Integrating Vision and Control to Achieve High Performance Active Tracking

João P. Barreto, Paulo Peixoto, Jorge Batista, Helder Araujo

Institute of Systems and Robotics
Dept. of Electrical and Computer Engineering
University of Coimbra
Coimbra, PORTUGAL

Abstract

To achieve high performance visual tracking a well balanced integration of vision and control is particularly important. This paper describes an approach for the improvement of tracking performance by a careful design and integration of visual processing routines and control architecture and algorithms. A layered architecture is described. In addition a new approach for the characterization of the performance of active vision systems is described. This approach combines the online generation of test images with the real time response of the mechanical system. Crucial for the performance improvement was the consideration that the number of mechanical degrees of freedom used to perform the tracking was smaller than the number of degrees of freedom of rigid motion (6). The improvements on the global performance are experimentally proved.

1 Introduction

The approach of controlling robot motion using visual information is referred to in the literature as visual servoing. Vision provides non contact measurements of the world, extending the robot ability to operate in unknown circumstances and environments. However the use of visual feedback can be a very challenging problem. Not only because image processing is far from being a trivial subject, but also because there are a lot of additional issues and trade-offs that must be considered. The design of a visual servoing system involves complex topics on computer vision, real time computing, control theory and system architecture.

Control issues are particularly important. The visual processing delay in the feedback loop is a major problem for the global system dynamics. In general the visual latency introduces a phase delay in the system frequency response. In extreme situations the phase delay can lead to unstable systems [1], and very often it is responsible for poor transitory behaviors [2]. As referred in [3, 4] there are several reports of visual servoing systems with a slightly oscillatory behavior or a very slow response. In the former the closed loop system is near the stability limit. In the latter the latency is not compensated by an adequate controller design/tuning. This illustrates the importance of dynamics to achieve high performance and robust visual control of motion [4, 5, 6, 7].

Active visual tracking of moving targets consists in having one or more cameras mounted on a moving platform whose motion is controlled such that the target projection is kept in a certain position in the image plane (usually the center). The moving platform can be a mobile robot, a manipulator end-effector or a robot head as the ones described in [8, 9, 10]. In [11, 12, 13] Batista et al. describe a system to perform binocular tracking using the ISR-MDOF robot head [8]. The proposed system is the starting point of this work which aims to modify the original design to achieve higher performances. Due to the high coupling between vision and control, optimizing visual servoing performance is not a trivial problem. There are several trade-offs that must be taken into account. For instance, more accurate and robust visual measurements do not necessarily lead to improved active tracking performances. If the modifications in the visual feedback loop imply an increased latency then the negative effects in system relative stability can overtake the advantages provided by the visual algorithms. Improvements in visual processing must be complemented by adjustments in the controller tuning and design. On the other hand changes on the control architecture to increase closed loop bandwidth affect the stability of image capture. This can deteriorate the visual feedback and decrease the global system performance.

The article starts by introducing the ISR-MDOF binocular tracking system. The system is modelled as a regulation control problem with both input and output defined in the image plane and target motion acting as a perturbation that must be compensated [3, 4]. High performance visual tracking can only be achieved with a well balanced integration of vision and controls. Good choices and decisions can only be made if both aspects can be characterized in a common

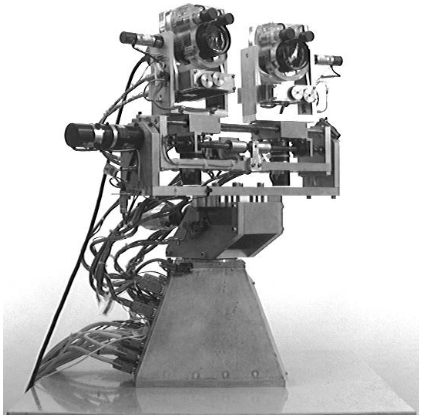

Figure 1: The ISR-MDOF robot head.

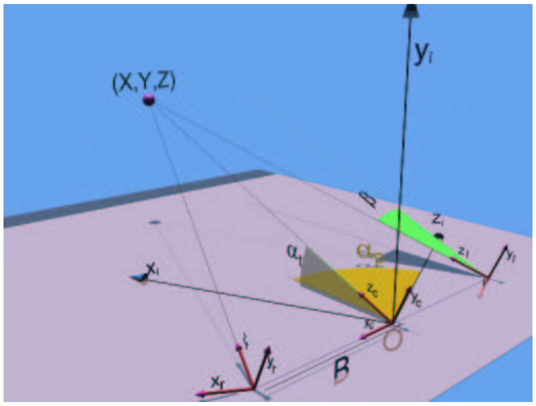

Figure 2: ISR-MDOF binocular fixation

framework. A metric for global performance evaluation is presented and discussed [14, 15]. The proposed evaluation framework is applied to the ISR-MDOF tracking system. The experimental measurements are used as guidelines in successive iterations where system design is modified and tracking performance is improved. Solutions for different aspects of visual control of motion are proposed and trade-offs between vision and control are illustrated by the experiments. A method for real time position and velocity estimation in the image is presented [16]. Kalman filtering of visual measurements is discussed for both monocular tracking and vergence control. A system architecture based on three concurrent processes is proposed (visual processing, gaze control and low-level servo control) [17]. Linear interpolation is used to cope with the visual processing delay [2, 18]. Model predictive control is applied in the gaze control [19]. A final demo showing system performance for both monocular and binocular tracking is provided in media content.

2 The ISR-MDOF Binocular Tracking System

Fig. 1 depicts the ISR-MDOF Binocular Tracking System. The MDOF robot head is a high performance 16 degrees of freedom (DOF) mechanical system. Each degree of freedom is actuated by a DC motor with “harmonic drive” gears which allows excellent positioning accuracy and repeatability with zero backlash. Each MDOF joint can be controlled both in position and in velocity. The position control is performed by a local closed loop with a PID digital filter running at 1KHz. The feedback position information is provided by an optical encoder mounted on motor back shaft.The velocity control is achieved using a profile generator that integrates the velocity command and generates the position reference for the closed loop. The communication with the servo control modules is synchronous at a rate of 166Hz. For further details about the MDOF robot head see [8, 20]. In [11, 12], Batista et al. propose an active binocular tracking algorithm using the ISR-MDOF platform. The algorithm is the starting point of our work.

Consider a target moving in 3D space. The goal of the binocular tracking algorithm is to control platform motion such that target image is kept in the fovea region of both retinas. The smooth pursuit behavior proposed in [11, 12] uses 4 DOF of the robotic head: neck pan and tilt and both eyes pan motion. Neck pan and tilt angles are respectively α_p and α_t. The pan motion of the eyes is known as vergence motion. Angles β_l and β_r are respectively the left and right vergence. The proposed binocular tracking strategy constrains the left and right vergence angles to be symmetric ($\beta_l = -\beta_r = \beta$). Fig. 2 depicts the described fixation geometry.

Assume a virtual eye, called cyclopean eye, positioned in the middle of left and right camera. The neck rotation angles α_p and α_t are controlled to align the cyclopean optical axis with the target in 3D space. Whenever the axis is aligned with the object the target image is projected in the center of the virtual cyclopean retina. Neck rotation control is similar to the control of the monocular active tracking of the moving target with the cyclopean eye. The vergence degree of freedom is used to adjust left and right camera position such that the target is projected in the center of both images. By assuming a symmetric vergence configuration the angle β only depends on the target 3D motion along cyclopean optical axis. Thus the vergence angle is a function of target depth in the cyclopean referential frame. This suggests that by assuming symmetric vergence the binocular tracking can be decoupled in two sub-problems: pan and tilt control of the cyclopean eye and vergence control.

This section shows how the binocular tracking problem can be split into monocular active tracking with the cyclopean eye and vergence control. The next subsection focuses on the visual processing of left and right images and the conversion of the obtained measurements to the virtual cyclopean eye.

2.1 Converting Visual Information to the Cyclopean Eye

$$\dot{\boldsymbol{x}}_{\boldsymbol{i}} = \dot{\boldsymbol{x}} + \dot{\boldsymbol{x}}_{\boldsymbol{ego}} \tag{1}$$

Consider the position $\mathbf{x_i}$ of a moving target in the image plane of a moving camera. The motion in the image depends both on the target motion in space and on the camera motion. The former is called target induced motion and the latter self-induced motion or egomotion. Equation 1 shows the target velocity in the image $\dot{\mathbf{x}}_\mathbf{i}$ as sum of two velocity components: the velocity induced by object independent motion $\dot{\mathbf{x}}$, and the egomotion velocity $\dot{\mathbf{x}}_\mathbf{ego}$. If the camera undergoes pure rotation then $\dot{\mathbf{x}}_\mathbf{ego}$ depends only on the sensor motion. In this case a pair of images of a static scene are related by an homography [21]. If the camera describes a motion with a translation component then $\dot{\mathbf{x}}_\mathbf{ego}$ depends both on the sensor motion and on the distance between the 3D imaged points and the camera. Fig.2 shows that there is an arm B between the cameras and neck joints which introduces a translational component of motion whenever neck rotates in pan. Nevertheless if the distance of the 3D imaged points to the system is much larger than the baseline length $2B$ then the egomotion image velocity due to translation can be neglected [22]. The assumption of a distance between the scene and the system much larger than the baseline holds in the remaining of the section.

The tracking algorithm presented in [11, 12, 13] integrates simultaneously position and velocity information to control platform motion. The target induced velocity $\dot{\mathbf{x}}$ is measured after an explicit compensation of the egomotion component. Consider two frames captured in successive time instants. Relative camera motion is determined using encoder information and self-induced image velocity is estimated. Given $\dot{\mathbf{x}}_\mathbf{ego}$, the egomotion compensation is performed by warping the first frame. The warped frame is subtracted to the second frame and a difference image is obtained. This last image contains the points where independent motion occurred. Assuming that there is a single moving object in the scene then the independent motion points must belong to the target. The target image is segmented and the corresponding position is measured as the average location of the difference points. Target induced velocity $\dot{\mathbf{x}}$ is estimated by computing the optical flow in these points (a constant velocity model is assumed).

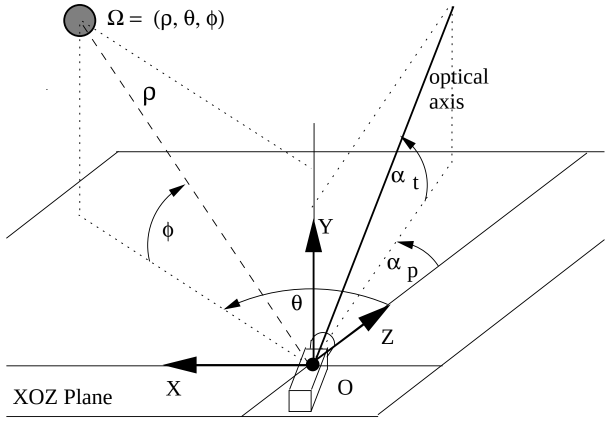

Figure 3: Camera mounted on a pan and tilt unit.

$$\boldsymbol{x_i} = \left[\begin{array}{c} \frac{x_i^l + x_i^r}{2\cos(\beta)^2(1-(x_i^l - x_i^r)\tan(\beta) - x_i^l x_i^r \tan(\beta)^2)} \\ \frac{y_i^l + y_i^r + (y_i^l x_i^r - y_i^r x_i^l)\tan(\beta)}{2\cos(\beta)(1-(x_i^l - x_i^r)\tan(\beta) - x_i^l x_i^r \tan(\beta)^2)} \end{array} \right] \tag{2}$$

$$\dot{\boldsymbol{x}} \approx \frac{\dot{\boldsymbol{x}}^{\boldsymbol{l}} + \dot{\boldsymbol{x}}^{\boldsymbol{r}}}{2} \tag{3}$$

The target position coordinates in the left and right images are respectively $\mathbf{x_i^l} = (x_i^l, y_i^l)^t$ and $\mathbf{x_i^r} = (x_i^r, y_i^r)^t$ (the focal length f is made unitary). Consider that the target would be projected in the virtual cyclopean camera at position $\mathbf{x_i} = (x_i, y_i)^t$. Equation 2 computes cyclopean coordinates $\mathbf{x_i}$ as a function of $\mathbf{x_i^l}$ and $\mathbf{x_i^r}$ and the vergence angle β. Notice that if the distance between the target and the robotic head is much larger than the baseline then it is reasonable to assume that the vergence angle is small. By making $\beta \approx 0$ equation 2 becomes $\mathbf{x_i} \approx (\mathbf{x_i^l} + \mathbf{x_i^r})/2$. Moreover given target induced velocities $\dot{\mathbf{x}}^\mathbf{l}$ and $\dot{\mathbf{x}}^\mathbf{r}$ in left and right retinas it can be shown that the velocity induced in the cyclopean image would be approximately given by equation 3. The mathematical details and the analysis of the conditions for the application of the approximations are provided in [22].

