays the frequency response when the motor is controlled in velocity. The improvements are due to the fact that the profile generator is running at 1KHz, updating the reference to the position loop every 1ms.

$M(z)$ is the transfer function of the servo-loop, considered from the point of view of the host PC (see Fig. 1).

3. CONTROL SCHEME

In a monocular tracking application the camera has typically two degrees of freedom: pan and tilt. The control of each of these degrees of freedom can be modeled by the schematic of Fig. 3. $M(z)$ represents the low-level control loop of Fig. 1 as seen by the user process at 166Hz (the communication frequency).

A visual active tracking system can be modeled as a regulator. The system input (reference) is the desired target position in the image ($X_R(k)$) and the output is

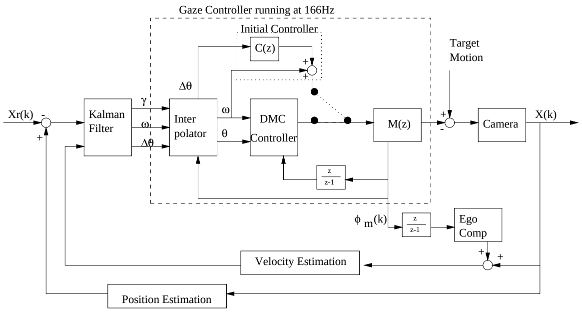

Figure 3: Smooth Pursuit Block Diagram. The visual loop runs at 25Hz while the sampling rate of the gaze controller is 166Hz. The figure depicts two different gaze controllers. One uses position error and velocity feedforward to compute motor commands. The other is based on a DMC controller. In both cases M(z) is assumed to be in velocity mode.

the actual target position in the image ($X(k)$). The goal of the system is to keep the projection of the moving target in the reference position (usually the center). Thus the target motion ($\theta(t)$) acts as a perturbation that has to be compensated for. To study and characterize the system regulation/control performance the usual control test signals (step, ramp, parabola and sinusoid) must be applied. The target trajectories to generate this signals have been established in [1].

The target position in the image yields the position error. Filtering it by $C(z)$ we obtain an actuation signal that is sent to $M(z)$. Notice that to achieve zero steady state error in position (type 1 system), $M(z)C(z)$ must have a pole in 1. By controlling the servo in velocity mode the motor frequency response is improved and the zero steady-state error can be guaranteed. Measuring velocity in the image after egomotion compensation yields an estimate of the target velocity in space. This information can be used to implement velocity feedforward and improve the system transient response. Notice that the velocity in the image must be measured after egomotion compensation. Otherwise the result is equivalent to include a derivative component in $C(z)$.

An alternative control strategy is to obtain a model of the plant ($M(z)$) and use model predictive control techniques to force the active platform to describe the desired trajectory. This trajectory is computed using estimates of the target position and velocity computed from visual processing. Both approaches were implemented and are compared in the last section of this paper, where the gaze controller is discussed.

The gaze controller runs at the communication frequency to optimize performance (166Hz). The visual feedback loop runs at 25Hz. In the next section the visual processing algorithms are briefly described.

4. THE VISUAL CONTROL LOOP

Visual processing latency affects the overall system performance. The computation time in extracting information from the images must be minimized. A trade-off between efficiency, robustness and accuracy must be achieved when selecting the visual processing algorithms. As it was previously stated, the position and velocity information are fundamental to achieve high performance smooth tracking behaviors. Thus both target position and velocity in the image must be estimated.

The image motion is a function of the target motion and the camera motion (egomotion). Since the image motion induced by the target has to be estimated, egomotion must be compensated for. Considering that the camera only performs pure rotations and that there is no motion in the scene, two images are related by a homography. The homography is easily computed if the camera rotation is measured using the motor encoders. Considering two consecutive frames, 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 non-zero brightness partial derivatives, with respect to X and Y, in the most recently grabbed image. It is assumed that all moving pixels in image have the same velocity. The velocity vector is estimated considering the flow constraint and applying a least-squares minimization. Multi-resolution gaussian pyramids are used to increase the range of image velocities that can be correctly estimated [2]. This leads to an increase in the velocity feedback bandwidth. Kalman filtering is used to estimate the target angular parameters of motion (error in position $\Delta\theta$, velocity ω and acceleration γ) assuming a constant acceleration model between frames. This assumption is acceptable for frame acquisition rates of 25Hz and higher. The Kalman filter tuning has been performed with the help of our evaluation tools [1].

Each grabbed image is processed (in a standard PC) and the parameters of the target motion are available at Kalman filter output 6ms after image acquisition. This is the visual latency. The high-level process (running at 25Hz) sends a time stamp and the motor position at the acquisition time instant to the gaze controller (running at 166Hz).

