Proceedings of 2014 IEEE
International Conference on Mechatronics and Automation
August 3 - 6, Tianjin, China

Kalman Filter Visual Servoing Control Law

Matthew Marshall
Department of Electrical and Mechatronics Engineering
Southern Polytechnic State University
Marietta, GA, USA
mqm@spsu.edu

Harvey Lipkin
School of Mechanical Engineering
Georgia Institute of Technology
Atlanta, GA, USA
harvey.lipkin@me.gatech.edu

***Abstract*—This paper introduces a Kalman filter based visual servoing control method that reduces noise sensitivity. It is shown to be completely controllable and observable under certain mild conditions. Visual servoing simulations are performed for a six-axis robot manipulator with both moving and static targets. The controller, if tuned properly, yields equivalent performance to Gauss-Newton for low-noise scenarios and improved performance in the presence of increased camera noise.**

***Index Terms*—Visual servoing, Kalman filter, control law**

I. Introduction

Integrating visual sensing is one way to enhance robot capabilities. When done in a closed-loop form this control is referred to as visual servoing (VS). Applications include agricultural tasks, food processing, explosive ordinance disposal, and disaster-zone inspections.

In VS systems cameras can be either mounted on the robot end-effector (eye-in-hand) or at a distance such that the end-effector of the robot is in the field of view (eye-to-hand). The camera data is used to track either the robot end-effector, a target object for the robot to follow, or both by identifying a set of features in the images. The aim of VS is to actuate the robot so that the features $\boldsymbol{y}$ from the vision system match a desired set $\boldsymbol{y}^*$. In other words, the robot joint displacements $\boldsymbol{\theta}$ are controlled to minimize the error

$$\boldsymbol{f}(\boldsymbol{\theta}, t) \equiv \boldsymbol{y}(\boldsymbol{\theta}, t) - \boldsymbol{y}^*(\boldsymbol{\theta}, t) \tag{1}$$

VS has been an active field of study for decades and it is commonly thought of as having two branches: image-based and position-based. Image-based visual servoing (IBVS) forms $\boldsymbol{y}$ based only on data available at the camera level (coordinates in the image space). In position-based visual servoing (PBVS) $\boldsymbol{y}$ comprises Cartesian coordinates with respect to a global frame, thus requiring knowledge of the camera position. There are advantages and disadvantages to each method [3].

A different VS taxonomy is calibrated or uncalibrated. Calibration refers to kinematic and optical parameters, for example link lengths of the robot, location of the camera, or pitch of the sensor (pixels per mm). Uncalibrated visual servoing (UVS) is able to control the robot without knowledge of these parameters.

Uncalibrated visual servoing is well-suited to unstructured environments because determining robot and camera parameters can be difficult therein. UVS is by necessity image-based. In UVS the effect of robot joint motions on image-plane coordinates is estimated through measurements of image feature coordinates and robot joint positions, rather than by analytical means.

When precise calibration is available, calibrated VS achieves better performance over UVS in metrics such as settling time and precision [14]. However, two cameras can yield better performance than one so when uncalibrated VS is required its performance can be improved by adding a camera [10]. To the authors' best knowledge the present research is the first to simultaneously employ more than two cameras in UVS. Gao and Su perform UVS with three cameras for a mobile robot however the system uses only two cameras at a time [7].

In IBVS the measurements are traditionally the image-plane coordinates of a set of points. Alternatives to using points as VS features have been studied [4], [19]. However, all of these approaches use the robot kinematic parameters. In an approach using no robot or extrinsic camera parameters, Marshall et al. [12] employ the relative Cartesian displacements of an eye-in-hand 3-D time-of-flight camera as feature measurements. The current research uses points as features and neither robot nor camera parameters.

There are two main components in a visual-servoing system: Jacobian estimation and the control law. The Jacobian relates robot velocity in joint space to the velocity of the end-effector in the image space. Once this relationship is estimated the control law determines an action that best reduces $\boldsymbol{f}$. Most VS literature focuses on the first component, Jacobian estimation, and this is especially true for uncalibrated VS.

To guide a robot based on data from a camera the relationship between joint displacements and image-plane displacements must be modeled. The relationship is nonlinear and can be high dimensional. A linearization of this map is called the (composite) Jacobian matrix, J, where

$$J \equiv \frac{\partial \boldsymbol{y}}{\partial \boldsymbol{\theta}}$$

978-1-4799-3979-4/14/$31.00 ©2014 IEEE

Formulating this matrix can be done analytically using the system physical parameters including robot kinematics, camera location, lens focal length and radial distortion parameter, and the size of the pixels. Alternatively the mapping can be estimated numerically.

Using a numerical approach constitutes uncalibrated visual servoing, introduced by Hosoda and Asada [9]. They estimate the Jacobian through recursive least squares (RLS) techniques and are limited to fixed targets and cameras. Using a different RLS method to estimate the Jacobian, Piepmeier et al. [16] demonstrate the ability to track a moving target with a robot-mounted (thus moving) camera. Qian and Su [17] define state variables formed from elements of the image Jacobian matrix and employ a Kalman filter to estimate them online. Large performance losses are noted in this type of controller as system noise increases [18].

The algorithm in Piepmeier et al. [15] is used herein for the Jacobian estimation. It has been shown that the population-based method for Jacobian estimation presented by Bonkovic et al. [1] tracks a moving target better than the recursive approach of Piepmeier, though the latter performs better for stationary targets and yields smoother robot motion [8]. A reason for eschewing the population-based method especially for multi-camera visual servoing is that population length increases with the number of images and this leads to computational requirements comparable to the cycle time of the robotic system.

