SAND REPORT

SAND2004-1868
Unlimited Release
Printed May 2004

Visual Servoing Using Statistical Pressure Snakes

Hanspeter Schaub, ORION International Technologies

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94-AL85000.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.doe.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online ordering: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2004-1868
Unlimited Release
Printed May 2004

Visual Servoing Using Statistical Pressure Snakes

Hanspeter Schaub
ORION International Technologies
Albuquerque, NM 87106
schaub@vt.edu

Abstract

A nonlinear visual servoing steering law is presented which is used to align a camera view with a visual target. A full color version of statistical pressure snakes is used to identify and track the target with a series of video frames. The nonlinear steering law provides camera-frame centric speed commands to a velocity based servo sub-system. To avoid saturating the subsystem, the commanded speeds are smoothly limited to remain within a finite range. Analytical error analysis is also provided illustrating how the two control gains contribute to the stiffness of the control. The algorithm is demonstrated on a pan and tilt camera system. The control law is able to smoothly realign the camera to point at the target.

Contents

Figures

Visual Servoing Using Statistical Pressure Snakes

Introduction

Statistical pressure snakes are a means to segment an image into specific target regions.[1] The target is assumed to be defined through a set of mean HSV color values with associated standard deviations.[2] The snake (closed parametric curve) will settle on the outer edge of the image region defined through these colors. Note that the snake will wrap around targets of arbitrary shapes. Using the HSV color space, the resulting snake curves are robust to large changes in lighting conditions. This feature makes them attractive for visual servoing applications in unstructured outdoor environments. Given a snake curve, it is possible to extract snake features such as the snake center of mass (CM), the snake principal axes of inertia (based on the area moments of the snake), as well as the heading angle of the primary principal axis.[3] Assuming the target is of a known dimension, it is also possible to estimate the target depth (z-axis location) from the snake features.[3] These snake features can then be used to servo a system to desired positions and orientations *relative* to the target.

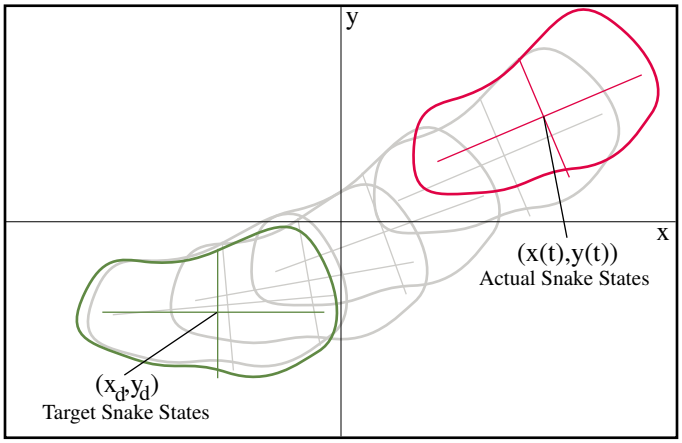

Figure 1. Illustration of the Statistical Pressure Snake Based Servoing System

This idea is illustrated in Figure 1. The current snake is shown in magenta, while the target snake location is shown in green. If a robot arm is to pick up a tool, then the green

target location would be the apriori known location that the robot arm must be at, as seen by the camera, in order to be properly lined up for the next maneuver. In this paper, unless noted otherwise, the target is assumed to be stationary in the inertial world frame. After engaging the snake-based visual servoing routine, we would like the camera to move such that the snake CM and orientation line up with the desired target values. Note that these target values do not have to be zero.

This technical report presents a simple velocity-based servo control law that will align the camera relative to a sensed target. Nonlinear modifications are introduced to avoid saturating the speed commands if large state errors exist. The resulting nonlinear control law is still analytically guaranteed to be asymptotically stabilizing, assuming the velocity-servo subsystem can accurately track the desired speed commands. A practical application of this snake-based visual servoing is discussed.

Camera Equations of Motion

In this control development, the camera is assumed to be attached to a manipulator system with known kinematics. Given a speed command relative to the camera frame (for example, move to the right), a servo-subsystem is present that will achieve the desired motion and solve the inverse kinematics required for the straight line motion relative to the camera frame. The availability of such a servo-subsystem greatly simplifies the following camera motion description, resulting in the complex manipulator kinematics becoming decoupled from the camera motion problem. Assuming the target is stationary (regulator servo problem), the end result is a simple linear camera motion model where the camera motion is directly proportional to the commanded camera speed.