2.2 Control Equations

This subsection introduces the kinematic equations for both monocular tracking and vergence control.

2.2.1 Monocular Tracking

$$\boldsymbol{x_i} = \left[\begin{array}{c} f\cos(\alpha_t)\tan(\delta_p) \\ f\tan(\delta_t) \end{array} \right] \tag{4}$$

Fig.3 depicts a perspective camera mounted on a platform with two rotative degrees of freedom: pan and tilt. The

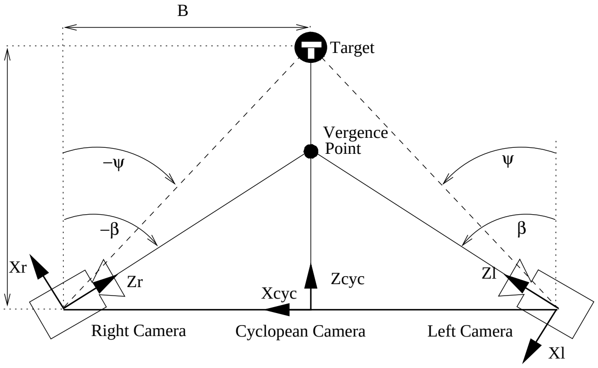

Figure 4: Top view of the binocular system. Vergence control

goal of the monocular tracking algorithm is to control platform pan and tilt angles (α_p and α_t) such that camera optical axis becomes aligned with the moving target. Consider that $\mathbf{\Omega} = (\rho, \theta, \phi)$ is target 3D position in spherical coordinates. Pan and tilt angular control errors are $\delta_p = \theta - \alpha_p$ and $\delta_t = \phi - \alpha_t$. Equation 4 establishes the relationship between the target position in the image $\mathbf{x_i} = (x_i, y_i)^t$ and the angular control errors. It assumes a perspective model of image formation (f is the focal length) and is valid for cameras with a small field of view [22]. Notice that the pan and tilt control are decoupled. The angular pan error δ_p is related to the target X coordinate in the image plane, and the tilt error δ_t depends only on the target Y coordinate. Without loss of generality we will focus our study in the pan control, knowing that all conclusions will be applicable to the tilt control.

$$x_i = f \tan(\delta_p) \tag{5}$$

$$\dot{x}_i = f \sec^2(\delta_p)\dot{\theta} - f \sec^2(\delta_p)\dot{\alpha}_p \tag{6}$$

Consider that the target ϕ coordinate is zero as well as the camera tilt angle α_t. The object is moving in the XOZ plane of Fig.3 and its image is projected in the X axis of the image. The goal is to control the camera pan motion such that the target is projected in the origin of the image plane. Equation 5 establishes the relation between the target image X coordinate and the angular control error δ_p. Equation 6 computes the target image X velocity as a function of the target angular velocity $\dot{\theta}$ and the camera pan velocity $\dot{\alpha}_p$. The first term on the right member is the velocity $\dot{x}$ induced by target motion. The second term is the egomotion velocity component $\dot{x}_{ego}$.

2.2.2 Vergence Control

Assume that the target is aligned with the cyclopean optical axis. The goal of the vergence control is to correct the left and right camera positions such that the target is projected in the center of both image planes. The 3D point where the left and right optical axes intersect is called vergence point. Thus the goal is to make the vergence point coincident with the target.

$$\Delta x_i = 2f \frac{B - \rho \tan(\beta)}{\rho + B \tan(\beta)} \tag{7}$$

$$\Delta x_i = 2f \tan(\delta_v) \tag{8}$$

Consider the disparity $\Delta x_i = x_i^l - x_i^r$ in the target X position between the left and right images. Equation 7 establishes the relation between the image disparity Δx_i, vergence angle β and the target position in the cyclopean optical axis ρ. Its derivation is a mere calculus exercise considering the target position in each camera coordinate frame. If the vergence angle is $\psi = \arctan(B/\rho)$ then the vergence point coincides with the target and the disparity Δx_i is zero (see Fig.4). The disparity Δx_i depends on the the angular vergence error $\delta_v = \psi - \beta$ as shown in equation 8 directly derived from 7.

$$\Delta \dot{x}_i = -\frac{2f(1 + \tan(\beta)^2)}{(\rho + B\tan(\beta))^2}(B\dot{\rho} + (\rho^2 + B^2)\dot{\beta}) \tag{9}$$

$$\Delta \dot{x}_i = -\frac{2fB}{\rho^2 + B^2}\dot{\rho} - 2f\dot{\beta} \tag{10}$$

Equation 9 is derived by differentiating equation 7, with $\Delta \dot{x}_i = \dot{x}_i^l - \dot{x}_i^r$ the velocity disparity between left and right image, $\dot{\beta}$ the angular velocity of vergence motion and $\dot{\rho}$ the target velocity along the cyclopean axis. If the vergence point is coincident with the target then $\delta_v = 0$ and $\tan(\beta) = B/\rho$. Equation 9 becomes equation 10. Notice that once again the first term is the target induced disparity velocity and the second term the egomotion velocity.

2.3 Active Tracking as a Regulation Control Problem

Active tracking of moving targets is a problem of visual control of motion. In this situation the system to be controlled is the robotic platform supporting the cameras. There are a wide variety of visual servoing applications in the literature. One typical application is to control the position of a robotic manipulator using visual feedback [23, 24, 25]. The proposed problem solutions can be classified into two major groups: position based and image based visual servoing [26]. In the former the control input is defined in the 3D space. Typically object pose is explicitly estimated knowing camera calibration [24]. In the latter the task function is defined in the image plane [25]. The image

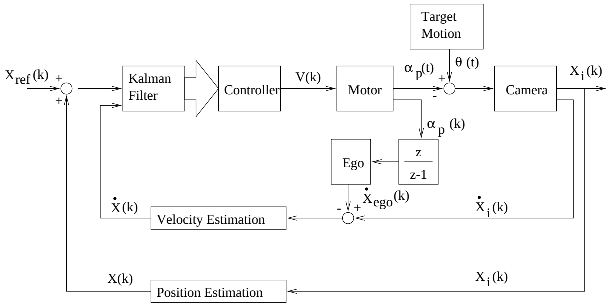

Figure 5: Block diagram for the monocular tracking pan degree of freedom. θ and α_p are target and camera angular positions. x_i and $\dot{x}_i$ are target image position and velocity. $\dot{x}_{ego}$ is the image velocity induced by camera motion (egomotion). x is the estimated target position in image (ideally $x = x_i$, and $\dot{x}$ the image velocity induced by target motion after egomotion compensation.

based approach is less sensitive to calibration errors, nevertheless it can present singularities and convergence problems [27]

In a similar manner we can have a position or an image based approach for active visual tracking. The former is a servo control problem while the latter is a regulation control problem. Consider the vergence control introduced previously. In the position based approach the reference input is the 3D target position and the controlled output the 3D position of the vergence point. The system is modelled as a servo control loop where the goal is to make the output converge to the input reference. In an image based approach the task function is defined in the image plane. The control goal is to keep a null target image disparity between the left and right retinas. The system is modelled as a regulator where the 3D target motion acts as a perturbation that must be compensated.

In [3, 4] Corke and Good model the active visual tracking system as a regulation control problem. The reference input is the desired target position in image (typically the center) and the output the actual target position. The target motion in 3D space is a perturbation that must be compensated by the robot motion to keep the target projection in the desired reference position. The performance of the system is measured by its perturbation rejection ability. This image based approach is not so dependent on the camera and system mechanical calibration, and problems of convergence and due to singularities do not arise due to the fact that the target is modelled as a moving point.

2.3.1 Monocular Tracking

Fig. 5 depicts the monocular tracking pan control loop. The input reference $x_{ref}(k)$ is the desired target position in image (typically $x_{ref} = 0$), and the control output the actual target position $x_i(k)$. The target motion in the scene acts as a perturbation that must be compensated for, such that the target projection is kept in the image center. The regulation is achieved by moving the platform in a way that $\delta_p = \theta - \alpha_p$ converges to zero. For this purpose both the target image position $x(k)$ and velocity $\dot{x}(k)$ are estimated after egomotion compensation. Notice that ideally measurement results $x(k) = x_i(k)$ and $\dot{x}(k) = \dot{x}_i(k) - \dot{x}_{ego}(k)$. In this particular case the camera undergoes a pure rotation motion. There is an homographic relation between two successive frames [21]. The egomotion velocity $\dot{x}_{ego}$ is estimated from motor encoder readings.

The target position in the image is directly related to the angular pan error. From equation 5 $\alpha_p = \arctan(x/f)$. The estimated velocity $\dot{x}$ corresponds to the first term of the right member of equation 6. Camera pan velocity $\dot{\alpha}_p$ must be such that the egomotion velocity cancels out the object induced velocity and the target velocity in image converges to zero ($\dot{x}_i = \dot{x} + \dot{x}_{ego} = 0$). From equation 1, 5 and 6 it results that $\dot{\alpha}_p = \dot{\theta} = f\dot{x}/(f^2 + x^2)$ which can be approximated to $\dot{\alpha}_p \approx \dot{x}/f$ assuming $x \approx 0$. The velocity command $V(k)$ sent to the motor velocity control loop is computed according to equation 11 where G is a constant gain. A Kalman filter is used to limit the effects of both noise and errors in the visual measurements allowing a smooth tracking behavior.

$$V(k) = \frac{1}{f}(\dot{x}(k) + G.x(k)) \tag{11}$$

2.3.2 Vergence Control

As discussed in section 2.2.2 symmetric vergence control is achieved using the image disparity information. Both the position disparity Δx and the velocity disparity induced by target motion $\Delta\dot{x}$ are estimated (see Fig.6). The angular position error δ_v can be determined knowing Δx (equation 8). The velocity disparity $\Delta\dot{x}$ after egomotion compensation corresponds to the first term of the right member of equation 10. Angular vergence velocity $\dot{\beta}$ must be such that image velocity disparity $\Delta\dot{x}_i$ converges to zero and regulation is achieved. Repeating the reasoning for the monocular tracking problem it results from equation 10 that $\dot{\beta} = \Delta\dot{x}/2f$ [11, 13].