The control law determines robot commands based on feedback and the state of the system. In visual servoing the state pertains to the positions (possibly also velocities and accelerations) of the robot and target as measured by the robot joint encoders and by machine vision. In a direct visual servo system commands are in the form of joint torques/forces with Wang et al. [19] as an example. The more typical dynamic look-and-move paradigm commands joint positions (or velocities) and there are many different such control laws in the literature. The most common VS control law is the Gauss-Newton (GN) algorithm detailed in §II-A. Alternatives exist to achieve various ends. Unique control laws that have been developed are only applicable to calibrated systems, for example [4] and [5]. The present research introduces a Kalman filter based control law that is applied to UVS and outperforms GN.

This paper is divided as follows. §II describes a traditional control law that uses the Gauss-Newton method and discusses the Kalman filter application to visual servo control. Stability of the Kalman filter method is assessed and the resulting system is shown to be observable and controllable. §III provides details of the physical system modeled in simulation. §IV provides simulation results and discussion. §V concludes the paper and offers direction for future work.

II. Control Law

This section presents a visual servoing control law using the Kalman filter, which estimates a changing state based on a process model and noisy measurements. In the current context the filter estimates the joint-space distance (and possibly its time derivative) between the robot and the target. Given accurate statistical process and measurement error models the filter is known to provide optimal estimates, minimizing the expected estimate error. The Kalman filter's recursive nature provides control less sensitive to noise than traditional UVS methods.

A. Traditional control laws

For comparison, the most common approach to designing a VS control law is discussed. It uses a version of Newton's root finding method to minimize the error $\boldsymbol{f}$ between robot and target image-plane coordinates. The Gauss-Newton variation is favored in practice because the difficult Hessian matrix ($H = \frac{\partial^2 \boldsymbol{y}}{\partial^2 \boldsymbol{\theta}} = \frac{\partial J}{\partial \boldsymbol{\theta}}$) estimation is not required.

Visual servoing systems are often over constrained where the number of image features is greater than the robot degrees of freedom, and so in general (due to camera noise) there is no robot position that makes the error zero. Therefore, the computation of the desired joint positions is routinely treated as a minimization problem, usually of the error-squared objective function G,

$$G(\boldsymbol{\theta}, t) = \boldsymbol{f}^{\mathsf{T}}(\boldsymbol{\theta}, t) W \boldsymbol{f}(\boldsymbol{\theta}, t)$$

where the weighting matrix W typically is the identity matrix I and $\boldsymbol{f}$ is the image-plane error defined in (1).

The robot position that solves the minimization problem at a given time t_k is denoted $\boldsymbol{\theta}^*$,

$$G(\boldsymbol{\theta}^*, t_k) = \min G(\boldsymbol{\theta}, t_k) \tag{2}$$

and can be estimated via Newton's method, yielding what has been called the dynamic Newton's method

$$\widehat{\boldsymbol{\phi}}_k = \left(J_k^{\mathsf{T}} J_k + S_k\right)^{-1} \left(J_k^{\mathsf{T}} \boldsymbol{f}_k - J_k^{\mathsf{T}} \frac{\partial \boldsymbol{y}^*(t)}{\partial t} h_t\right) \tag{3}$$

where h_t is the time step, $S_k = \frac{\partial J_k^{\mathsf{T}}}{\partial \boldsymbol{\theta}} \boldsymbol{f}_k$, and

$$\widehat{\boldsymbol{\phi}} \equiv \boldsymbol{\theta}_k - \widehat{\boldsymbol{\theta}_k^*}$$

is the estimated joint-space vector between the current robot position and the one that minimizes $\boldsymbol{f}^{\mathsf{T}} \boldsymbol{f}$, that is, the goal robot position $\boldsymbol{\theta}^*$ introduced in (2).

The term $\frac{\partial \boldsymbol{y}^*(t)}{\partial t} h_t$ in (3) facilitates servoing to moving targets. Leaving it out gives the more standard Newton's equation for visual servoing

$$\widehat{\boldsymbol{\phi}}_k = \left(J_k^{\mathsf{T}} J_k + S_k\right)^{-1} J_k^{\mathsf{T}} \boldsymbol{f}_k \tag{4}$$

With uncalibrated VS the Jacobian J is estimated as $\hat{J}$. The S term in (4) and (3) is both difficult to compute and small when near the target so it is often left out, yielding the Gauss-Newton method.

$$\widehat{\boldsymbol{\phi}}_k = \left(J_k^\mathsf{T} J_k\right)^{-1} J_k^\mathsf{T} \boldsymbol{f}_k$$

An exception to this practice is the work of Munnae [13] who retains the full Hessian matrix $G_{\theta\theta} = J_k^\mathsf{T} J_k + \frac{\partial J_k^\mathsf{T}}{\partial \boldsymbol{\theta}} \boldsymbol{f}_k = J_k^\mathsf{T} J_k + S_k$ and as the robot approaches its goal uses heuristic criteria to switch from an initial quasi-Newton method to a Gauss-Newton method.

Besides Newton and Gauss-Newton, other minimization-based control laws are described in the literature. An example is the Jacobian Transpose control (or steepest descent) method.

$$\widehat{\boldsymbol{\phi}}_k = J_k^\mathsf{T} \boldsymbol{f}_k$$

However, the Gauss-Newton control law is the most common in visual servoing. For example, Deng and Jägersand [6] use it to compare three different Jacobian estimation methods. It therefore serves as the benchmark for the Kalman filter control law described below.

B. Kalman filter visual servoing control law

The Kalman equations recursively solve weighted least squares for a system with the following process and observation models,

$$\boldsymbol{x}_k = F_{k-1}\boldsymbol{x}_{k-1} + B_k \boldsymbol{u}_k + \boldsymbol{w}_k, \quad \boldsymbol{w}_k \sim N(\boldsymbol{0}, Q_k) \tag{5}$$

$$\boldsymbol{z}_k = H_k \boldsymbol{x}_k + \boldsymbol{v}_k, \quad \boldsymbol{v}_k \sim N(\boldsymbol{0}, R_k) \tag{6}$$

where $N(\boldsymbol{\mu}, \Sigma)$ refers to Gaussian noise with expected value $\boldsymbol{\mu}$ and covariance Σ.