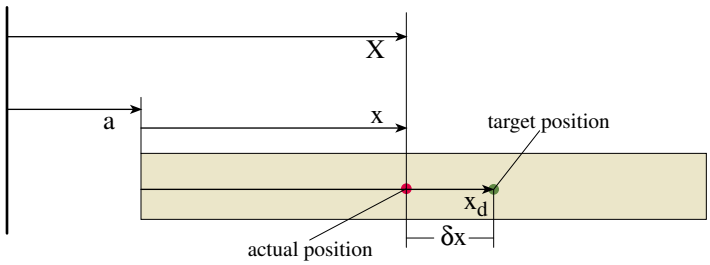

Figure 2. One-Dimensional Illustration of the Camera Motion

Due to the linear nature of the camera motion, the motion of each degree of freedom

is decoupled from the other degrees of freedom. This allows us to focus on one degree of freedom, the x-axis motion for example, without loss of generality. Figure 2 illustrates x-axis motion of the target as seen by the camera. The position of the target (snake center of mass x-axis location in this setting) relative to the inertial world frame is given by X. The perceived location of the target in the camera frame is given by x, with a being the world location of the camera frame origin. These three states are related through

$$X = a + x \tag{1}$$

Taking the first time derivative of this kinematic relationship, we find

$$\dot{X} = \dot{a} + \dot{x} \tag{2}$$

Note that $u = \dot{a}$ is the horizontal camera speed, which for the current application is also our control variable. The perceived target motion $\dot{x}$ in the camera frame is then expressed as

$$\dot{x} = \dot{X} - u \tag{3}$$

Note the mirroring effect. For the perceived target to move to the right, the camera must be commanded to move to the left. Eq. (3) is the camera equation of motion for the x-axis. Equivalent equations can be written for the remaining snake degrees of freedom (y-axis, heading and depth motion).

Let the desired position of the target in the camera frame be given by x_d. We would like to have the the actual snake coordinate x line up with x_d. The tracking error δx is then defined as

$$\delta x = x - x_d \tag{4}$$

Since the desired target position is constant as seen by the camera frame, note that $\dot{x}_d = 0$. Taking the first time derivative of Eq. (4) and making use of Eq. (3), we find

$$\delta \dot{x} = \dot{x} - \dot{x}_d = \dot{x} = \dot{X} - u \tag{5}$$

This equation represents the tracking error dynamics for a given camera speed command u. Further, if the target is assumed to be stationary in the inertial world frame, then $\dot{X} = 0$ and the tracking error dynamics are simplified to

$$\delta \dot{x} = -u \tag{6}$$

Nonlinear Velocity-Based Steering Law

This section presents the nonlinear steering law that is used by the visual servo system to align the snake degrees of freedom with known, stationary camera frame locations. Again, we focus on the camera x-axis motion control. However, the same control can also be applied to the other snake degrees of freedom due to the decoupled nature of the camera motion.

Let the x-axis camera speed command u be defined as

$$u = \underbrace{\frac{\ln 2}{T_{1/2}}}_{\gamma} f(\delta x) \tag{7}$$

where $f(\delta x)$ is an odd function satisfying the property

$$f(\delta x) \cdot \delta x > 0 \tag{8}$$

and $\gamma > 0$ is a control gain. If we choose the simple linear function $f(\delta x) = \delta x$, then the control law in Eq. (7) simplifies to the linear servo steering law

$$u = \gamma \delta x \tag{9}$$

This is a classical linear position error feedback law. Substituting Eq. (9) into the error dynamics in Eq. (6), we find the corresponding closed-loop error dynamics to be

$$\delta \dot{x} + \gamma \delta x = 0 \tag{10}$$

Solving this differential equation, we find that the speed control in Eq. (9) will cause any initial tracking error $\delta x(t_0)$ to exponentially decay to zero with the half-life constant of $T_{1/2}$. If the inertial target location is not constant (i.e. $\dot{X} \neq 0$), the closed-loop dynamics is then given by

$$\delta \dot{x} + \gamma \delta x = \dot{X} \tag{11}$$

If $\dot{X} =$ constant, the the steady-state tracking error will converge to

$$\delta x_{ss} = \frac{\dot{X}}{\gamma} \tag{12}$$

The larger the gain γ is, the smaller the steady-state tracking error will be.