3 Defining the Test Signals

As discussed in 1 visual control of motion depends both on vision and controls. To achieve high performance visual servoing the integration of these two aspects must be taken into account. Nevertheless a well balanced solution can be difficult to establish. The characterization of vision and

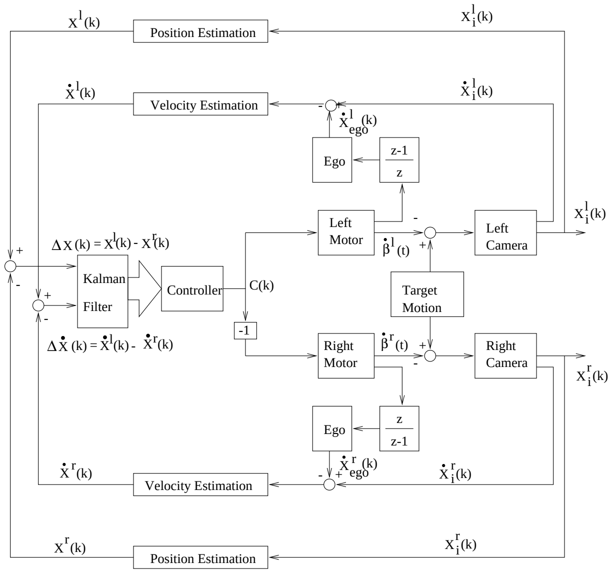

Figure 6: Block diagram for symmetric vergence control. The control is made using position and velocity disparity between left and right images. β^l and β^r are left and right vergence angles which are symmetric.

control aspects must be performed simultaneously within a common framework. In linear control theory there are a set of test signals used in performance evaluation. These signals are the step, the ramp, the parabola and the sinusoid which allow the characterization of the different system aspects. Our common evaluation framework is based on the standard linear control test signals.

Consider the regulation control loops depicted in Fig. 5 and 6. The control test signals can be defined in the 3D target motion space. A ramp perturbation input would correspond to a constant velocity motion of the target in the scene. However this definition of the perturbation test signals is not coherent with our image based approach to the visual servoing problem. The input and output of the regulation control loops are defined in the image plane and not in the 3D space. Moreover both monocular tracking and the vergence control are not linear systems. The visual processing introduces non linearities in the feedback loop. In general a ramp defined in the 3D target motion space does not induce a constant velocity in image. The control test signals defined in this way do not make sense in the evaluation of the visual components (position and velocity estimation).

The test signals are defined in the image plane within the framework of the image based approach. A step perturbation is an abrupt change of target image position. A ramp/parabola occurs when the target motion in space induces a constant velocity/acceleration in the image plane. A sinusoid is generated whenever the image target velocity and position are described by sinusoidal function of time. The test signals are meaningful for the visual estimation algorithms allowing an evaluation of both vision and controls in a common framework.

Two problems remain:

- To accurately generate the perturbation test signals;
- To determine the target 3D trajectories that induce a step, ramp, parabola and sinusoid in the image plane for both monocular tracking and vergence control;

3.1 Accurate Generation of the Test Signals

To characterize the system ability to reject the perturbation due to target motion, specific test signals have to be generated. Our methodology assumes a virtual target moving along a specified 3D space trajectory. The virtual object is projected in the image plane at each frame time instant. Image generation depends both on 3D target position and camera pose at that instant. The generated images are used as inputs of the visual feedback loop. System behavior to compensate for the perturbation is observed and the regulation performance is measured.

Given a predefined target motion, captured frames will depend, not only on the target position, but also on the camera orientation. System geometry changes along time as a result of the tracking behavior. Thus, images have to be generated on line to take into account the specific geometry at each time instant. Therefore at each frame time instant both the target position and the camera orientation have to be known in the same inertial coordinate frame. The former is calculated using the specific motion model of the virtual target. Camera orientation is computed by taking into account the motor encoders readings and head inverse kinematics. The inertial coordinate frame origin is placed at camera optical center (monocular tracking) or at the origin of the cyclopean referential (vergence control).

The proposed emulation technique makes possible an accurate generation of the test perturbation signals. The trajectory of the virtual target is defined mathematically. The knowledge of the input signal used to excite the system is essential for a rigorous performance characterization. Moreover the effort to perform experiments is minimized and an excellent experimental repeatability is assured.

3.2 Reference Trajectories. Monocular Tracking

A method for accurate generation of perturbation test signals has been derived. This section discusses the 3D target trajectories that correspond to an image step, ramp, parabola and sinusoid for the monocular tracking system.

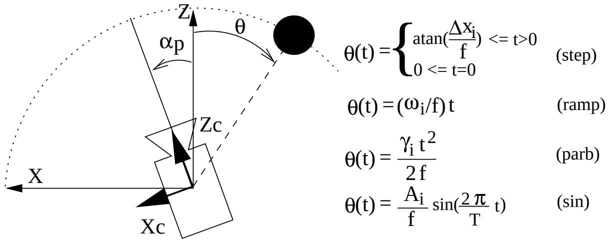

Figure 7: Test signals for the monocular tracking. The target moves along a circular path centered in the camera optical center. θ and α_p are respectively target and camera angular positions.

A step perturbation is an abrupt change of the target image position x_i (equation 5). If the camera is static, any sudden change of the object 3D position generates the desired effect in the image plane. The only exception occurs if the target moves along the projection ray. In this case the target image will remain the same. Consider that the target 3D position is given in spherical coordinates as shown in Fig.7. From the equation 5 it results that a step Δx_i in the image plane corresponds to a step $\Delta\theta = \arctan(\Delta x_i/f)$ in the target 3D angular position.

The target velocity in the image $\dot{x_i}$ depends both on the target induced velocity $\dot{x}$ and the egomotion $\dot{x}_{ego}$ (equations 1 and 6). For a perfect tracking situation the former is compensated by the latter and no motion is detected in the image. Whenever perfect tracking does not happen there will be image motion as a result of the tracking error. Therefore, the objective of tracking is to move the camera in such a way that the egomotion compensates for the motion induced in the image by the target. From this point of view the system perturbation will be the motion induced by the target.

$$\omega_i = f\dot{\theta}\sec(\delta_p)^2 \tag{12}$$

A ramp perturbation occurs whenever the target motion induces a constant velocity ω_i in the image plane ($\dot{x} = \omega_i$). This 3D motion can be computed by solving differential equation 12, derived from 6, in order to θ. The difficulty is that the reference trajectory $\theta(t)$ will depend on the system response to the perturbation. To induce a constant velocity in image during operation, the target 3D angular velocity must be computed at each frame time instant as a function of the the tracking position error δ_p. If perfect tracking occurs then $\delta_p = 0$ and an object motion along a circular path with constant angular velocity $\dot{\theta} = \omega_i/f$ induces the desired ramp perturbation (Fig.7). During smooth pursuit operation the target image usually lies near the center and the position tracking error is small. Thus, if the tracking is not perfect then the uniform angular motion does not induce a constant image velocity. However the variation on $\dot{x}$ tends to be negligible. We will assume a null tracking error to derive a ramp reference trajectory that does not depend on the system response to the perturbation.

$$\gamma_i t = f\dot{\theta}\sec(\delta_p)^2 \tag{13}$$

$$A_i\frac{2\pi}{T}\cos(\frac{2\pi}{T}t) = f\dot{\theta}\sec(\delta_p)^2 \tag{14}$$

The parabola perturbation occurs whenever target 3D motion induces a constant acceleration γ_i in the image and the velocity $\dot{x}$ is given by $\dot{x}(t) = \gamma_i t$. In a similar way the a sinusoid perturbation with amplitude A_i and period T corresponds to a co-sinusoidal variation of $\dot{x}$ with amplitude $(2\pi A_i)/T$ and equal period. The reference trajectories that generate parabola and sinusoid perturbations can be computed by solving equations 13 and 14. The null tracking error assumption is used to avoid that the derived test signals depend on the system response to the perturbation.

Fig. 7 summarizes the reference trajectories used for monocular tracking characterization. The object moves in a circular path centered in the camera optical center. An abrupt change on the angular position generates a step input in the image. A perturbation in ramp/parabola corresponds to a motion with constant angular velocity/acceleration. The sinusoid is induced by a sinusoidal motion of the object on the circular path.

3.3 Reference Trajectories. Vergence Control

Fig. 4 depicts the vergence control scheme. It is assumed that target moves along the cyclopean optical axis with position ρ. The vergence is symmetric with angle β. Vergence control is achieved using the image disparity between left and right retinas (equations 7 to 10). We want to derive the reference trajectories $\rho(t)$ that generate the desired perturbation test signals in the image plane.

$$2fB\dot{\rho} + v_i\rho^2 = -v_iB^2 \tag{15}$$

$$2fB\dot{\rho} + a_it\rho^2 = -a_itB^2 \tag{16}$$

$$2fB\dot{\rho} + A_i\frac{2\pi}{T}\cos(\frac{2\pi}{T}t)\rho^2 = -A_i\frac{2\pi}{T}\cos(\frac{2\pi}{T}t)B^2 \tag{17}$$

For the vergence control system a step perturbation is an abrupt change on the position disparity which is induced by a sudden change on target position along the cyclopean axis (equations 7 and 8). A ramp input occurs whenever the disparity of the target induced velocity in image $\Delta\dot{x}$ is constant along time. Assuming that the vergence point is coincident with the target ($\Delta x_i = 0$), it is obtained from equation 10 that $\Delta\dot{x} = -(2fB\dot{\rho})/(\rho^2 + B^2)$. The target 3D trajectory $\rho(t)$ such that $\Delta\dot{x} = v_i$ can be computed by solving the differential equation 15. Reasoning in a similar way to the monocular tracking problem, the reference trajectories for

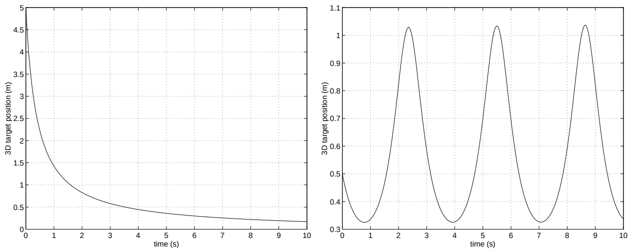

Figure 8: Left: Ramp perturbation. Reference trajectory inducing a constant velocity disparity $v_i = 1$(pixel/frame) ($\rho(0) = 5$(m)). Right: Sinusoidal Perturbation. Reference trajectory that generates a sinusoidal disparity with $A_i = 2$(pixel) and $T = \pi$(s). ($\rho(0) = 1$(m))

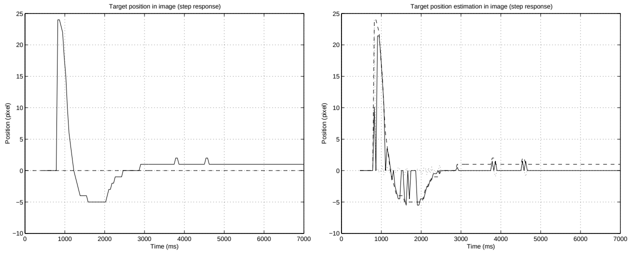

Figure 9: System response to a step perturbation of amplitude $\Delta x_i = 24$ pixels (the angular amplitude is $\Delta\theta = 6°$). Left: Regulation in image (x_i [–]). Right: Image position estimation (x_i [- -]; x [–])

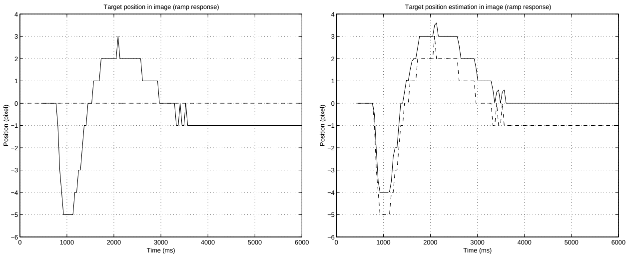

Figure 10: System response to a ramp perturbation of amplitude $\omega_i = -1.6$ pixels/frame (the angular velocity is $\dot{\theta} = -10°$/s). Left: Regulation in image (x_i [–]). Right: Image position estimation (x_i [- -]; x [–])

the parabola and sinusoid perturbations can be determined from equations 16 and 17 (a_i is the parabola constant acceleration and A_i and T the sinusoid amplitude and period).

Test signals obtained by solving the differential equations 15 and 17 are depicted in Fig.8. Notice that to induce a constant velocity disparity in the images the 3D target velocity increases with depth. This is due to the perspective projection.

4 System Performance Evaluation

The reference trajectories derived in the section 3.2 are used to evaluate the performance of the monocular tracking system depicted in Fig. 5.