At time k, the Kalman filter provides an estimate $\hat{\boldsymbol{x}}_{k|k}$ and its error covariance matrix $P_{k|k}$ according to

$$\hat{\boldsymbol{x}}_{k|k-1} = F_{k-1}\hat{\boldsymbol{x}}_{k-1|k-1} + B_k \boldsymbol{u}_k \tag{7}$$

$$P_{k|k-1} = F_{k-1} P_{k-1|k-1} F_{k-1}^\mathsf{T} + Q_k \tag{8}$$

$$K_k = P_{k|k-1} H_k^\mathsf{T} \left(H_k P_{k|k-1} H_k^\mathsf{T} + R_k\right)^{-1} \tag{9}$$

$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + K_k \left(\boldsymbol{z}_k - H_k \hat{\boldsymbol{x}}_{k|k-1}\right) \tag{10}$$

$$P_{k|k} = \left(I - K_k H_k\right) P_{k|k-1} \tag{11}$$

The prediction step comprises (7) and (8), and the correction step comprises (9)–(11).

The state $\boldsymbol{x}$ being estimated is based on the difference between the current and goal robot positions $\boldsymbol{\phi} = \boldsymbol{\theta} - \boldsymbol{\theta}^*$ of (2). This joint-space error generally is a function of both $\boldsymbol{\theta}$ and time.

The measurement $\boldsymbol{z}$ for an image-based visual servoing system is the image-plane error $\boldsymbol{f}$. It is necessary to relate the measurements to the state in order to use the Kalman filter. The measurement model uses the following approximation to (1)

$$\boldsymbol{f}_k \equiv \boldsymbol{y}_k - \boldsymbol{y}_k^* \approx J_k(\boldsymbol{\theta}_k - \boldsymbol{\theta}_k^*) = J_k \boldsymbol{\phi}_k \tag{12}$$

where J_k is the composite Jacobian at time k mapping robot velocity in joint space to the velocity of the end-effector in the image space. Due to non-linear robot kinematics the approximation (12) generally is valid only in the area around $\boldsymbol{\theta}_k$.

Here the system (5) and (6) is formulated for IBVS in four ways depending on the order of the state space representation and whether input $\boldsymbol{u}$ is included. For clarity the state and measurement variables $\boldsymbol{x}$ and $\boldsymbol{z}$ are replaced with $\boldsymbol{\phi}$ (or $\begin{bmatrix}\boldsymbol{\phi}^\mathsf{T} & \dot{\boldsymbol{\phi}}^\mathsf{T}\end{bmatrix}^\mathsf{T}$) and $\boldsymbol{f}$.

a) Zeroth order without input: The process equation

$$\boldsymbol{\phi}_k = \boldsymbol{\phi}_{k-1} + \boldsymbol{w}_k$$

predicts constant joint-space error $\boldsymbol{\phi}$. The measurement equation is

$$\boldsymbol{f}_k = J_k \boldsymbol{\phi}_k + \boldsymbol{v}_k$$

b) Zeroth order with input: The process equation

$$\boldsymbol{\phi}_k = \boldsymbol{\phi}_{k-1} + \boldsymbol{h}_{\theta,k} + \boldsymbol{w}_k$$

predicts a stationary target position $\boldsymbol{\theta}^*$. The measurement equation is

$$\boldsymbol{f}_k = J_k \boldsymbol{\phi}_k + \boldsymbol{v}_k$$

c) First order without input: The process equation

$$\begin{bmatrix}\boldsymbol{\phi}_k \\ \dot{\boldsymbol{\phi}}_k\end{bmatrix} = \begin{bmatrix} I & h_t I \\ 0 & I \end{bmatrix} \begin{bmatrix}\boldsymbol{\phi}_{k-1} \\ \dot{\boldsymbol{\phi}}_{k-1}\end{bmatrix} + \boldsymbol{w}_k$$

predicts a constant joint-space error velocity $\dot{\boldsymbol{\phi}}$. The measurement equation is

$$\boldsymbol{f}_k = \begin{bmatrix} J_k & 0 \end{bmatrix} \begin{bmatrix}\boldsymbol{\phi}_k \\ \dot{\boldsymbol{\phi}}_k\end{bmatrix} + \boldsymbol{v}_k$$

d) First order with input: The process equation

$$\begin{bmatrix}\boldsymbol{\phi}_k \\ \dot{\boldsymbol{\phi}}_k\end{bmatrix} = \begin{bmatrix} I & h_t I \\ 0 & I \end{bmatrix} \begin{bmatrix}\boldsymbol{\phi}_{k-1} \\ \dot{\boldsymbol{\phi}}_{k-1}\end{bmatrix} + \begin{bmatrix} 0 \\ \boldsymbol{h}_{\omega,k}\end{bmatrix} + \boldsymbol{w}_k$$

predicts a constant-joint-space-velocity target. Two other input forms are possible but yield unlikely predictions [11]. The measurement equation is

$$\boldsymbol{f}_k = \begin{bmatrix} J_k & 0 \end{bmatrix} \begin{bmatrix}\boldsymbol{\phi}_k \\ \dot{\boldsymbol{\phi}}_k\end{bmatrix} + \boldsymbol{v}_k$$

C. Control action

The current joint-space error estimate $\widehat{\phi}_k$ generated by the control law is used to set the robot joint offset command as

$$\Delta\boldsymbol{\theta}_{k+1} = \begin{cases} -\widehat{\phi}_{k|k}, & \text{if} \quad \left\|\widehat{\phi}_{k|k}\right\| < \mu \\ -\mu \dfrac{\widehat{\phi}_{k|k}}{\left\|\widehat{\phi}_{k|k}\right\|}, & \text{otherwise} \end{cases} \tag{13}$$

where μ is the maximum allowable joint offset norm, which is necessary to keep the robot motion confined to one VS control period.