The drawback of the simple steering law in Eq. (9) is that for large tracking errors δx, the corresponding camera speed command $u = \gamma \delta x$ will saturate the velocity servo system

that is trying to implement these speed commands. This saturation will cause non-trivial interaction between the visual servo control u and the velocity servo subsystem. To avoid asking for speed commands that are not physically achievable, the gain γ could be reduced. However, this will result in very slow error cancellation for small tracking errors. A more common method to deal with this issue is to clip or saturate the control signal u itself. This will guarantee that the value of u is never larger than what is physically achievable by the remote manipulator system that is moving the camera. To achieve this, the introduction of the function $f()$ allows us to *smoothly* limit the values of the control variable u. If we assume that the function is bounded by ± 1 through

$$\lim_{\delta x \to \pm\infty} f(\delta x) = \pm 1 \tag{13}$$

then the maximum value that the speed control in Eq. (7) can achieve is γ. Thus, the control parameter γ is referred to as the maximum speed gain. When tuning the servo control parameters, the parameter γ should be set just small enough such that the velocity servo subsystem does not become saturated. If needed, this allows the visual servoing system to exploit the physically available manipulator speed.

There is an infinite choice of functions that satisfy the two conditions in Eq. (8) and (13). For example, using the $\arctan()$ function, we can define $f(\delta x)$ to be

$$f(\delta x) = \frac{2}{\pi} \arctan\left(\frac{\alpha}{\gamma}\frac{\pi}{2}\delta x\right) \tag{14}$$

Note that as $\delta x \to \pm\infty$, then $f(\delta x) \to \pm 1$. The parameter $\alpha > 0$ determines how fast the saturation occurs. The larger this value is, the more sensitive the steering law becomes. If α is very large, then this control essentially becomes a bang-bang control about the target state. Using this smooth saturation function, the final nonlinear velocity steering law is given by

$$\boxed{u = \gamma\frac{2}{\pi} \arctan\left(\frac{\alpha}{\gamma}\frac{\pi}{2}\delta x\right)} \tag{15}$$

Figure 3 illustrates the control law in Eq. (15). Note that the zero crossing slope is given through the parameter α, while the maximum possible speed magnitude is determined through γ. As such, γ should be set less than the velocity limit, and α should be set less than the acceleration limit of the velocity servo sub-system.

In the end game of the regulation control, when the tracking errors δx can be assumed to be very small, the speed commands can be linearized to the form

$$u \approx \alpha \delta x \tag{16}$$

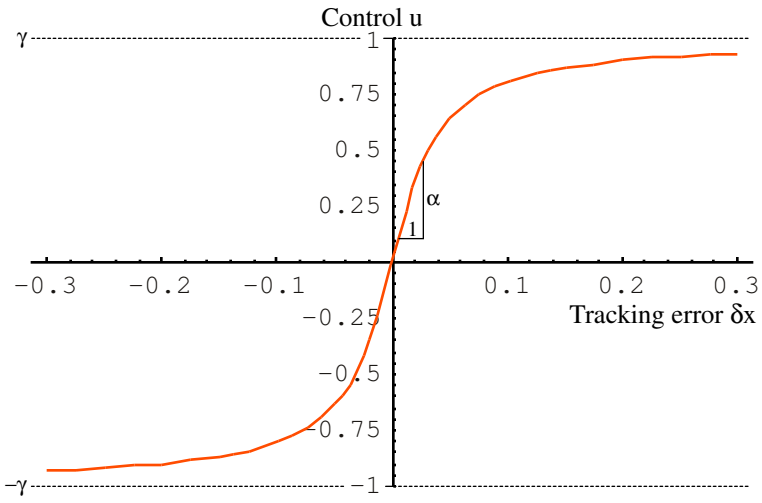

Figure 3. Illustration of the Saturated Control Signal

Thus, the gain α is the stiffness or feedback gain of a classic linear proportional feedback control law. Note that α determines how fast the speed command will change about the zero crossing. If α is too large, then the change in commanded speed will saturate the acceleration of the velocity servo subsystem. The gain α should be kept small enough to avoid acceleration saturation.