5. THE GAZE CONTROLLER

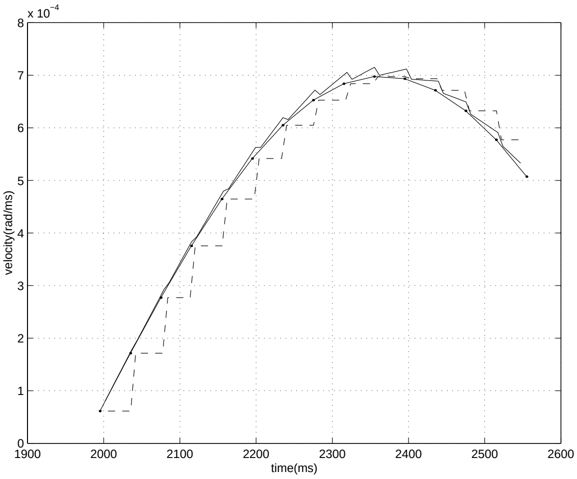

Figure 4: Interpolation. Target angular velocity (.-). Velocity estimation at gaze controller input before interpolation (--) and after interpolation (-).

The target motion data is sent to the gaze controller every 40ms, with a delay of 6ms. This introduces a phase-lag that deteriorates the global system performance. Interpolation assuming a constant acceleration model is used to cope with visual latency and to compensate the low sampling rate of the visual control loop. Fig. 4 compares target velocity estimation in the gaze controller with and without interpolation. Notice ripple at the top of the sinusoid due to the highly non-linear variation of the target velocity that is not described by the constant acceleration model that is used for interpolation.

In the first control strategy the velocity command sent to the low-level loop is obtained by adding the filtered position error ($C(z)$ is a PD controller) with the velocity estimate. While the position component is fundamental to assure step disturbance rejection, velocity feedforward improves the transient response.
The system depicted in Fig. 3 is highly non-linear mainly due to the visual processing of information. Despite that, the graphic of Fig. 5 gives an approximation to the frequency response of the regulator in compensating the target 3D motion. It has been obtained by perturbing the system with sinusoidal target trajectories of different frequencies and measuring the amplitude and phase-lag in the trajectory described by the platform. To limit the non-linearities care must be taken in keeping the induced velocity in the image within the measurement range of the multi-resolution flow algorithm (±7pixels/frame that corresponds to ±38°/s). The closed loop system has a bandwidth of approximately 3Hz.

6. IMPLEMENTATIONS OF THE GAZE CONTROLLER USING MODEL PREDICTIVE TECHNIQUES

Fig. 2 shows that the low-level servo loop has a phase-lag. The main reasons for this phase-lag are the communication delays and the mechanical inertia

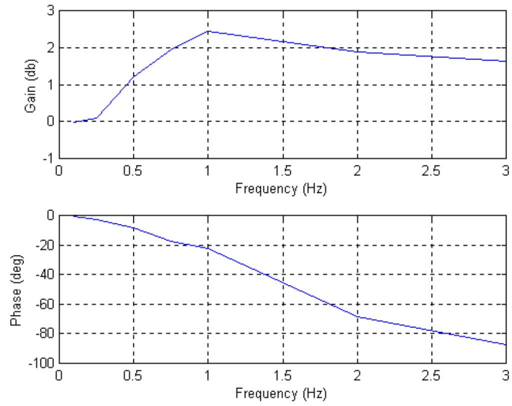

Figure 5: Smooth Pursuit Frequency Response.

(and in particular the static friction due to the harmonic drives gearheads). Standard system identification techniques can be used to obtain the transfer function of the low-level control loop. The input considered is the velocity command sent to the profile generator ($u(k)$) and the output is the motor velocity ($\phi_m(k)$). In the case of the system described in this paper the transfer function has a deadbeat of 2 sampling periods. It means that actuator/plant delay as seen by the gaze controller is nearly 12ms. This section discusses the use of model predictive controllers to cope with this delay.

$$J = \sum_{i=N_1}^{N_2} \left(y(n+i \mid n) - w(n+i) \right)^2 + \sum_{j=N_1}^{N_2} \lambda \Delta u(n+j-1)^2 \tag{1}$$

There are a wide variety of MPC algorithms, but they always have three elements in common: a prediction model, an objective function and a minimization process to obtain the control law. The prediction model is used to estimate the system output $y(n+k \mid n)$ at future time instants knowing previous inputs and outputs. The general aim is to make the future system outputs to converge to a desired reference $w(n)$. For that an objective function J is established. The general expression for such a function is given by equation (1). N_1 and N_2 bound the cost horizon, N_u is the control horizon, $u(n)$ is the control signal, $\Delta u(n)$ is the control increment ($\Delta u(n) = u(n)$ - $u(n-1)$) and λ is a relative weight used to adjust the smoothness of the control. In order to obtain present and future values of control law $u(n)$ the functional J is minimized.