D. Kalman filter control law stability

It is assumed that the joint-level servoing can attain any desired joint offset in one time step for sufficiently small μ. Thus, the state equation for the robot position is

$$\boldsymbol{\theta}_{k+1} = F\boldsymbol{\theta}_k + B\Delta\boldsymbol{\theta}_{k+1}$$

and the *reachability matrix* $C = \begin{bmatrix} B & FB & F^2B & \dots & F^{a-1}B \end{bmatrix}$ has rank a, the robot degrees of freedom, because both F and B are the $a \times a$ identity matrix. A full-rank reachability matrix is the criterion for a fully controllable discrete linear time-invariant system. Therefore the robot is completely controllable under the assumption above.

The Kalman filter acts as an observer of the state $\boldsymbol{x} = \boldsymbol{\phi}$ (or $\begin{bmatrix} \boldsymbol{\phi}^\mathsf{T} & \dot{\boldsymbol{\phi}}^\mathsf{T} \end{bmatrix}^\mathsf{T}$ for first order representation). A linear system is said to be *observable* at t_0 if $\boldsymbol{x}_0$ can be determined from the output sequence $\boldsymbol{z}_0 \dots \boldsymbol{z}_j$ for $t_0 \leq t_j$, where t_j is some finite time. If this is true for all t_0 and x_0 then the system is said to be *completely observable* [2]. This property ensures the convergence of P and bounded uncertainty (that is, error covariance).

In order for the system (5) and (6) to be completely observable at $k = 0$ there must exist some finite index N such that the square matrix

$$X = \sum_{k=0}^{N} \Phi_{k,0}^\mathsf{T} H_k^\mathsf{T} H_k \Phi_{k,0} \tag{14}$$

is positive definite, where Φ is the transition matrix. Provided the input sequence $\boldsymbol{u}_0 \dots \boldsymbol{u}_k$ is zero then given the state $\boldsymbol{x}_j$, the state at any other time k is given by the mapping $\boldsymbol{x}_k = \Phi_{k,j}\boldsymbol{x}_j$ [2].

For the formulations of §II-B the process matrix F is constant. In such a case the transition matrix is $\Phi_{k,j} = F^{k-j}$ and (14) reduces to

$$X = \sum_{k=0}^{N} \left[\left(F^k\right)^\mathsf{T} H_k^\mathsf{T} H_k F^k \right]$$

This observability criterion is examined for the two state space representations of §II-B and sufficient conditions are derived to ensure positive definite X for all $N \geq 1$.

e) Zeroth order representation: In this case $F = I$ and $H_k = J_k$ so $X = \sum_{k=0}^{N} J_k^\mathsf{T} J_k$. To ensure positive definite X a sufficient condition is that J_l has full column rank for some l, where $0 \leq l \leq N$. The robot must be in a full column rank configuration at least once during servoing. In that case X is the sum of a positive definite matrix and N positive semidefinite matrices and is therefore positive definite.

f) First order representation: In this case $F = \begin{bmatrix} I & h_t I \\ 0 & I \end{bmatrix}$ and $H_k = \begin{bmatrix} J_k & 0 \end{bmatrix}$, where $0 < h_t \leq 1$, which leads to the block matrix

$$X = \sum_{k=0}^{N} \begin{bmatrix} J_k^\mathsf{T} J_k & k h_t J_k^\mathsf{T} J_k \\ k h_t J_k^\mathsf{T} J_k & k^2 h_t^2 J_k^\mathsf{T} J_k \end{bmatrix}$$

X is positive definite if J_0 and some J_m, $0 < m \leq N$ are both full column rank.

Proof.

$$\begin{aligned} X = &\begin{bmatrix} J_0^\mathsf{T} J_0 + J_m^\mathsf{T} J_m & mhJ_m^\mathsf{T} J_m \\ mhJ_m^\mathsf{T} J_m & m^2 h^2 J_m^\mathsf{T} J_m \end{bmatrix} + \\ &\sum_{k=1}^{m-1} \begin{bmatrix} J_k^\mathsf{T} J_k & k h_t J_k^\mathsf{T} J_k \\ k h_t J_k^\mathsf{T} J_k & k^2 h_t^2 J_k^\mathsf{T} J_k \end{bmatrix} + \\ &\sum_{k=m+1}^{N} \begin{bmatrix} J_k^\mathsf{T} J_k & k h_t J_k^\mathsf{T} J_k \\ k h_t J_k^\mathsf{T} J_k & k^2 h_t^2 J_k^\mathsf{T} J_k \end{bmatrix} \end{aligned} \tag{15}$$

Let $M_k \equiv \begin{bmatrix} J_k & k h_t J_k \end{bmatrix}$, then the block matrices for $k \neq 0$, m of (15) are $M_k^\mathsf{T} M_k$ and thus are always positive semidefinite. Further, let

$$A = \begin{bmatrix} J_0^\mathsf{T} J_0 + J_m^\mathsf{T} J_m & mhJ_m^\mathsf{T} J_m \\ mhJ_m^\mathsf{T} J_m & m^2 h^2 J_m^\mathsf{T} J_m \end{bmatrix}$$

and subtract $1/mh$ times column 2 of A from the first column of A to get A'

$$A' = \begin{bmatrix} J_0^\mathsf{T} J_0 & mhJ_m^\mathsf{T} J_m \\ 0 & m^2 h^2 J_m^\mathsf{T} J_m \end{bmatrix}$$

which has the same determinant as A. The determinant of a block triangular matrix is the product of the determinants of its diagonal entries, so

$$\det(A) = \det(A') = |J_0^\mathsf{T} J_0| \cdot |m^2 h^2 J_m^\mathsf{T} J_m| > 0$$

since $J_0^\mathsf{T} J_0$ and $J_m^\mathsf{T} J_m$ are both positive definite. Since A is therefore positive definite and the block matrices in the summation (15) are positive semidefinite then X is positive definite.