4.1 Step Response

A step in position is applied to the system. Fig. 9(L) shows the evolution of the target position (x_i) in the image. The system regulation response presents an overshoot of about 5 pixels and a steady state error of 1 pixel. It is a typical second order step response of a type 0 system. However in experiments with smaller amplitude steps the system fully compensates for the perturbation induced by target motion. In these situations the regulation steady state error is 0 and we have a type 1 system. The response type depends on the step amplitude which clearly indicates a non-linear behavior.

Fig. 9(R) shows the performance of the image position estimator in the visual feedback loop. The target position is estimated as the average location of the set of points with non-zero optical flow in two consecutive frames after egomotion compensation. The estimated value x is the center of motion instead of the target image position x_i. Notice in Fig. 9(R) that at the step transition instant x_i changes from 0 to 24 pixels and the corresponding estimation x is about 10 pixels. For smaller step amplitudes the regulation response improves because the center of motion in the transition instant is closer to the target image position x_i and the estimation x becomes more accurate.

If the object is stopped in the scene then the induced velocity $\dot{x}$ is zero and any image displacement is due to the self induced component $\dot{x}_{ego}$ (equation 1). After egomotion compensation, no motion is detected in the image and the position estimation x is null. For a step perturbation there is only independent motion on the transition instant. Therefore target position would only be estimated at the step transition time instant. However this is not the case as shown in Fig. 9(R). Only with egomotion as a pure rotation would this occur. In practice sampling and misalignment errors between the rotation axis and the center of projection introduce small errors. The egomotion compensation is not perfect and a few points appear with non-zero optical flow. The target position is estimated as the average location of this set of points.

4.2 Ramp Response

Fig.10 exhibits the system response to a ramp perturbation. The target moves about 5 pixels off the center of image be-

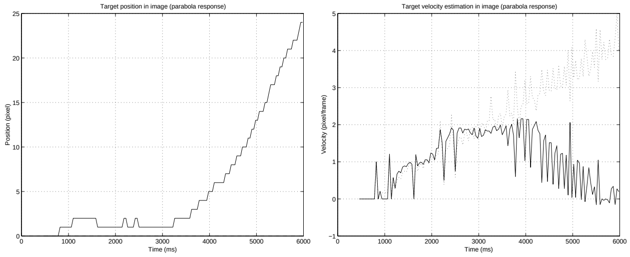

Figure 11: System response to a parabola perturbation of amplitude $\gamma_i = 0.032$ pixels/frame^2 (the angular acceleration is $\dot{\theta} = 5°/s^2$) Left: Regulation in image (x_i [–]). Right: Estimation of image velocity induced by target motion ($\dot{x}_i - \dot{x}_{ego}$ [...]; $\dot{x}$ [–])

fore the system starts to reject the perturbation. It clearly presents an initial inertia due to the visual processing delay and the Kalman filter action (see Fig. 5). The regulation loop behaves like a type 1 system with a steady state error of -1 pixel. This is due to the deficient estimation x that yields the center of motion instead of the target position x_i (Fig. 10(R)). For a ramp perturbation the estimation error is about half of the induced velocity $\dot{x}$. In Fig. 10(R) $\dot{x} = -1.6$ pixels/frame, thus the estimation error should be around 0.8 pixels, which is in accordance with the observed steady state error. Additional experiments showed that the system presents a reasonable performance for ramp inputs with angular amplitude up to $20°$/s.

4.3 Parabola Response

Fig. 11 shows the regulation performance in rejecting a parabola perturbation of amplitude $\gamma_i = 0.032$ pixels/frame^2. After a certain point the system is clearly unable to track the moving target. The algorithm for velocity estimation using optical flow only performs well for velocities up to 2 pixels/frame. When the velocity $\dot{x}$ increases beyond certain values, flow underestimation bounds the global regulation performance. The system becomes unable to follow the object and compensate for its velocity. As a consequence the object image is increasingly off the image center. The limitations on velocity measurement are due to the mask size used to compute the gradient brightness in the flow estimation. We will overcome this difficulty using a multi scalar approach.

4.4 Sinusoid Response

The system response for a sinusoid input shown in Fig. 4 confirms the previous conclusions. Velocities beyond 2 pixels/frame are underestimated and the global regulation performance is clearly affected (Fig. 4(TL) and (BL)). However the errors in velocity measurement are partially compensated for by the position component on the motor command $V(k)$ (equation 11) and by the Kalman filter action (Fig. 4(BR)). The Kalman filtering limits the effect of measurement errors and allows smooth motion without oscillations. The phase lag observable in the angular position response (Fig. 4(TR)) is mainly caused by the inertia introduced by Kalman filter smoothing action.

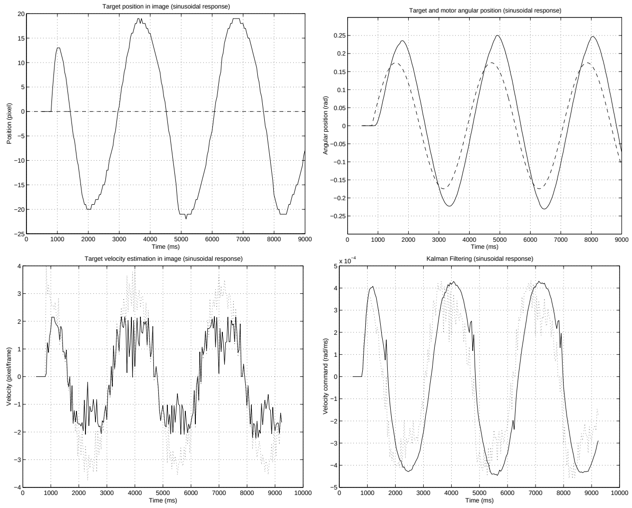

Figure 12: System response to a sinusoid perturbation of amplitude $A_i = 40$ pixels and period $T = \pi$s (the angular amplitude is $A = 10°$). Top Left: Regulation in image (x_i [–]). Top Right: Angular position response (α_p [–]; θ [- -]). Bottom Left: Estimation of image velocity induced by target motion ($\dot{x}_i - \dot{x}_{ego}$ [...]; $\dot{x}$ [–]). Bottom Right: Kalman filering of the command V(k) (input [...]; output [–])

5 Improving Visual Feedback

The characterization of the monocular tracking system (Fig. 5) showed that an accurate estimation of target image position is fundamental to obtain zero steady state error regulation. The actual algorithm is measuring the center of motion instead of the target position x_i and the position estimation is made only when there is an induced velocity in image ($\dot{x} \neq 0$). Moreover the velocity estimation algorithm is only able to measure velocities up to 2 pixels/frame. Whenever $\dot{x}$ goes beyond this limit the flow is underestimated and the global tracking performance is seriously affected. System transient response is higly dependent on the velocity estimation performance.

The present section discusses improvements in the visual feedback loop to overcome the detected problems. Several methods to measure position and velocity in image have been proposed [28, 29, 30, 31]. Many of these techniques

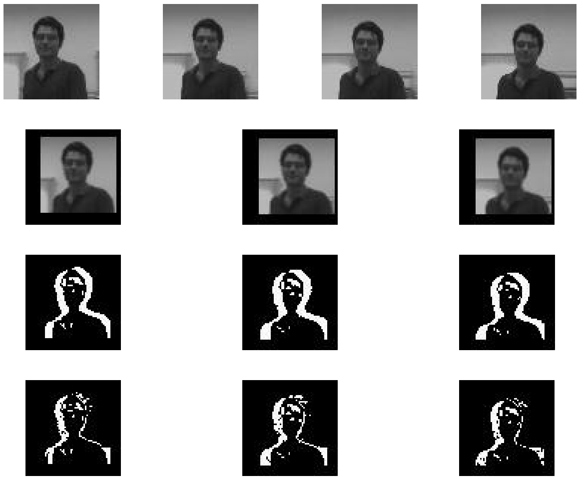

Figure 13: Image sequence acquired during monocular tracking. First row: Original images. Second row: Images after egomotion compensation. Third row: Set of image points with non zero velocity $\dot{x}$. Fourth row: Points used for target position estimation

are accurate and robust but also computationally expensives. The chosen algorithms must present a good balance between accurancy and computational efficiency [32].

5.1 Position Estimation

Consider the image sequence acquired during the monocular tracking of a person Fig. 13(FR). A static backgound (a door and some shelves) is visible behind the person. Velocity $\dot{x}_{ego}$ is computed for each pair of consecutive frames from the motor encoder readings assuming camera pure rotation. The first image of the pair is warped to compensate for the egomotion component (Fig 13(SR)). The MDOF cameras have a narrow field of view (about 15°) thus the egomotion vector is nearly equal in all image points. The obtained warped image is subtracted to the second frame of the pair to segment the points where independent motion occur. The difference frames in the third row of Fig 13 contain the points where $\dot{x} \neq 0$. The points corresponding to the static background have been removed.

Velocity estimation is performed using differential flow. The gradient of the brightness function $\nabla \mathbf{I} = (I_x, I_y, I_z)^t$ is computed in the image points where independent motion occurs. The original position estimation algorithm computes target position as the average location of this set of points. The measured value x gives the center of motion which is a very rough approximation of x_i specially if target induced velocity $\dot{x}$ is high. Restrict the set to the image points of the most recently acquired frame that have nonzero spatial derivatives I_x and I_y (Fig 13(FR)). The average location gives a much better approximation of actual x_i as observed in Fig. 14. Moreover the proposed modification do not introduce aditional computational effort.

Figure 14: Improvement of position estimation performance. Position estimation x using the original algorithm [x] and the new method [o]

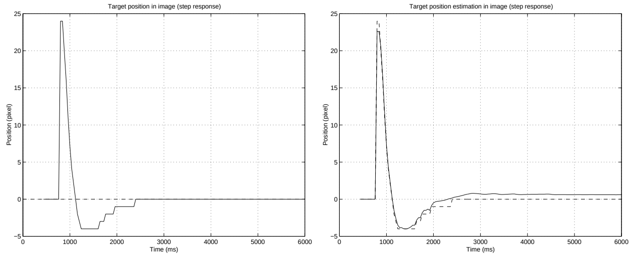

Figure 15: Improvement on position estimation. System response to a step perturbation of amplitude $\Delta x_i = 24$ pixels (the angular amplitude is $\Delta\theta = 6^\circ$). Left: Regulation in image (x_i [–]). Right: Image position estimation (x_i [- -]; x [–])

$$x(k) = x(k-1) + \dot{x}_{ego}(k) \tag{18}$$

The monocular tracking characterization verified that target position is only estimated when target induced velocity $\dot{x}$ is not null. If no target motion is detected after egomotion compensation then the target did not move. The new position estimate should be equal to the previous estimate compensated for the induced displacement due to camera motion (equation 18). Compare the step perturbation response of Fig. 15 with the one of Fig. 9. The modifications in the position estimation algorithm improved the global regulation performance leading to a zero steady error step response.

5.2 Velocity Estimation

To estimate the image velocity induced by target motion, the brightness function gradient is calculated in all pixels where independent motion occur. Only the points that appear in the difference images are taken into account for the flow estimation ((Fig. 13(TR)). Given the flow constraint, and assuming that all points belong to the target and move with the same velocity, the velocity vector $\dot{x} = (u, v)^t$ is estimated using a least squares method [33].

$$I_x.u + I_y.v + I_t = 0 \qquad (19)$$

The flow constraint of equation 19 is valid for a continuous brightness function (under the assumption of brightness constancy). However the actual function $I(x, y, t)$ is discrete in time and space. Aliasing problems in partial derivatives computation can compromise a correct velocity estimation. When the target image moves very slowly high spatial resolution is needed in order to correctly compute the derivatives I_x and I_y and to estimate the velocity. On the other hand, if the target image moves fast, there are high frequencies in time and I_t must be computed with shorter sampling periods. However the sampling frequency is limited to $25Hz$. One solution to estimate high target velocities is to decrease the spatial resolution. The drawback of this approach is that high spatial frequencies are lost, and small target movements will no longer be detected.