To prove that the nonlinear steering law in Eq. (9) is asymptotically stable if $\dot{X} = 0$, we use Lyapunov stability theory. Consider the positive definite Lyapunov function V:

$$V(\delta x) = \frac{1}{2}\delta x^2 \tag{17}$$

If the derivative $\dot{V}$ is negative definite in δx, then the system is said to be asymptotically stabilizing. Computing the derivative of V we find

$$\dot{V}(\delta x) = \delta x \cdot \delta \dot{x} \tag{18}$$

Substituting the camera equations of motion given in Eq. (6) and the control u expressed Eq. (9), we find

$$\dot{V}(\delta x) = \delta x(-u) = -\gamma \delta x \cdot f(\delta x) \tag{19}$$

Using the $f(\delta x)$ function property in Eq. (8), we find that

$$\dot{V}(\delta x) \leq 0 \tag{20}$$

and thus $\dot{V}$ is negative definite in δx. Thus for any control u with $f()$ satisfying Eq. (8), we are guaranteed asymptotic stability of the nonlinear closed loop dynamics. Note that the condition in Eq. (13) does not need to be satisfied for the presented nonlinear control to be stable. The $f(\delta x)$ function property in Eq. (13) is introduced to provide the control gain γ with the physically intuitive interpretation of being the maximum allowable speed command.

Assuming a small tracking error δx and general inertial target motion $\dot{X}$, then the closed-loop error dynamics are defined through

$$\delta\dot{x} + \alpha\delta x \approx \dot{X} \tag{21}$$

The steady-state tracking error of the nonlinear control for a constant $\dot{X}$ is then approximated by

$$\delta x_{ss} \approx \frac{\dot{X}}{\alpha} \tag{22}$$

Again, the larger the gain α is, the smaller the tracking errors will be for a constant $\dot{X}$ target motion.

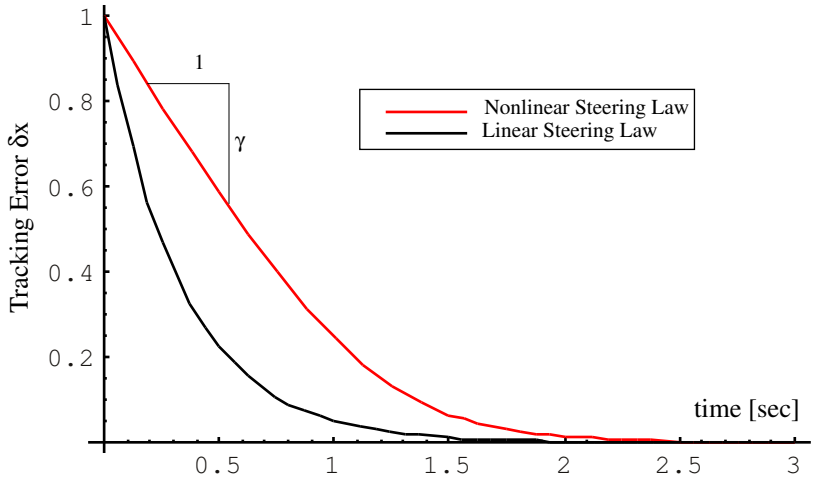

Figure 4. Error Dynamics Comparison of Linear and Nonlinear Steering Law

Figure 4 illustrates the performance difference between the linear steering law in Eq. (9) and the speed limited, nonlinear steering law in Eq. (15). While the linear steering law will cause the tracking errors to exponentially decay to zero, it will also demand very large servo

velocities to achieve this (steep slope near the initial tracking error). Compare this to the performance of the nonlinear steering law. Here the speed command is smoothly limited to be γ or less in magnitude. This results in the initial tracking error decaying essentially linearly. Once the error is small enough to where the commanded speed limiting is no longer active, then the error automatically begins to decay in a near-exponential manner to zero.

Note that because the error dynamics are a first order differential equations, given a perfect velocity servo subsystem, the tracking error dynamics will never overshoot the target states. This greatly simplifies the control gain selection process. The gain γ is chosen large enough without saturating the manipulator speed capability, while α is chosen to be large enough without causing saturation in the manipulator acceleration capability. As long as the manipulator motion is not saturated, the visual servo tracking error response will appear to be similar to a critically damped second order system.

The final implementation of the nonlinear steering law includes a smoothing parameter during the start-up, whose purpose is to avoid hitting the manipulator servo subsystem with a large instantaneous change in commanded speed. The parameter smoothly increases from 0 to 1, causing the commanded speeds to smoothly vary between 0 and their final value determined through the control laws shown above. Further, a deadband is introduced into the $f(\delta x)$ function. If the tracking errors fall within a certain tolerance, then a zero speed command will be sent to the manipulator velocity servo system.