□

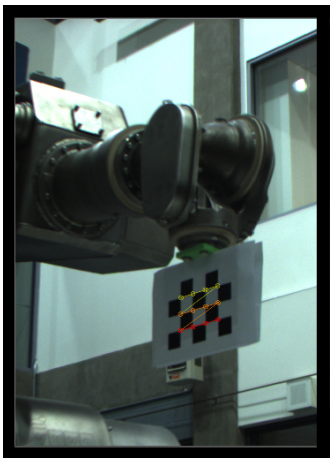

Fig. 1. Image feature extraction comprising twelve checkerboard vertices constituting $\boldsymbol{y}$ (goal coordinates not shown).

It follows that the zeroth order systems are completely observable if the robot passes through a full column rank configuration at least once during servoing. Complete observability of the first order systems necessitates the additional condition that servoing begin with the robot in a full column rank configuration. That the four formulations are shown to be completely observable implies bounded error on the direction in which the robot is driven via (13).

III. Simulation

The modeled robot manipulator is a KUKA KR 15 SL. It has maximum joint speeds ranging from 156 deg/s (axis 1) to 609 deg/s (axis 6). Robot joint positioning errors are modeled by 0.01 degree white noise for each joint.

Three stationary cameras observe the manipulator so the results presented here are for the eye-to-hand case. Figure 1 shows a checkerboard pattern mounted to the robot tool flange. The image-plane coordinates of the four outermost vertices make up the image feature vector of the i-th camera

$$\boldsymbol{y}^{(i)} = [u_1^{(i)}\ v_1^{(i)}\ u_2^{(i)}\ v_2^{(i)}\ u_3^{(i)}\ v_3^{(i)}\ u_4^{(i)}\ v_4^{(i)}]^\mathsf{T}$$

The goal coordinates $\boldsymbol{y}^*$ are generated analytically as described at the end of this section.

The rectangle described by the four outermost checkerboard vertices is 81×52 mm. Image data are obtained using a pinhole camera model with

$$\begin{bmatrix} u_p \\ v_p \end{bmatrix} = \frac{l}{z_c} \begin{bmatrix} x_c \\ y_c \end{bmatrix}$$

where l is the focal length and $(u_p,\ v_p)$ are the image-plane coordinates of a point located at $[x_c \quad y_c \quad z_c]^\mathsf{T}$ relative to the camera. White noise is added to u_p and v_p in order to more closely approximate real conditions.

In a physical system feature image coordinate noise depends on several sources. Image noise plays a factor; for the same lighting conditions image noise increases with decreasing sensor size or reduction in lighting. Noise is also influenced by feature type. For example, the coordinates for the centroid of a sphere are noisier than those of a checkerboard vertex due to color artifacts and lighting effects at the edge of the sphere image. Physical pixel size (or focal length, or depth) directly affects accuracy. Two camera parameter sets are employed, simulating a high-cost camera and a low-cost camera. The high-cost camera model has a 10 mm focal length and a maximum additive noise magnitude of 0.5 pixels. A low-cost camera likely has a wide-angle lens and a smaller sensor and is modeled as $l = 3.95$ mm with 2 pixels of noise.

Simulations use heterogeneous camera sets to represent unstructured environments, where camera placement options and lighting conditions vary. For a given situation, a camera far from the robot will have a lower signal to noise ratio (SNR) than one close. Poor lighting also degrades the SNR. Thus it is important to simulate using sets combining high- and low-cost camera models. Static- and moving-target simulations use all possible combinations of high- and low-cost cameras for a three-camera system. That is, $C_H + C_L = C = 3$ with $C_H = 0 \ldots 3$, where C_H is the number of high-cost cameras used and C_L is the number of low-cost cameras.

Two different tasks are performed requiring the visual servoing system to either servo to a stationary target or to track a moving one. The starting robot position for the static-target case is offset from the goal position $\boldsymbol{\theta}^*$ by $[-11.79 \quad 18.66 \quad -37.60 \quad -5.6 \quad 24.66 \quad -6.34]^\mathsf{T}$ deg. In Cartesian space the change between these two positions for the checkerboard is a 200 mm translation in each of the three principal directions of the ground coordinate system (346.4 mm total) and a 5.63 deg rotation in three directions: first about the world z-axis, then about the new checkerboard y-axis, and finally about the new checkerboard x-axis. Static-target trials are concluded when either the image-plane error norm in each of the C cameras is less than some threshold value ϵ

$$\left\| \boldsymbol{f}^{(i)} \right\| < \epsilon \qquad \text{for } i = 1 \ldots C \tag{16}$$

or the number of iterations reaches a preset value (six hundred), which is considered failure. For static-target trials $\mu = 1$ degree.

The target joint positions for the moving-target case are determined by the inverse kinematics of the robot controller such that the checkerboard traces the following path:

1) Translate 320 mm in the world z direction
2) Translate 320 mm in the world y direction
3) Translate 320 mm in the world x direction

4) Translate -320 mm in the world z direction while rotating 15 degrees about an axis through the checkerboard center, parallel to world z
5) Translate -320 mm in the world y direction while rotating -15 degrees about an axis through the checkerboard center, parallel to world y
6) Translate -320 mm in the world x direction while rotating 15 degrees about an axis through the checkerboard center, parallel to world x
7) Simultaneously translate 320 mm in each of the principal directions while rotating 15 degrees about the three axes

The initial robot position places the checkerboard coincident with the start of this path. The linear displacement of the checkerboard center between each image acquisition is 20 mm for the first six sections of the path and 34.6 mm in the final section.

For moving-target trials $\mu = 8$ degrees in order to achieve the 100 mm/sec target speed given that three Chameleon cameras connected to a single USB 2.0 bus would operate at 3 Hz.