The original velocity estimation algorithm uses a 2×2 mask to compute brightness partial derivatives in 64×64 images [34]. The upper limit of the measurable velocity range depends on the mask dimension. Thus the system is unable to estimate velocities beyond 2 pixels/frame. Consider a lower resolution 32×32 image, obtained by subsampling the original 64×64 image. Using the same 2×2 mask we are able to measure velocities up to 2 pixels/frame at 32×32 resolution, which corresponds to 4 pixels/frame at 64×64 resolution. The problem of decreasing the image spacial resolution is that the lower bound of the measurable velocity range is also increased. To perform accurate measurement of a wide range of image velocities the brightness gradient must be computed using a multi scalar approach. Different possible solutions have been tested [16]. The method presenting best results is introduced in the next paragraph.

A pyramid with three levels of image resolution is built. The lower, middle and higher levels have a resolution of 64×64, 32×32 and 16×16 respectively. Target image velocity is estimated at the higher level. A 2×2 mask is used to determine the partial derivatives of the brightness function. The estimated velocity controls the mask dimension at the next resolution level. The brightness gradient in the 32×32 image can be computed using a mask with a size of 2,3 or 4 pixels. The process is repeated for the middle and lower level of the pyramid. Target velocity measured in the 32×32 image controls the lower level mask size. For the 64×64 image the mask dimension can be between 2 and 8 pixels. The optical flow estimated at the lower pyramid level is the final velocity measurement used in system control. The decision about mask dimension for gradient computation is taken based on intervals between predefined threshold values. These threshold values were empirically determined from the experiments results. The three level pyramid allows the accurate measurement of target velocities between 0 and 8 pixels/frame. The range of velocity estimation can be increased by using more levels in the pyramid.

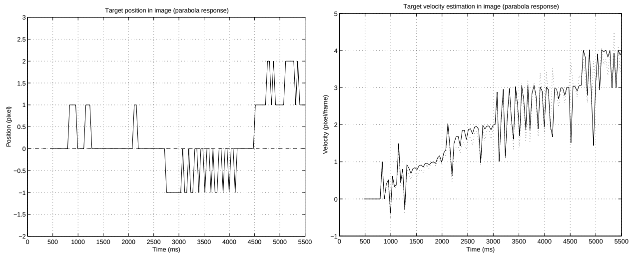

Figure 16: Improvement on velocity estimation. System response to a parabola perturbation of amplitude $\gamma_i = 0.032$ pixels/frame^2 (the angular acceleration is $\dot{\theta} = 5°/\text{s}^2$) Left: Regulation in image (x_i [–]). Right: Estimation of image velocity induced by target motion ($\dot{x}_i - \dot{x}_{ego}$ [...]; $\dot{x}$ [–])

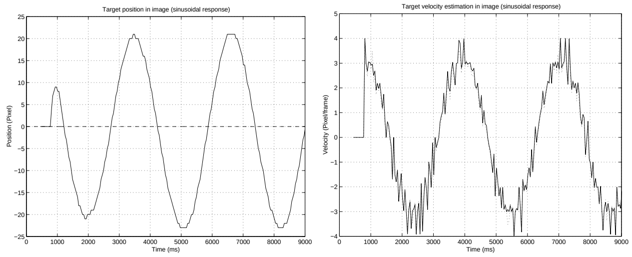

Figure 17: Improvement on velocity estimation. System response to a sinusoid perturbation of amplitude $A_i = 40$ pixels and period $T = \pi$s (the angular amplitude is $A = 10°$). Left: Regulation in image (x_i [–]). Right: Estimation of image velocity induced by target motion ($\dot{x}_i - \dot{x}_{ego}$ [...]; $\dot{x}$ [–]).

Fig. 16 shows the system response to the parabola perturbation using the multi scalar velocity algorithm. The image velocity $\dot{x}$ is correctly estimated and the global regulation performance is much better than the one exhibited in Fig. 11. Fig. 17 shows the response for the sinusoid perturbation. Comparing with the results of Fig. 12 the velocity estimation is perfect but the regulation performance does not significantly improve. As refered in the previous section, the Kalman filter introduces a smoothing effect that compensates for image position and velocity mismeasurements (Fig. 12(BR)). However the filter inertia also limits the impact of the improved velocity estimation in the global system performance. The latency is more noticeable for a sinusoid perturbation, with constant changes of acceleration, than for a ramp or parabola input. The improved visual processing must be complemented by a changes in the control design to achieve better global performances. Next section will discuss Kalman filter tunning and design.

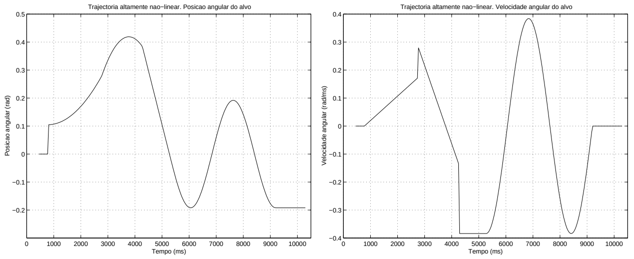

Figure 18: Highly non linear perturbation test signal. Left: Target position θ. Right: Target velocity $\dot{\theta}$

5.3 Kalman Filter for Monocular Tracking

$$\boldsymbol{u}(n) = \begin{bmatrix} 1 & T_i & \frac{T_i^2}{2} \\ 0 & 1 & T_i \\ 0 & 0 & 1 \end{bmatrix} \boldsymbol{u}(n-1) - \begin{bmatrix} \Delta\alpha_p \\ 0 \\ 0 \end{bmatrix} \qquad (20)$$

$$\boldsymbol{Q} = \begin{bmatrix} \sigma_p^2 & 0 & 0 \\ 0 & \sigma_v^2 & 0 \\ 0 & 0 & \sigma_a^2 \end{bmatrix} \qquad (21)$$

The monocular tracking system uses Kalman filtering [42] to estimate target motion in the scene from the visual measurements. Assume that target angular position , velocity and acceleration are respectively θ, $\dot{\theta}$ and $\ddot{\theta}$. Considering that a new frame is grabbed each 40ms, it is reasonable to assume that target angular acceleration is constant between two frames. Thus the filter uses a constant angular acceleration model to describe object motion. Equation 20 is the Kalman state transition equation. The state vector is $\mathbf{u} = (\delta_p, \dot{\theta}, \ddot{\theta})^t$ with $\delta_p = \theta - \alpha_p$ the tracking angular position error and $\dot{\theta}$ and $\ddot{\theta}$ target angular velocity and acceleration. T_i is the frame acquisition period (usually 40ms) and $\Delta\alpha_p$ is the change on pan angular position between two successive iterations $\Delta\alpha_p = \alpha_p(n) - \alpha_p(n-1)$. $\mathbf{Q}$ is the covariance matrix of the error vector of the model $\mathbf{q}$. We will assume that $\mathbf{Q}$ is a diagonal matrix (equation 21) which means that δ_p, $\dot{\theta}$ and $\ddot{\theta}$ are considered statistically independent.

$$\boldsymbol{v}(n) = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \end{bmatrix} \boldsymbol{u}(n) \qquad (22)$$

$$\boldsymbol{R} = \begin{bmatrix} \sigma_{pi}^2 & 0 \\ 0 & \sigma_{vi}^2 \end{bmatrix} \qquad (23)$$

Consider vector $\mathbf{v} = (x, \dot{x})^t$ with x and $\dot{x}$ target position and velocity measured in the image plane. From the first order Taylor expansion of equations 5 and 6 it comes that $x \approx f\delta_p$ and $\dot{x} \approx f\dot{\theta}$. Equation 22 is the Kalman filter measurement equation. $\mathbf{r}$ is the error vector of the measurement equation and $\mathbf{R}$ the corresponding covariance matrix (equation 23). Target position and velocity measurements in the image are assumed statistically decoupled.

	$\mathbf{T_1}$	$\mathbf{T_2}$	$\mathbf{T_3}$	
$\sigma_\mathbf{p}$	0.5	1	1	$(°)$
$\sigma_\mathbf{v}$	0.5	1.5	2	$(°/ms)$
$\sigma_\mathbf{a}$	1.5	2	0	$(°/ms^2)$
$\sigma_\mathbf{pi}$	2	1.5	1.5	(pix)
$\sigma_\mathbf{vi}$	2	1.5	1.5	$(pix/frame)$

Figure 19: Different tunnings of the Kalman filter

Consider the initial state vector $\mathbf{u}(0) = (0, 0, 0)^t$ (target stopped and aligned with the camera optical axis). The tunning of the Kalman filter consists in an adequate choice of the covariance matrices $\mathbf{Q}$ and $\mathbf{R}$.

The choice of σ_{pi} and σ_{vi} depend on the confidance on the visual measurements. If the position and velocity measurements are accurate then the corresponding standard deviations must be low, if not then σ_{pi} and σ_{vi} must be high. In a similar way σ_p, σ_v and σ_a reflect the confidance in the model of motion to describe tracking behavior. If the system is used in tracking a target without sudden acceleration changes then the model fits well the motion and low standard deviations must be chosen. But if the object motion is far from constant acceleration then the trust on the model must be decreased by increasing the standard deviations.

Tab. 19 summarizes three different tunnings to be compared. $\mathbf{T_1}$ is the original tunning that showed good performances in the experiments of section 4. In Fig. 12 the Kalman filter compensates the errors in the velocity estimation allowing a smooth tracking behavior. However this same Kalman filter also avoids that improvements in the visual processing have a positive impact in the system global regulation performance (Fig. 17). Tunning $\mathbf{T_2}$ increases the confidance in the measurements and decreases the confidance in the model. The idea is to reduce the inertia introduced by imposing a pre-defined model of motion. The resultant smoothing effect is useful to avoid unstable tracking behaviors when the visual measurements are unaccurate. The drawback is that it also decreases system responsiveness. In tunning $\mathbf{T_3}$ the standard deviation σ_a of the angular acceleration is set to zero. By doing this the Kalman filter assumes a constant velocity model of motion ($\ddot{\theta}$ is zero all the time).