Application of the Snake-Based Visual Servoing

The steering law presented in Eq. (15) was tested initially on a simple pan and tilt mounted camera system. The target is a red square surrounded by a light-blue border. Note that only the snake center of mass (x,y) coordinates are controlled here. The snakes and the visual servo steering law are implemented in the UMBRA frame-work. Both are programmed as UMBRA modules tied together though the connector class. The velocity servo sub-system was also implemented as an UMBRA module, which integrates the speed commands and sends the stepper motor of the pan and tilt unit new position commands. While not a particularly smooth velocity servo sub-system, it is still very effective in demonstrating the snake-based visual servo steering law.

A Tcl script is created that will first search the video image for a region where the target color appears. Then, a statistical pressure snake is automatically created about this region to track the visual target (defined through some HSV colors). Once the snake has converged, the servo control law in Eq. (15) is invoked to bring the target center to the

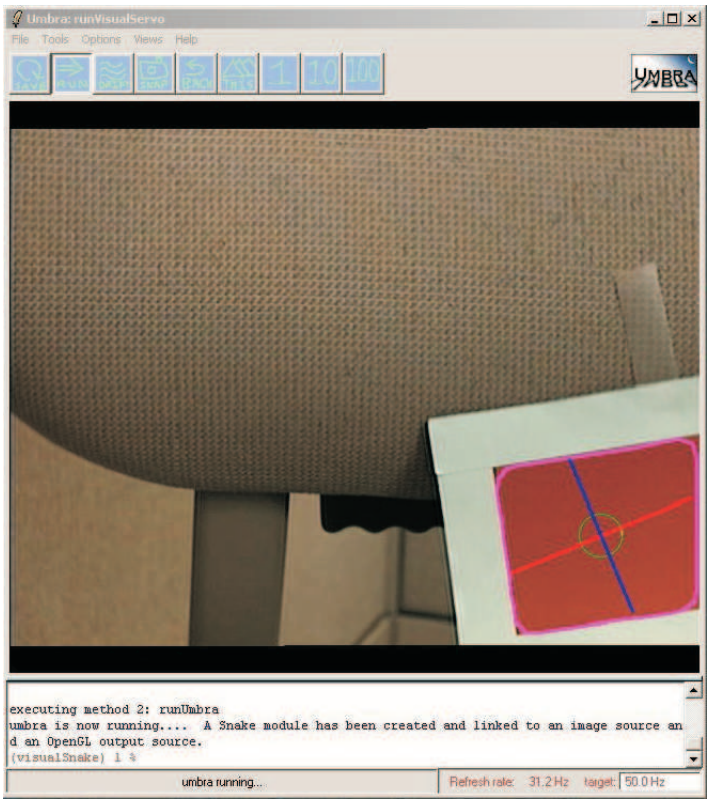

Figure 5. Initial Acquisition of Target by Visual Snake

image center.

Figure 5 shows the red target initially at the lower right corner of the camera image. The Tcl algorithm has already determined the rough location of the target and created a visual snake that has converged onto the visible target shape. After engaging the visual steering law in Eq. (15), the camera is commanded to move such that the snake center of mass is aligned with the image center. A deadband of 1 pixel was used in this application. Note that the snake is capable of computing the snake CM to better than pixel accuracy. The stepper motor used of the pan and tilt unit has an angular resolution of about 0.01 degrees. The converged result is shown in Figure 6.

The speed commands sent to the pan and tilt velocity sub-system are shown in Figure 7, while the tracking errors are illustrated in Figure 8. When activated, the commands are smoothly increased from zero to their final values over the first 1-2 seconds. After this, the speed commands are gradually decreased until they are terminated when the target has