IV. RESULTS

All four Kalman filter (KF) formulations are tested. Different values of Q and R are tested. Gauss-Newton is tested for comparison. For each camera scenario one hundred trials are run, each with a different random-number-generator seed that is used for:

- The noise added to robot joint positions
- The noise added to the pixel coordinates
- The position and orientation of each camera

When comparing controllers, the same seed is used. For example, Trial 10 of the $C_H = 3$, static-target case using zeroth order KF has the same noise seed as Trial 10 of the $C_H = 3$, static-target case using GN. Ramifications include identically positioned cameras for both trials and identical target pixel coordinates.

A. Results

The robot path for a static-target trial using KF is shown in Figures 2 (Cartesian coordinates) and 3 (image-plane coordinates). The straight lines in Figure 2 represent the ideal (shortest) checkerboard path between the start and end positions. The starting point is at the bottom right in the figure. The final offset between target and servoed checkerboard positions are a function of the threshold value ϵ of (16). The black dots in Figure 3 are the target image coordinates and the colored lines trace the servoed image-plane path in the three cameras. The random-number seed for this trial results in fifty-one steps for convergence with KF (and 256 with GN).

The performance metric for static-target servoing is the average number of iterations to convergence

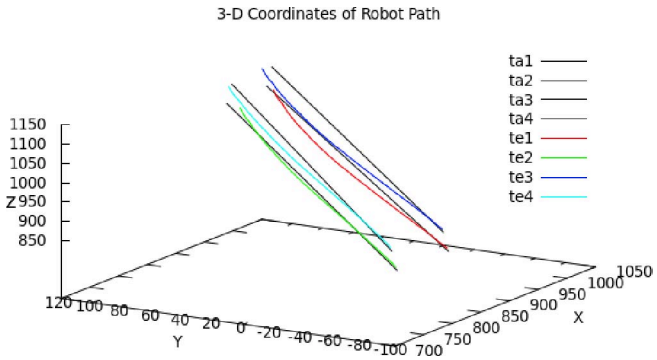

Fig. 2. Exemplar static-target trial paths for checkerboard corners: ideal (ta1–ta4) and servoed (te1–te4) using KF with $Q = 5I$ and $R = I$, one high-cost camera and two low-cost cameras ($C_H = 1$ and $C_L = 2$)

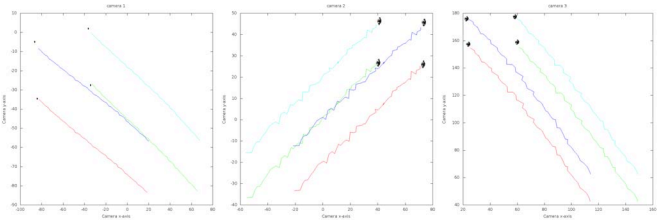

Fig. 3. Image-plane data from three cameras for $C_H = 1$ ("camera 1") and $C_L = 2$ ("camera 2" and "camera 3") static-target trial using KF with $Q = 5I$ and $R = I$

$$\bar{S} = \frac{1}{T}\sum_{l=1}^{T} S_l$$

Here the number of trials is $T = 100$. Convergence means the image-plane error norm is less than ϵ in all $C = 3$ cameras. For these trials the threshold is $\epsilon = 10$ pixels.

Figure 4 shows $\bar{S}$ for the different KF formulations and varied covariance matrices as well as GN in four different camera scenarios, each taken over one hundred trials. These figures show $\bar{S}$ along the vertical axes. Different camera scenarios are listed along the horizontal: "HHH" stands for $C_H = 3$, "HHL" is for $C_H = 2$, etc. The legends describe the control laws used.

For example, in the top-left subfigure the "HLL" columns (one high-cost and two low-cost cameras) show that with $Q = R = I$ the four KF controllers (zeroth and first order, both with and without input) all converge in about eighty iterations while GN requires 110. A missing column in Figure 4 indicates that in one trial the termination threshold was not reached within six hundred iterations for that controller.

Examining Figure 4 reveals that on average KF converges faster than GN for all scenarios tested since $\bar{S}$ is lower in every case. It also shows that the performance gain increases with camera noise. For example, KF and GN take about the

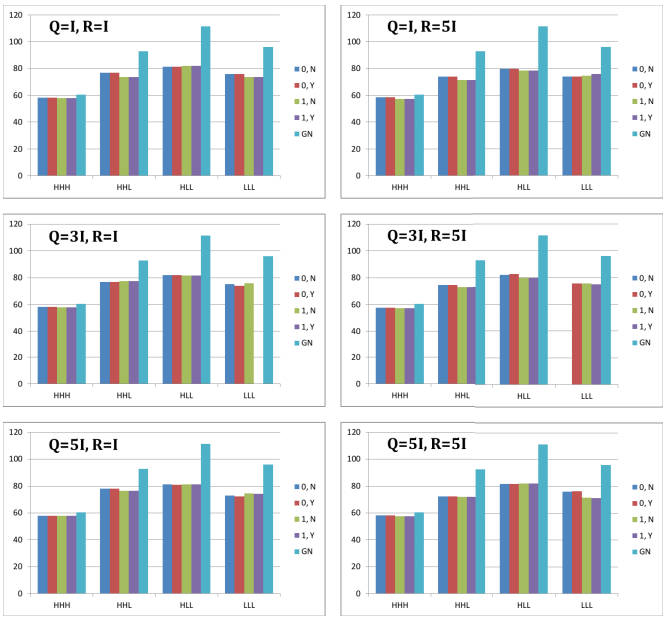

Fig. 4. $\bar{S}$ using combinations of high- and low-cost cameras with GN and 24 KF variations, "0" or "1" refers to the state space representation order and "Y" or "N" refers to with or without input

TABLE I
MINIMUM, AVERAGE, AND MAXIMUM ITERATIONS TO CONVERGENCE S FOR ZEROTH-ORDER KF USING INPUT WITH $Q = 5I$ AND $R = I$ PLUS GN FOR FOUR DIFFERENT CAMERA SCENARIOS.