Fig. 20 compares the regulation performance of the monocular tracking system for the different filter tunnings of Tab. 19. Consider the system response to a parabola perturbation. At the initial instants $\mathbf{T_2}$ and $\mathbf{T_3}$ present better results than $\mathbf{T_1}$. However this situation suffers an abrupt inversion when the target induces a velocity $\dot{x}$ that is beyond the range of measurable velocities. The velocity measurement becomes unaccurate and the more conservative tunning presents the best performance. Fig. 20(M) shows the regulation performance for the sinusoid perturbation. Sys-

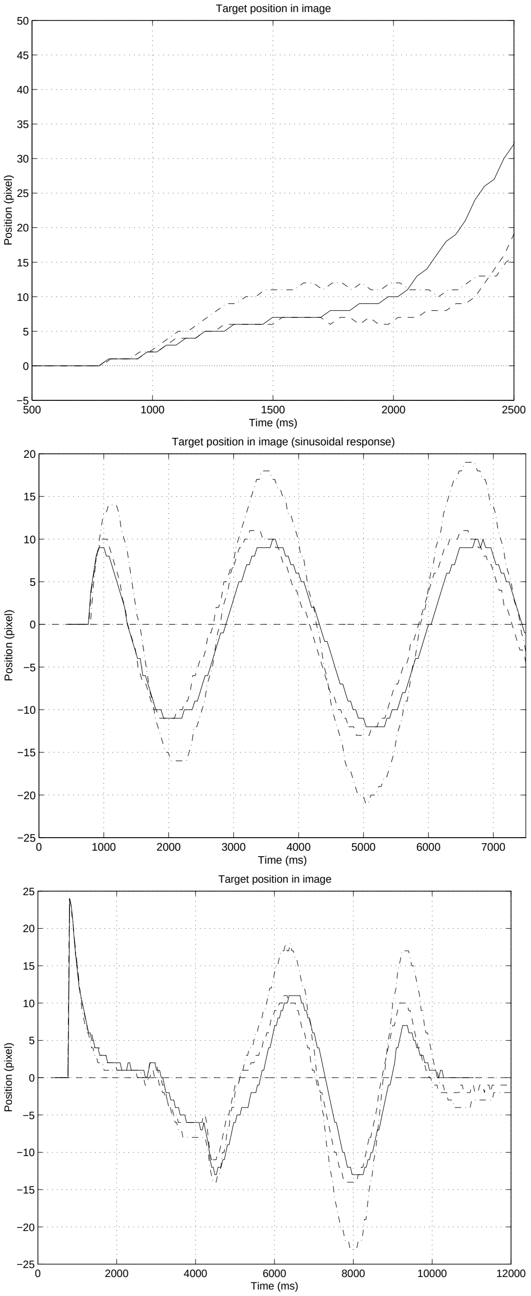

Figure 20: Comparing the different Kalman filter tunnings. Regulation in image (target position x_i. Up: System response to parabola perturbation (T_1[-.-];T_2[- -]; T_3[–]). Midlle: System response to sinusoid perturbation (T_1[-.-];T_2[- -]; T_3[–]). Down: System response to non-linear perturbation of Fig. 18 (T_1[-.-];T_2[--]; T_3[–]).

tem performance with tunnings $\mathbf{T_2}$ and $\mathbf{T_3}$ is clearly better than with $\mathbf{T_1}$. Now the improvements in the position and velocity estimation have a positive impact in the global system performance. The results for the non-linear input confirm the the increased responsiveness of the system. Notice that the performance with $\mathbf{T_2}$ and $\mathbf{T_3}$ is quite similar. However $\mathbf{T_3}$ is slightly better when the target suffers abrupt acceleration changes. If nothing is said we will assume tunning $\mathbf{T_3}$

5.4 Kalman Filter for Vergence Control

A scheme of the vergence control is presented in Fig. 4. The target moves along the cyclopean axis and the goal is to make camera angular position β to converge for the vergence angle ψ.

$$\left\{ \begin{array}{l} \rho(n) = \rho(n-1) + \dot{\rho}(n-1)T_i \\ \dot{\rho}(n) = \dot{\rho}(n-1) \end{array} \right. \tag{24}$$

$$\boldsymbol{Q_\rho} = \left[\begin{array}{cc} \sigma_p^2 & 0 \\ 0 & \sigma_v^2 \end{array} \right] \tag{25}$$

The Kalman filter for the vergence control assumes that the target has a constant velocity motion along the cyclopean axis. The model of motion is given by equation 24 where ρ is the target depth in cyclopean coordinates, $\dot{\rho}$ the corresponding velocity and T_i the time between two frames. Q_ρ is the covariance matrix of the error vector of the model. Target position and velocity are considered statistically decoupled.

$$\left\{ \begin{array}{l} \psi(n) = \arctan(\frac{B}{\rho(n)}) \approx \frac{B}{\rho(n)} \\ \dot{\psi}(n) \approx -\frac{B}{\rho^2(n)}\dot{\rho}(n) \end{array} \right. \tag{26}$$

Equation 26 establishes the relation between target depth ρ and the vergence angle ψ ($2B$ corresponds to the baseline lengths). If $\beta = \psi$ the vergence point coincides with the target and the object is projected in the center of both retinas. In the model of equation 24 target motion is not parameterized in the same space as the controlled variables. The relations presented in equation 26 are used to rewrite the model of motion in term of the vergence angular position and velocity.

$$\boldsymbol{u}(n) = \left[\begin{array}{c} \frac{(\delta_v(n-1)+\beta(n-1))^2}{\delta_v(n-1)+\beta(n-1)-\dot{\psi}(n-1)T_i} - \beta(n) \\ \frac{(\delta_v(n-1)+\beta(n-1))^2}{(\delta(n-1)+\beta(n-1)-\dot{\psi}(n-1)T_i)^2}\dot{\psi}(n-1) \end{array} \right] \tag{27}$$

$$\boldsymbol{Q}(\mathrm{n}) = \frac{B^2}{m_p^4(n)} \left[\begin{array}{cc} \sigma_p^2 & -\frac{2m_v(n)}{m_p(n)}\sigma_p^2 \\ -\frac{2m_v(n)}{m_p(n)}\sigma_p^2 & \frac{4m_v^2(n)}{m_p^2(n)}\sigma_p^2 + \sigma_v^2 \end{array} \right] \tag{28}$$

$$\begin{array}{rcl} m_p(n) & = & \frac{B(\delta_v(n-1)+\beta(n-1)-\dot{\psi}(n-1)T_i)}{(\delta_v(n-1)+\beta(n-1))^2} \\ m_v(n) & = & -\frac{B\dot{\psi}(n-1)}{(\delta_v(n-1)+\beta(n-1))^2} \end{array} \tag{29}$$

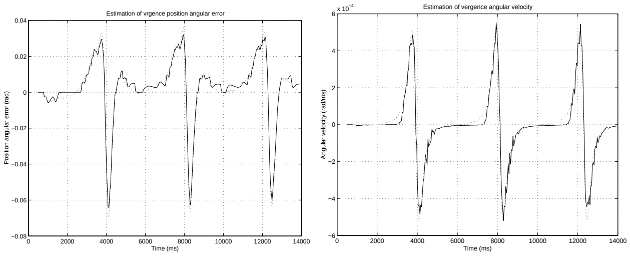

Figure 21: Kalman filter estimation performance for a sinusoidal perturbation ($A = 5$(m),$T = 1.3\pi$(s)). Left: Estimation of the vergence position angular error δ_v (correct δ_v [...]; estimated δ_v [–]). Right: Estimation of the vergence angular velocity $\dot{\psi}$ (correct $\dot{\psi}$ [...]; estimated $\dot{\psi}$ [–])

The resultant state transition equation is showed in 27. The state vector is $\mathbf{u} = (\delta_v, \dot{\psi})^t$ with $\delta_v = \psi - \beta$ the vergence angular position error and $\dot{\psi}$ the vergence angular velocity. The dynamic equation is non linear and an Extended Kalman Filter (EKF) must be used [43]. The EKF is a non optimal estimation algorithm that uses the Jacobian matrix of the non-linear function to compute the covariance information. The model covariance matrix $\mathbf{Q}$ is derived from $\mathbf{Q}_\rho$ using the function of equation 26 to propagate the covariances [40].

$$\boldsymbol{v}(n) = \left[\begin{array}{cc} 2f & 0 \\ 0 & 2f \end{array} \right] \boldsymbol{u}(n) \tag{30}$$

$$\boldsymbol{R} = \left[\begin{array}{cc} \sigma_{pi}^2 & 0 \\ 0 & \sigma_{vi}^2 \end{array} \right] \tag{31}$$

The kalman filter measurement equation is given in 30 and the corresponding covariance matrix $\mathbf{R}$ in 31. The measurement vector is $\mathbf{v} = (\Delta x, \Delta \dot{x})^t$ with Δx the target position disparity and $\Delta \dot{x}$ the target velocity disparity. The linear relation between $\mathbf{v}$ and the state vector $\mathbf{u}$ is derived from the first order Taylor expansion of equations 8 and 10. The Kalman filter is tunned by the procedings used for the monocular tracking situation.

$$\boldsymbol{K} = \left[\begin{array}{cc} k_{11} & k_{12} \\ 0 & k_{22} \end{array} \right] \tag{32}$$

Fig. 21 shows the state estimation for a sinusoidal variation of target depth. While in the monocular tracking system the Kalman gain matrix converges for constant values, for the vergence control $\mathbf{K}$ changes during operation. Its dynamic behavior can be observed in Fig. 22. Notice that when target depth diminishes the gains tend to increase. If the target is close to the system any small depth variation induces a significant disparity in image. However if the target is far away the same motion can become visually imperceptible. The Kalman filter estimation copes with this situation by increasing the confidance on the visual measurements when the object is near and transfering the trust to the motion model when the target is far away.

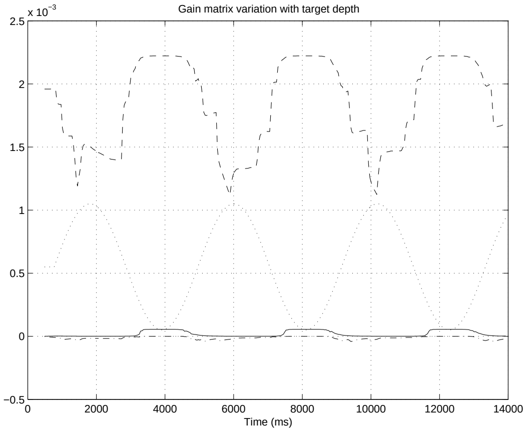

Figure 22: Dynamic behavior of the Kalman gain matrix $\mathbf{K}$. Response for a sinusoidal perturbation. (virtual target depth ρ [...]; k_{11} [- -]; k_{12} [-.-]; k_{22} [–]).

Fig. 23 shows the vergence regulation performance for the different perturbation signals derived in section 3.3

6 System Architecture and Gaze Control

The present section discusses the architecture and control of active tracking systems. We will focus on the monocular tracking however most of the concepts and ideas can be extended to vergence control and general visual servoing applications. 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 24.

The visual feedback loop is discussed in previous sections. The image processing must be fast, accurate and robust to achieve high performance behaviors. Kalman filtering is 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 sampling rate of the visual loop is equal to the frame rate (typically 25Hz).

The low-level servo controller commands the active plataform actuators. The choice of these actuators and the corresponding control strategie are discussed in [35]. 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

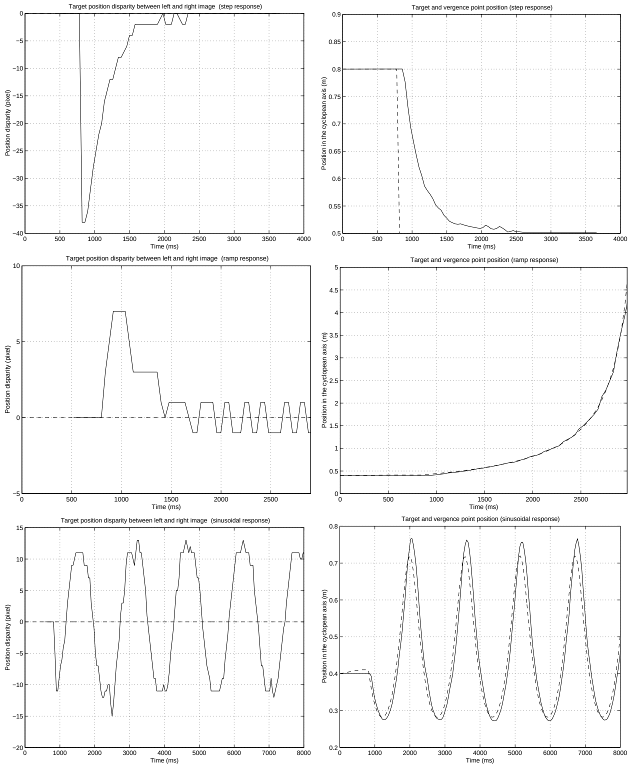

Figure 23: Performance evaluation of the vergence control system. First row: Step perturbation. Second row: Ramp perturbation. Third row: Sinusoid perturbation. Left Column: Regulation in image (target position disparity Δx_i [–]). Right Column: Response in position along the cyclopean axis (vergence point depth [–], target depth ρ [- -])

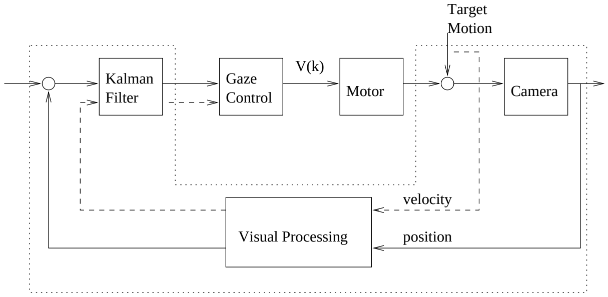

Figure 24: Visual tracking system architecture. Three concurrent processes can be identified: image processing, low-level servo-control and gaze control

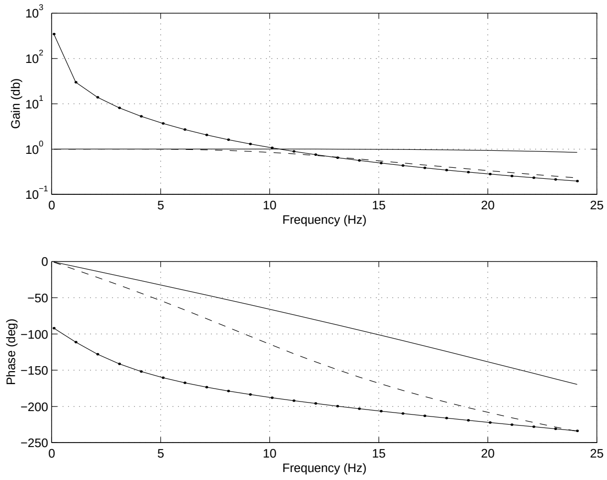

Figure 25: Actuator frequency response for different control strategies (open loop [-.-]; closed loop - position mode [- -]; closed loop - velocity mode/velocity output [–])

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 [18, 32, 2]. 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.