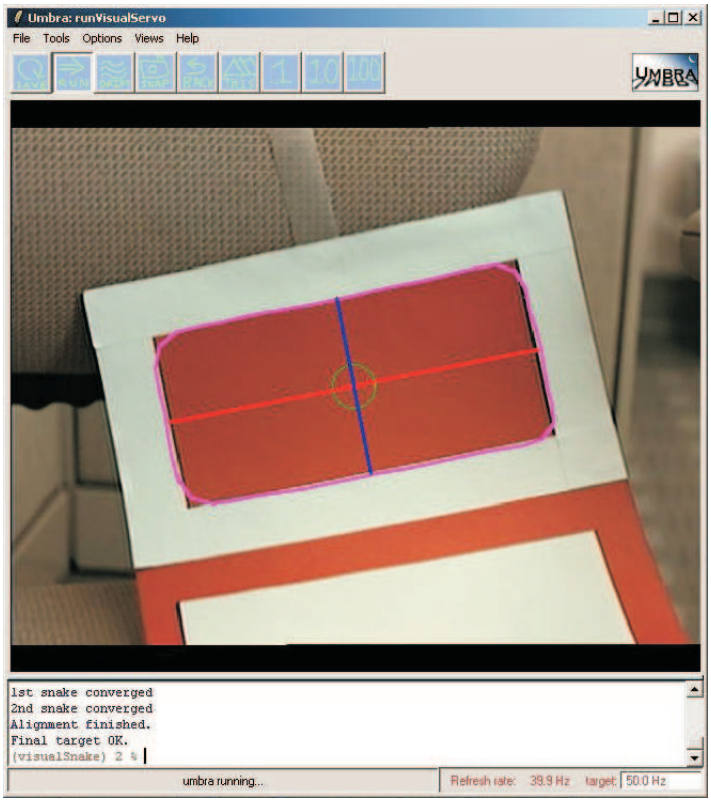

Figure 6. Target at the End of the Visual Servoing Maneuver

converged within the deadband region of the target pixel location. Note that as the camera moves, the dynamic snake curve will react as well. However, the error signals are still reasonably smooth and all states converge to their desired values.

Conclusion

This report presents the nonlinear visual servoing steering law which is used to align a camera view with a visual target. Statistical pressure snakes are used to segment the video frame and identify the target location. The nonlinear steering law provides camera-frame centric speed commands to a velocity based servo sub-system. To avoid saturating the subsystem, the commanded speeds are smoothly limited to remain within a finite range. A small error analysis is also provided illustrating how the two control gains contribute to the

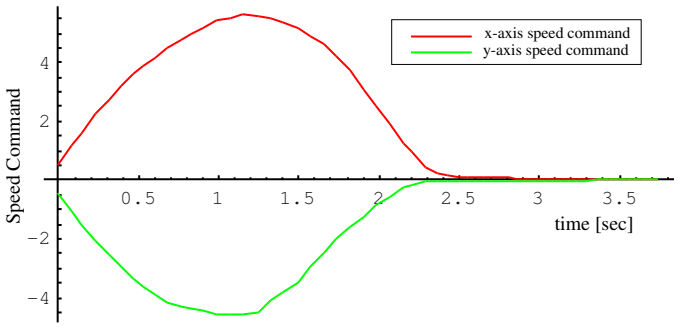

Figure 7. Speed Command Time Histories for Servo Maneuver

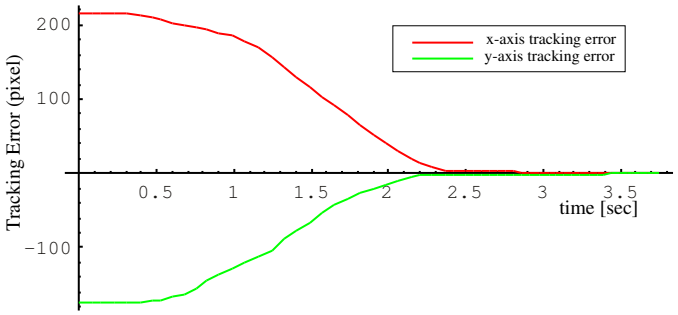

Figure 8. Tracking Error Time Histories for Servo Maneuver

stiffness of the control. A practical example is shown illustrating how the snake degrees of freedom and driven to desired values.

References

[1] Perrin, D. P. and Smith, C. E., “Rethinking Classical Internal Forces for Active Contour Models,” *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, December 2001, pp. 615–620.

[2] Schaub, H., “Statistical Pressure Snakes Based on Color Images,” Technical report, Sandia National Labs, Albuquerque, NM, 2003.

[3] Schaub, H., “Extracting Primary Features of a Statistical Pressure Snake,” Technical report, Sandia National Labs, Albuquerque, NM, 2003.

DISTRIBUTION:

1 Virginia Polytechnic Institute
Attn: Hanspeter Schaub
Aerospace and Ocean Engineering Department
228 Randolph Hall
Blacksburg, VA 24061-0203

1 MS 1125
Phil Bennett, 15252

1 MS 1125
Dan Marrow, 15252

1 MS 1125
Robert J. Anderson, 15252

1 MS 1003
Chris Wilson, 15212

1 MS 9018
Central Technical Files, 8940-2

2 MS 0899
Technical Library, 4916

2 MS 0619
Review & Approval Desk, 4916