	$C_H = 3, C_L = 0$			$C_H = 2, C_L = 1$			$C_H = 1, C_L = 2$			$C_H = 0, C_L = 3$		
	$S_{\min}$	$\bar{S}$	$S_{\max}$	$S_{\min}$	$\bar{S}$	$S_{\max}$	$S_{\min}$	$\bar{S}$	$S_{\max}$	$S_{\min}$	$\bar{S}$	$S_{\max}$
$KF_{0,Y,5,1}$	50	58.0	106	50	78	288	47	81.1	305	46	72.4	273
GN	50	60.5	169	49	92.8	274	48	111.3	436	48	96.0	461

same number of steps to reach the target with three high-cost cameras but when two of those are exchanged for noisier cameras then average KF convergence is 37% faster than GN.

Table I provides the minimum and maximum number of iterations out of the hundred trials for two controllers in each of the four camera scenarios. This reveals that not only is KF better on average but also it yields more consistent performance as evidenced by the smaller outliers (for example, when using three low-cost cameras $S_{\max} = 273$ for KF and $S_{\max} = 461$ for GN). Also, the disparity between $S_{\min}$ and $S_{\max}$ for a given set of cameras indicates that UVS performance is dependent on camera placement since that is different in every one of the one hundred trials performed with each controller.

The robot path for a moving-target trial using KF is shown in Figures 5 (Cartesian coordinates) and 6 (image-plane coordinates). The black lines in Figure 5 represent the target checkerboard path.

The performance metric for moving-target servoing is the Cartesian error norm for the checkerboard corners, averaged over all $T \times S$ iterations, where T is the number of trials run

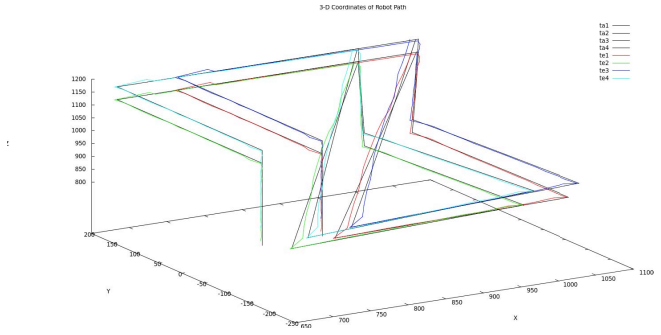

Fig. 5. Exemplar moving-target trial paths for checkerboard corners: ideal (ta1–ta4) and servoed (te1–te4) using KF with $Q = 5I$ and $R = I$, three high-cost cameras ($C_H = 3$)

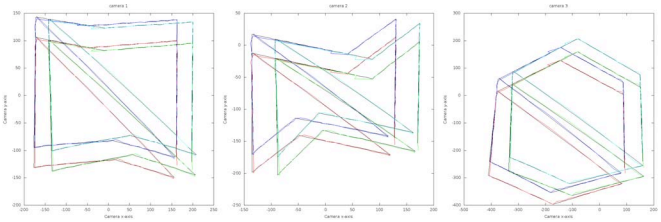

Fig. 6. Image-plane data from three cameras for $C_H = 3$ moving-target trial using KF with $Q = 5I$ and $R = I$

and S is the number of iterations in a given trial.

$$\bar{e}_{\text{check},T,S} = \frac{1}{T \times S} \sum_{l=1}^{T} \sum_{j=1}^{S} \left\| \boldsymbol{p}_{\text{check},l,j} - \boldsymbol{p}^{*}_{\text{check},j} \right\| \qquad (17)$$

where $\boldsymbol{p}_{\text{check},l,j} = [\boldsymbol{p}_{1,l,j} \quad \boldsymbol{p}_{2,l,j} \quad \boldsymbol{p}_{3,l,j} \quad \boldsymbol{p}_{4,l,j}]^{\mathsf{T}}$ comprises the $\boldsymbol{p}_{i,l,j}$ Cartesian position vectors to the the $i = 1 \ldots 4$ corners of the checkerboard at time step j of trial l. Note that in (17) $\boldsymbol{p}^{*}$ has no l subscript because the target path is identical in every trial.

Figure 7 plots $\bar{e}_{\text{check},T,S}$ for the different KF formulations of §II-B with varied covariance matrices as well as for GN. Contrary to the static-target results in Figure 4 here KF often performs worse than GN. The notable exception being when $Q = 5I$ and $R = I$, for which KF outperforms GN. This shows that the moving-target simulations are more sensitive to the KF noise covariance matrices. The maximum allowable joint offset norm μ of (13) is much higher for the moving-target trials, which might be why KF results are better in stationary-target simulations. The fact that KF tracking improves as Q grows with respect to R (see Figure 7) indicates that the process models are more accurate when the target is stationary than when it moves along the edges of a cube. The figures show that with proper noise covariance modeling KF outperforms GN for both static and moving targets.

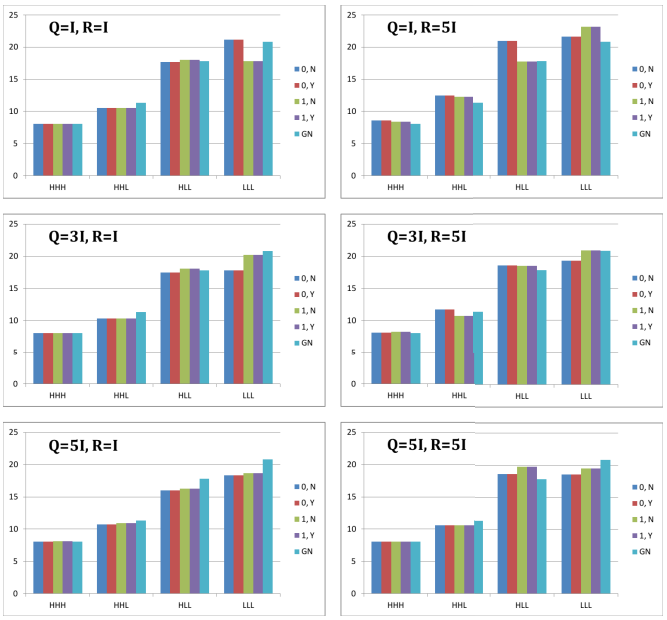

Fig. 7. Average checkerboard-corners error norm using combinations of high- and low-cost cameras with GN and 24 KF variations, "0" or "1" refers to the state space representation order and "Y" or "N" refers to with or without input