The cost horizon is the future time interval where it is desirable for the output to follow the reference. This process has a dead time of 2. Therefore $N_1 = 2$ (the output can not be forced before that). Assuming a frame rate of 25Hz, the gaze controller (running at 166Hz) sends at most 7 velocity commands to the

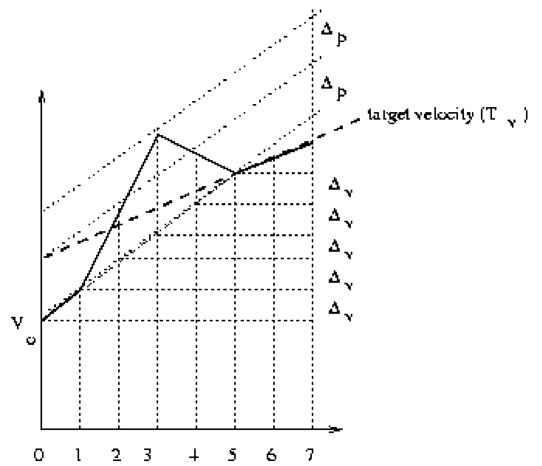

Figure 6: The Reference **w**. Target estimated velocity (--) and the desired convergence reference trajectory (-).

low-level loop without new visual information. Thus we are going to consider $N_1 = 8$.

Consider the step response $g(n)$ of a stable linear process without integrators. If $g(n) = 1$ for $n > N$ the system is completely described by knowing the N first instants of $g(n)$. This is the cornerstone for a simple, robust and intuitive model predictive controller: the dynamic matrix control algorithm.

$$\Delta \boldsymbol{u} = \left(\boldsymbol{G}\boldsymbol{G}^t + \lambda \boldsymbol{I} \right)^{-1} \boldsymbol{G}^t \left(\omega - \boldsymbol{f} \right) \qquad (2)$$

DMC uses the N first instants from step response to predict system output (in our case N=7). It assumes a constant disturbance along the cost horizon. The disturbance is given by the difference between the actual system output and the predicted output ($d(n) = y(n) - y(n \mid n)$). The goal of the controller is to drive the output as close as possible to the reference in the least-squares sense. The control action for that is computed by equation (2). $\boldsymbol{G}$ is the dynamic matrix of the system, $\Delta\boldsymbol{u}$ is the control vector and $\boldsymbol{\omega}$ is the reference vector. $\boldsymbol{f}$ is called the free response vector because it does not depend on the future control actions. Notice that only the first element of $\Delta\boldsymbol{u}$ is really sent to the motor. The vector is computed at each iteration to increase the robustness of the control to disturbances in the model. For more details on DMC controllers see [4].

$$\Delta_p = \frac{1}{6}\left(\frac{E_p}{T} - MV_0 \right) - \frac{\Delta_v}{3}\left(\frac{M^2}{4} + 2 \right) \qquad (3)$$

The goal of the DMC controller is to force the motor to have the same motion as the target in a near future. Whenever a new image is grabbed, visual processing is used to compute the target velocity and the tracking position error. Perfect tracking is achieved if, at the next frame time instant, the system compensates for the error in position and moves at the estimated velocity. This is the goal considered to establish the reference $\boldsymbol{\omega}$ whose profile is depicted in Fig. 6. Consider that P_0 and V_0 are the current motor position and velocity and that $P_t(i)$ and $V_t(i)$ are the target position and velocity at instant i. Then $\Delta_v = \left(V_t(M) - V_0\right)/M$ and Δ_p is computed by equation (3) where $E_p = P_t(M) - P_0$. M is the instant of convergence. By using M=5 the motor velocity converges to the target velocity in 5 sample intervals (30ms). In this time interval the motor accelerates and then slightly decelerates to compensate for the position error. The velocity reference $\boldsymbol{\omega}$ is computed for each control iteration.