6.1 Low-Level Servo Control Loop

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. The DC drives of the MDOF head are equipped with 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 directly controlled using visual feedback. In this case the sampling rate is equal to the frame rate (25Hz). Fig. 25 shows the motor open loop frequency response. The extra phase-lag introduced by the visual processing will tend to worsen the stability of the system. System unstability can be avoided by decreasing the

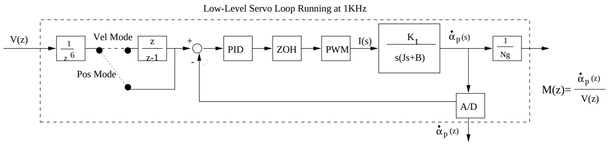

Figure 26: Low-level servo control loop. The loop runs at 1KHz and can operate both on position and velocity mode. The communication is syncronous at a rate of 166Hz

loop gain. However the low sample rate associated to the small gains will lead to sluggish system responses.

Encoder information can be used to implement a local servo loop running at a high sample rate. This servo loop will improve the frequency response of the mechanical actuator. The MDOF system comprises a commercial multi-axis PC controller card with a dedicated servo control module for each degree of freedom of the platform [20]. Each axis is controlled in position by a local closed loop with a digital PID filter running at 1KHz (Fig. 26). 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 6 ms interval. The communication delay introduces undesired phase-lag in the loop. Despite of that Fig. 25 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 (Fig. 26). 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. 25 displays the frequency response when the motor is controlled in velocity mode. The improvements are due to the fact that the profile generator is running at 1KHz, updating the reference to the position loop every 1ms.

Motor control will be performed using the velocity mode. The servo system input is the velocity command $V(k)$ and the output the motor velocity $\dot{\alpha}_p(k)$ measured using the optical encoder. The loop runs at 1KHz but the maximum communication rate is 166Hz. $M(z)$ is the servo-loop transfer function considered from the point of view of the host PC (see Fig 26). The PID filter was tuned such that $M(z)$ presents a critically damped response without overshoot. The final closed loop transfer function was determined using standard linear system identification techniques.

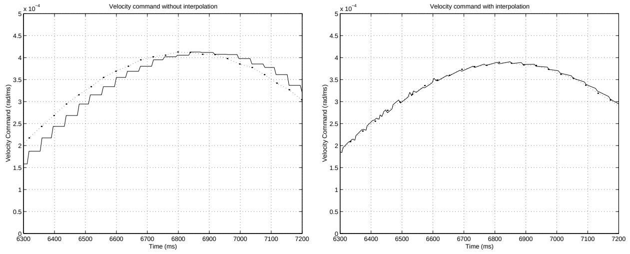

Figure 27: Using interpolation to cope with visual latency. Left: Motor command $V(k)$ without interpolation ($V(k)$[–]; ideal $V(k)$[...]; frame acquisition time instant [.]). Right: Motor command $V(k)$ with interpolation ($V(k)$[–]; ideal $V(k)$[...]; frame acquisition time instant [.])

6.2 Interpolation to Cope with Visual Latency

When a frame is acquired it is not immediately available to be processed. The image acquisition process presents a certain inertia. The time interval going between the acquisiton and the availability for processing is called the image acquisition delay. The image acquisition delay in the MDOF system is about 30ms. The visual processing delay is the time the system needs to process the images. For the monocular tracking the delay is 6ms and for the vergence control, where two frames must be processed, is 11ms. The sum of these two delays is called the system visual latency. This section discusses the visual latency for the monocular tracking system. The presented concepts can be generalized to the vergence control.

In the monocular tracking system target motion parameters are available at Kalman filter output 36ms after image acquisition (30ms of acquisition delay and 6ms of processing delay). The results presented in Fig. 27 were obtained for the non-linear system perturbation of Fig. 18. Assume no errors on visual measurements and perfect estimation of target angular motion parameters. The target motion information is sent to the gaze controller every 40ms, with a delay of 36ms. Fig. 27(L) compares the ideal velocity command with the actual motion command $V(k)$. The visual latency introduces a phase-lag that is going to deteriorate system global performance. Interpolation can be used to cope with the visual latency.

Fig. 28 shows the block diagram of the monocular tracking system. The visual loop sends the target motion estimation $\mathbf{u} = (\delta_p, \dot{\theta}, \ddot{\theta})^t$ to the gaze controller. The communication rate with the low level servo loop is 166Hz. This means that the gaze controller can send commands and read motor encoders every 6ms. However the visual control loop runs at 25Hz and state vector $\mathbf{u}$ is updated with a 40ms interval. Without interpolation command $V(k)$ sent to the

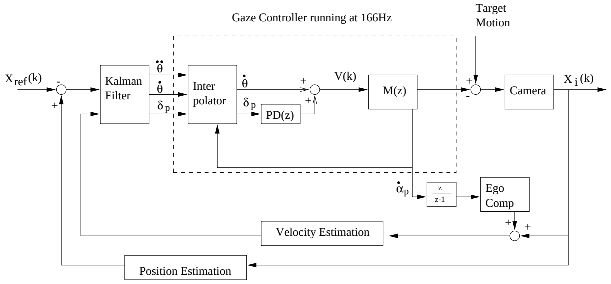

Figure 28: Block diagram fo the monocular tracking system using interpolation and a proportional-derivative gaze controller.

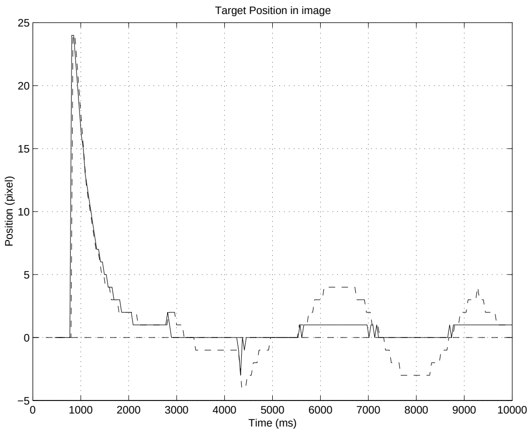

Figure 29: Using interpolation to cope with visual latency. System response to the non-linear perturbation of Fig. 18. Regulation in image (x_i using interpolation[–], x_i without using interpolation [- -]).

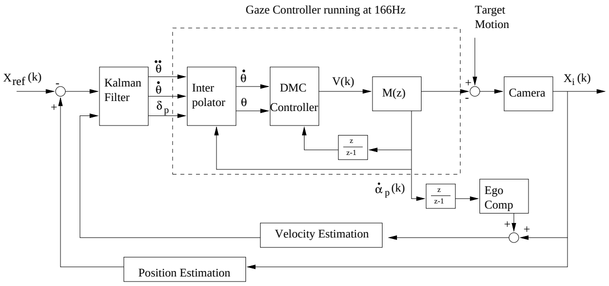

Figure 30: Block diagram fo the monocular tracking system using interpolation and a model predictive gaze controller.

servo control is constant during visual update interval (see Fig. 27(L)). Interpolation, assuming a constant acceleration model for target motion, is used to cope both with visual latency and the low sampling rate of the visual loop. The interpolator updates the velocity command $V(k)$ every 6ms taking the maximum advantage of the communication rate with the servo loop.

Fig. 27(R) shows that by using interpolation the velocity command $V(k)$ converges to the ideal command. The interpolation performance highly depends on the assumed model of motion and on the accuracy of the visual estimations. Considering the rate of the visual loop, the constant acceleration motion between frames is a reasonable assumption. In Fig. 27(R) the ripple at the top of the sinusoid is due to the highly non-linear variation of the target velocity that is not described by the constant acceleration model. The observed ripple has almost no effect on system performance. However it is important to remark that the results of Fig. 27 assume perfect estimation of target motion. If the results of visual estimation are very noisy or unaccurate then using intepolation can have a negative impact on the global system performance. Basically interpolation amplifies the performance of the visual loop, both in positive and negative direction.

Fig. 29 shows the improvements introduced by interpolation in the system regulation global performance.

6.3 Gaze Control

Fig. 28 shows the monocular tracking control loop. The gaze controller establishes the interface between the visual control loop, running at 25Hz, and the low level servo loop, running at 1KHz. The sampling rate of the gaze controller must be equal to the communication rate with the servo loop in order to maximize system performance [36].

$$V(z) = \dot{\theta}(z) + PD(z).\delta_p(z) \tag{33}$$

The gaze controller used in the previous experiments is given by equation 33. PD(z) is a proportional derivative controller that filters the angular position error δ_p. Notice that to achieve a zero steady state error the transfer function PD(z).M(z) must have one unitary pole (type 1 system). This unitary pole is introduced by the velocity mode profile generator of the servo-loop. The velocity component $\dot{\theta}$ improves global system transient response (velocity feedforward).

6.3.1 Model Predictive Gaze Controller

Fig. 25 shows the frequency response of the low level servo control loop (velocity mode). The observable phase-lag is introduced the communication delays and motor mechanical inertia. The transfer function $M(z)$, determined using standard system identification techniques, 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 [17, 19].

$$J = \sum_{i=N_1}^{N_2} (y(n+i|n) - w(n+i))^2 + \sum_{j=N_1}^{N_u} \lambda \Delta u(n+j-1)^2 \tag{34}$$

There is 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 $y(n+k|n)$ at future time instants knowing previous inputs and outputs. The general aim is to make future system outputs to converge for a desired reference $w(n)$. For that an objective function J is established. The general expression for such a function is given by equation 34. 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 relative weight used to achieve a more or less smooth 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. Our process has a dead time of 2, thus we are going to consider $N_1 = 2$ (the output can not be forced before that). Assuming a frame rate of 25Hz, the middle level controller sends at most 7 velocity commands to the low-level loop without new visual information. Thus we are going to consider $N_2 = 8$.

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 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.

$$\boldsymbol{\Delta u} = (\boldsymbol{G}\boldsymbol{G}^{\boldsymbol{t}} + \lambda \boldsymbol{I})^{-1} \boldsymbol{G}^{\boldsymbol{t}} (\boldsymbol{w} - \boldsymbol{f}) \tag{35}$$

DMC uses the N first instants from the step response to predict the 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 ($d(n) = y(n) - y(n|n)$). The goal of our 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 35. $\mathbf{G}$ is the dynamic matrix of the system, $\mathbf{\Delta u}$ is the control vector and $\mathbf{w}$ is the reference vector. $\mathbf{f}$ is called the free response vector because it does not depend on the future control actions. λ is the penalty for the control effort, by increasing this value the system becomes less responsive and smoother. Notice that only the first element of $\mathbf{\Delta u}$ 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 [37].