V. Conclusion and Future Work

A Kalman filter based visual servo control method is developed and analyzed. Numerous control laws exist in the calibrated VS literature. Few options are available for uncalibrated VS, with a Gauss-Newton based method being the de facto standard. Comparison is made between this conventional VS control law and the Kalman filter approach. With sufficiently high estimates of the covariance matrix Q the Kalman filter is shown to outperform GN in all camera and target scenarios tested. Four KF formulations are discussed and tested in simulation that vary in state space order and the use of input. Simulations show that including input in the KF equations yields little performance change. All formulations are shown to be completely controllable and observable given that servoing begins with the robot in a full column rank configuration and the robot passes through a full column rank configuration at least once during servoing.

This paper demonstrates that uncalibrated visual servoing can be improved by using the Kalman filter to estimate the robot joint positions at the desired image pose. The Kalman filter has previously been applied in position based (hence calibrated) visual servoing for target pose estimation and in uncalibrated VS for Jacobian estimation. In this work the Cartesian target pose is not calculated and Jacobian estimation is assumed; the recursive least squares technique of Piepmeier [15] is used in the simulations.

Future inquiries should obtain moving-target data via experiments and simulations using a smaller maximum allowable joint offset norm. Since the KF control law performance is shown to be dependent on noise covariance values additional gains could be obtained by better characterizing these terms or employing some adaptive filtering methods.

Acknowledgements: This work was funded as an Independent Research and Development project at Georgia Tech Research Institute, Food Processing Technology Division.

References

[1] Bonkovic, M., Hace, A., and Jezernik, K., Population-based uncalibrated visual servoing, Mechatronics, IEEE/ASME Transactions on, vol. 13, pp. 393 397, june 2008.

[2] Brogan, W. L., Modern Control Theory Third Edition. Prentice Hall, 1991.

[3] Chaumette, F. and Hutchinson, S., Visual servo control, part I: Basic approaches, IEEE Robotics and Automation Magazine, vol. 13, pp. 8290, Dec. 2006.

[4] Collewet, C., Marchand, E., and Chaumette, F., Visual servoing set free from image processing, in Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on, pp. 81 86, may 2008.

[5] Dame, A. and Marchand, E., Improving mutual information-based visual servoing, in Robotics and Automation (ICRA), 2010 IEEE International Conference on, pp. 5531 5536, may 2010.

[6] Deng, Z. and Jagersand, M., Evaluation of model independent image-based visual servoing, in Computer and Robot Vision, 2004. Proceedings. First Canadian Conference on, pp. 138 144, 17-19, 2004.

[7] Gao, Z. and Su, J., Switch images based on fusion in uncalibrated visual servoing, 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems (IEEE Cat. No. 06CH37780D), pp. 38038, 2006.

[8] Hao, M. and Sun, Z., A universal state-space approach to uncalibrated model-free visual servoing, Mechatronics, IEEE/ASME Transactions on, vol. PP, no. 99, pp. 114, 2011.

[9] Hosoda, K. and Asada, M., Versatile visual servoing without knowledge of true jacobian, Proceedings of the IEEE/RSJ/GI International Conference on Intelligent Robots and Systems, vol. 1, pp. 186193, 12-16 1994.

[10] Lippiello, V., Siciliano, B., and Villani, L., Eye-in-hand/eye-to-hand multi-camera visual servoing, in Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC 05. 44th IEEE Conference on, pp. 5354 5359, dec. 2005.

[11] Marshall, Multi-Camera Uncalibrated Visual Servoing. PhD thesis, Georgia Institute of Technology, 2013.

[12] Marshall, M., Matthews, M., Hu, A.-P., McMurray, G., and Lipkin, H., Uncalibrated visual servoing for intuitive human guidance of robots, in Robotics and Automation (ICRA), 2012 IEEE International Conference on, pp. 4463 4468, may 2012.

[13] Munnae, Uncalibrated Robotic Visual Servo Tracking For Large Residual Problems. PhD thesis, Georgia Institute of Technology, 2010.

[14] Pari, L., Sebastian, J., Traslosheros, A., and Angel, L., A comparative study between analytic and estimated image jacobian by using a stereoscopic system of cameras, in Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, pp. 6208 6215, oct. 2010.

[15] Piepmeier, J., McMurray, G., and Lipkin, H., Uncalibrated dynamic visual servoing, IEEE Transactions on Robotics and Automation, vol. 20, pp. 143147, feb. 2004.

[16] Piepmeier, J., Experimental results for uncalibrated eye-in-hand visual servoing, pp. 335 339, 2003.

[17] Qian, J. and Su, J., Online estimation of image jacobian matrix by kalman-bucy filter for uncalibrated stereo vision feedback, Proceedings of the 2002 IEEE International Conference on Robotics and Automation, vol. 1, pp. 562567, 2002.

[18] Sebastin, J., Pari, L., Angel, L., and Traslosheros, A., Uncalibrated visual servoing using the fundamental matrix, Robotics and Autonomous Systems, vol. 57, no. 1, pp. 110, 2009.

[19] Wang, H., Liu, Y.-H., and Zhou, D., Adaptive visual servoing using point and line features with an uncalibrated eye-in-hand camera, Robotics, IEEE Transactions on, vol. 24, pp. 843 857, aug. 2008.