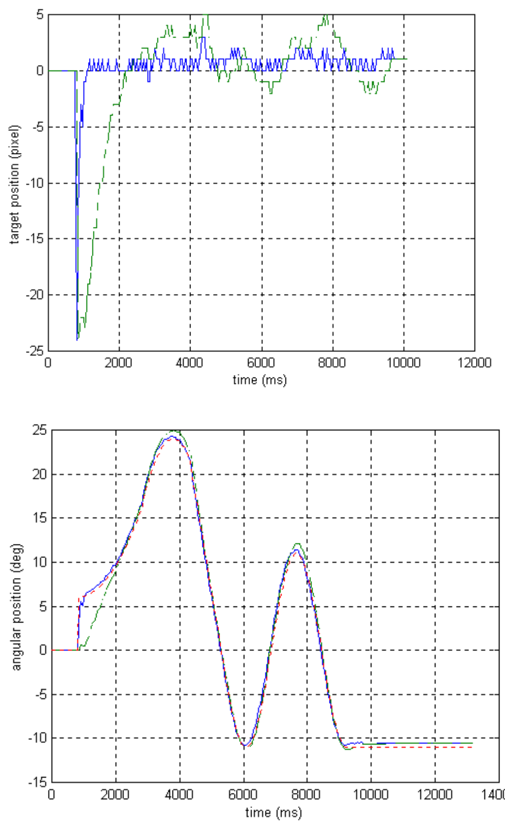

Figure 7: Tracking a target with a non-linear trajectory. Up: regulation performance in image. Target position in image for the first gaze controller (--) and for the DMC controller. Down: angular regulation in position. Target angular position (:) and motor position for the first controller (--) and using DMC (-).

The increase in performance due to the DMC controller can be observed in Fig 7. The error in position is immediately compensated and the target is kept in the center of the image during its non-linear motion.

7. CONCLUSIONS

In this paper the problem of system delays and latencies in visually guided systems has been analyzed. A flexible structure based in three independent control loops has been proposed: a low-level control loop running at 1KHz, a middle level loop running at 166Hz and a high-level visual processing loop running at 25Hz. Interpolation using a constant acceleration model of motion as been used to deal with the visual processing delay. A DMC controller as been specifically designed to cope with mechanical latency in visual control of motion tasks. The improvements in performance were evaluated using specific techniques where both visual processing and control were characterized. Bandwidths of the system were also estimated. Since the system is non-linear bandwidths for specific behaviors/ modes of operation were estimated and in particular for smooth pursuit.

8. REFERENCES

[1] Barreto, João P. and Paulo Peixoto and Jorge Batista and Helder Araujo, Evaluation of the Robustness of Visual Behaviors through Performance Characterization, Robust Vision for Vision-Based Control of Motion, Markus Vincze and Gregory D. Hager editors, IEEE Press, 1999.

[2] Barreto, João P. and Paulo Peixoto and Jorge Batista and Helder Araujo, Improving Active 3D Visual Tracking, Int. Conference on Vision Systems ICVS99, pp. 412-431, 1999.

[3] Brown, C., Gaze Controls with Interactions and Delays, IEEE Trans. on Systems, Man and Cybern., Vol. 20, No. 2, pp. 518-527, 1990.

[4] Camacho, E. F. and C. Bordons, Model Predictive Control, Springer-Verlag, 1999.

[5] Corke, P.I., Visual Control of Robot Manipulators-a review, Visual Servoing, K. Hashimoto editor, World Scientific,1993.

[6] Dickmanns, E. D., and B. Mysliwetz, and T. Christians, An Integrated Spatio-Temporal Approach to Automatic Visual Guidance of Autonomous Vehicles, IEEE Trans. on Systems, Man and Cybernetics, Vol. 20, No. 6, pp. 1273-1284, November/December, 1990.

[7] Hutchinson, S. and Hager, G. and Corke, P.I., A Tutorial on Visual Servo Control, IEEE Trans. on Robotics and Automation, Vol. 12, No. 5, pp. 651-670, October, 1996.

[8] Krautgartner, P and Markus Vincze, Performance Evaluation of Vision-Based Control Tasks, IEEE International Conference on Robotics and Automation, Leuven, Belgium,1998

[9] Pahlavan, K., and T. Uhlin, and J. Ekhlund, Integrating Primary Ocular Processes, Proc. 2nd European Conf. on Computer Vision, pp. 526-541, 1994.

[10] Sharkey, P. M., and D. W. Murray, Delay Versus Performance of Visually Guided Systems, IEE Proc.-Control Theory Appl., Vol. 143, No. 5, pp. 436-447, 1996.

[11] Sharkey, P. M., Bandwidth Issues for Active Vision Systems, Proceedings of the International Conference on Mechatronics and Machine Vision in Practice, Hong Kong, 1995.

[12] Sharkey, P. M. and Murray, D. W. and McLauchlan, P. F. and Brooker, J. P., Hardware Development of the Yorick Series of Active Vision Systems, Microprocessors and Microsystems, Vol. 21, pp. 363-375, 1998.