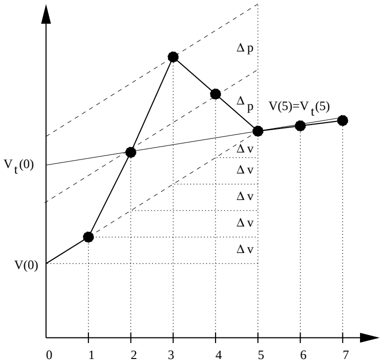

Figure 31: Reference w

Interpolation can be used, not only to compensate for the visual processing delay, but also to estimate target parameters of motion for future time instants. Visual information at the Kalman filter output is used to compute current and future target angular position and velocity assuming a constant acceleration model of motion. The goal of the DMC controller is to force the motor to have the same motion as the target in a near future.

$$\Delta_p = \frac{1}{6}(\frac{E_p}{T} - MV(0)) - \frac{\Delta_v}{3}(\frac{M^2}{4} + 2) \tag{36}$$

Whenever a new image is grabbed, visual processing is used to compute target velocity and 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 **w** whose profile is depicted in Fig. 31. Consider $P(0)$ and $V(0)$ are the current motor position and velocity and $P_t(i)$ and $V_t(i)$ are the target position and velocity at instant i. Then $\Delta_v = (V_t(M) - V(0))/M$ and Δ_p is computed by equation 36 where $E_p = P_t(M) - P(0)$. M is the instant of convergence, making $M = 5$ motor velocity converge to target velocity in 5 samples (30ms). In this time interval the motor accelerates and then slightly decelerates to compensate for the position error (see Fig.31).

The increase in performance introduced by the DMC controller can be observed in Fig32. The error in position is immediately compensated and the target is kept in the center of the image during its non linear motion. However the performance of the DMC controller is highly dependent on the ability to predict the target motion. If the visual measurements are too noisy or the interpolation model is not accurate the use of model ptrdictive control techiques can deteriorate the tracking system global performance [38].

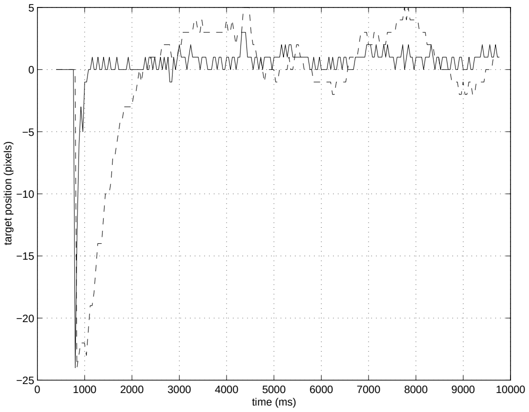

Figure 32: Using DMC controller to cope with motor mechanical inertia. System response to the non-linear perturbation of Fig. 18. Regulation in image (x_i for DMC gaze control [–], x_i for the PD gaze control [- -]).

7 Conclusions

The starting point this work is the ISR-MDOF binocular active tracking system [12, 11, 13]. Visual control of motion depends both in vision and controls. An approach to characterize these both aspects in a common framework is introduced. System performance is evaluated. The high coupling between visual and control is illustrated by extensive experimental results. Performance evaluation is used as guideline to develop a high performance active vision system. Real time algorithms for position and velocity image measurement are proposed. Specific Kalman filters are designed for both monocular tracking and vergence control and the corresponding tunning is exhaustively discussed. An architecture, based on three concurrent processes, is suggested. Interpolation to cope with visual latency is proposed. A gaze controller based on a model predivtive approach is designed. A demo of the high performance visual tracking is presented.

References

[1] J. Hill and W. T. Park, "Real time control of a robot with a mobile camera," in *in Proc. of 9th ISIR*, Washington DC, March 1979, pp. 233–246.

[2] P. M. Sharkey and D. W. Murray, "Delay versus performance of visually guided systems," *IEE Proc.-Control Theory Appl.*, vol. 143, no. 5, pp. 436–447, September 1996.

[3] P. I. Corke and M. C. Good, "Dynamic effects in visual closed-loop systems," *IEEE Trans. on Robotics and Automation*, vol. 12, no. 5, pp. 671–683, October 1996.

[4] P. I. Corke, *Visual Control of Robots: High-Peformance Visual Servoing*, Mechatronics. John Wiley, 1996.

[5] H. I. Christensen, J. Horstmann, and T. Rasmussen, "A control theoretic approach to active vision," *Asian Conference on Computer Vision*, pp. 201–210, December 1995.

[6] E. Dickmanns, "An approach to robust dynamic vision," in *in Proc. of the 15th Int. Conf. on Artificial Intelligence*, Nagoya, Japan, May 1997.

[7] N. Papanikolopoulos, P. K. Khosla, and T. Kanade, "Visual tracking of moving targets by a camera mounted on a robot. a combination of vision and control," *IEEE Trans. on Robotics and Automation*, vol. 9, no. 1, pp. 429–445, 1993.

[8] J. Batista, J. Dias, H. Araújo, and A. Almeida, "The isr multi-degree-of-freedom active vision robot head: Design and calibration," in *in Proc. of M2VIP'95 - Second International Conference on Mechatronics and Machine Vision in Practice*, Hong-Kong, September 1995.

[9] J. Ekhlund, K. Pahlavan, and T. Uhlin, "The kth head-eye system," in *Vision as a Process*, J. Crowley and H. Christensen, Eds., pp. 237–259. Springer-Verlag, 1995.

[10] D. W. Murray, P. McLauchlan, I. Reid, and P. Sharkey, "Reactions to peripheral image motion using a head/eye platform," in *ICCV93 - IEEE International Conference on Computer Vision*, 1993, pp. 403–411.

[11] J. Batista, P. Peixoto, and H. Araujo, "Real-time visual behaviors with a binocular vision system," in *ICRA97 - IEEE International Conference on Robotics and Automation*, New Mexico, Setembro 1997.

[12] J. Batista, P. Peixoto, and H. Araujo, "Real-time vergence and binocular gaze control," in *IROS97– IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Grenoble, France, September 1997.

[13] J. Batista, P. Peixoto, and H. Araujo, "Real-time active visual surveillance by integrating peripheral motion detection with foveated tracking," in *Proc. of the IEEE Workshop on Visual Surveillance*, Bombay, India, January 1998, pp. 18–25.

[14] João P. Barreto, P. Peixoto, J. Batista, and H. Araujo, "Evaluation of the robustness of visual behaviors

through performance characterization," in *Robust Vision for Vision-Based Control of Motion*, Markus Vincze and Gregory D. Hager, Eds. IEEE Press, 1999.

[15] João P. Barreto, P. Peixoto, J. Batista, and H. Araújo, "Control performance issues in a binocular active vision system," in *IROS98–IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Victoria, Canada, October 1998, pp. 233–246.

[16] João P. Barreto, P. Peixoto, J. Batista, and H. Araújo, "Improving 3d active visual tracking," in *Computer Vision Systems, Lectures Notes in Computer Science 1542*, Henrik I. Christensen, Ed. 1999, Springer.

[17] João P. Barreto, J. Batista, and H. Araújo, "Solution for visual control of motion: Active tracking applications," in *MED2000 – IEEE Mediterranean Conference on Automatic Control*, Patras, Greece, July 2000, pp. 233–246.

[18] C. Brown, "Gaze controls with interactions and delays," *IEEE Trans. on Systems, Man and Cybern.*, vol. 20, no. 2, pp. 518–527, 1990.

[19] João P. Barreto, J. Batista, and H. Araújo, "Model predictive control to improve visual control of motion," in *ICPR2000–Int. Conf. on Pattern Recognition*, Barcelona, Spain, September 2000, pp. 233–246.

[20] *DCX-AT100 Modular Multi-Axis Motion Controller, Users Manual, Version 1.1a*, Precision MicroControl, 1993.

[21] Richard Hartley, "Self-calibration from multiple views with a rotating camera," in *Proc. of European Conference of Computer Vision*, 1992.

[22] Joao P. Barreto, P. Peixoto, J. Batista, and H. Araujo, "Mathematical analysis for visual tracking assuming perspective projection," in *SIRS2000–8th Int. Symposium on Intelligent Robotic Systems*, Reading, England, July 2000, pp. 233–246.

[23] P. I. Corke, "Visual control of robot manipulators– a review," in *Visual Servoing*, K. Hashimoto, Ed., pp. 1–31. World Scientific, New York, 1993.

[24] W. Wilson, C. Hulls, and G. Belles, "Relative end effector control using cartesian position-based visual servoing," *IEEE Trans. on Robotics and Automation*, vol. 12, no. 5, pp. 651–670, October 1996.

[25] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Trans. on Robot. and Automat.*, vol. 8, no. 3, pp. 313–326, June 1992.

[26] S. Hutchinson, G. Hager, and P. I. Corke, "A tutorial on visual servo control," *IEEE Trans. on Robotics and Automation*, vol. 12, no. 5, pp. 651–670, October 1996.

[27] Francois Chaumette, "Potential problems of stability and convergence in image based and position based visual servoing," in *The Confluence of Vision and Control*. in Lecture Notes in Control and Information Systems, Springer-Verlag, 1998.

[28] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Performance of optical flow techniques," *International Journal of Computer Vision*, vol. 12, no. 1, pp. 43–77, 1994.

[29] David J. Heeger, "Optical flow using spatio temporal filters," *International Journal of Computer Vision*, pp. 279–302, 1988.

[30] M. Okutomi and T. Kanade, "A locally adaptive window for signal matching," *International Jounal of Computer Vision*, , no. 7:2, pp. 143–162, 1992.

[31] Maximo Tistarelli, "Multiple constraints to compute optical flow," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 18, no. 12, pp. 1243–1250, December 1996.

[32] P. Krautgartner and M. Vincze, "Performance evaluation of vision-based control tasks," in *IEEE International Confernce on Robotics and Automation*, Leuven, Belgium, May 1998.

[33] J.R. Bergen, P.J. Burton, R. Hingorani, and S. Peleg, "Computing two motions from three frames," in *Technical Report, David Sarnoff Research Center*, April 1990.

[34] Stan Birchfield, "Derivation of kanade-lucas-tomasi tracking equation," in *Technical Report*, January 1997.

[35] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the American Society of Mechanical Engineers. Journal of Basic Engineering*, pp. 35–45, Mars 1960.

[36] Yaakov Bar-Shalom and Xiao-Rong Li, *Multitarget-Multisensor Tracking: Principles and Techniques*, Yaakov Bar-Shalom, 1995.

[37] Athanasios Papoulis, *Probability, Random Variables and Stochastic Processes*, MacGraw-Hill International Editions, 1991.

[38] P. M.Sharkey and D. W. Murray, "Bandwidth issues for active vision systems," in *in Proc. of M2VIP'95*

- *Second International Conference on Mechatronics and Machine Vision in Practice*, Hong-Kong, September 1995.

[39] S. Chroust, M. Vincze, R. Traxl, and P. Krautgartner, "Evaluation of processing architecture and control law on the performance of vision-based control systems," in *AMC00–IEEE/RSJ Int. Workshop on Advanced Motion Control*, Japan, April 2000, pp. 19–24.

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

[41] S. Chroust, J. P. Barreto, H. Araujo, and M. Vincze, "Comparison of control structures for visual servoing," in *ICAR2001 - 10th International Conference on Advanced Robotics*, Budapest, Hugary, August 2001.

[42] P. M.Sharkey, P. F. MaLauchlan D. W. Murray, and J. P. Brooker, "Hardware development of the yorick series of active vision systems," *Microprocessors and Microsystems*, , no. 21, pp. 363–375, 1998.