ASME 2012 5th Annual Dynamic Systems and Control Conference joint with the JSME 2012 11th Motion and Vibration Conference
DSCC2012-MOVIC2012
October 17-19, 2012, Fort Lauderdale, Florida, USA

DSCC2012-MOVIC2012-8778

EXPERIMENTAL RESULTS FOR MOVING OBJECT STRUCTURE ESTIMATION USING AN UNKNOWN INPUT OBSERVER APPROACH

Sujin Jang, Ashwin P. Dani, Carl D. Crane III and Warren E. Dixon*
Department of Mechanical and Aerospace Engineering
University of Florida
Gainesville, Florida, 32611
Email: sweetoboe@ufl.edu;ashwin31@ufl.edu;ccrane@ufl.edu;wdixon@ufl.edu

ABSTRACT

An application and experimental verification of the online structure from motion (SFM) method is presented to estimate the position of a moving object using a moving camera. An unknown input observer is implemented for the position estimation of a moving object attached to a two-link robot observed by a moving camera attached to a PUMA robot. The velocity of the object is considered as an unknown input to the perspective dynamical system. Series of experiments are performed with different camera and object motions. The method is used to estimate the position of the static object as well as the moving object. The position estimates are compared with ground-truth data computed using forward kinematics of the PUMA and the two-link robot. The observer gain design problem is formulated as a convex optimization problem to obtain an optimal observer gain.

INTRODUCTION

Recovering the structure of a moving object using a moving camera has been well studied in literature the past decade [1–8]. In [6], a batch algorithm is developed by approximating the trajectories of a moving object using a linear combination of discrete cosine transform (DCT) basis vectors. Batch algorithms use an algebraic relationship between 3D coordinates of points in the camera coordinate frame and corresponding 2D projections on the image frame collected over n images to estimate the structure. Hence, batch algorithms are not useful in real-time control algorithms. For visual servo control or video-based surveillance tasks, online structure estimation algorithms are required.

Recently, a causal algorithm is presented in [7] to estimate the structure and motion of objects moving with constant linear velocities observed by a moving camera with known camera motions. A new method based on an unknown input observer (UIO) is developed in [8] to estimate the structure of an object moving with time-varying velocities using a moving camera with known velocities.

The contributions of this work is to experimentally verify the unknown input observer in [8] for structure estimation of a moving object. A series of experiments are conducted on a PUMA 560 and a two-link robot. A camera is attached to the PUMA and the target is attached to the moving two-link robot. The camera images are processed to track a feature point while camera velocities are measured using the joint encoders. To obtain the ground-truth data, the distance between the origin of the PUMA and origin of the two-link robot is measured and positions of the camera and moving object with respect to respective origins are obtained using the forward kinematics of the robots. The estimated position of object is compared with the ground-truth data. The experiments are conducted to estimate the structure of a static as well as a moving object keeping the same observer structure. The experiments prove the advantage of the observer in the sense that a-priori knowledge of object state (static or moving) is not required. A convex optimization problem is solved for computing an observer gain to reduce the effects of noise and disturbances.

PERSPECTIVE CAMERA MODEL

In this section, the kinematic relationship of the moving camera and the object, and geometric relationship of image formation is briefly described.

*Address all correspondence to this author.

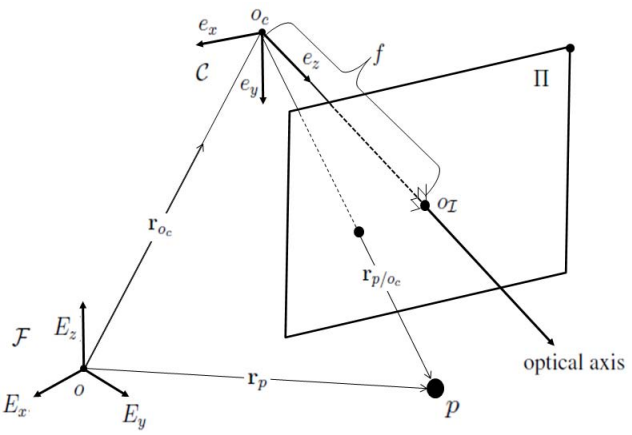

Figure 1. A PERSPECTIVE PROJECTION AND KINEMATIC CAMERA MODEL.

Kinematic Modeling

Considering a moving camera observing an object, define an inertially fixed reference frame, $\mathcal{F}:\{o;\,\mathbf{E}_x,\,\mathbf{E}_y,\,\mathbf{E}_z\}$, and a reference frame fixed to a camera, $\mathcal{C}:\{o_c;\,\mathbf{e}_x,\,\mathbf{e}_y,\,\mathbf{e}_z\}$ as shown in Fig. 1. The position of a point p relative to a point o is denoted by $\underline{\mathrm{r}}_p$. The position of o_c (origin of $\mathcal{C}$) relative to the point o (origin of $\mathcal{F}$) is denoted by $\underline{\mathrm{r}}_{o_c}$. In the following development every vector and tensor are expressed in terms of the basis $\{\mathbf{e}_x,\,\mathbf{e}_y,\,\mathbf{e}_z\}$ fixed in $\mathcal{C}$ [1]. The position of p measured relative to the point o_c is expressed as

$$\mathbf{r}_{p/o_c}=\mathbf{r}_p-\mathbf{r}_{o_c}=\begin{bmatrix}X(t)\;Y(t)\;Z(t)\end{bmatrix}^T \tag{1}$$

where $X(t)$, $Y(t)$ and $Z(t)\in\mathbb{R}$. The linear velocity of the object and the camera as viewed by an observer in the inertial reference frame are given by

$${}^{\mathcal{F}}\mathbf{V}_p=\overset{F}{\frac{d}{dt}}(\mathbf{r}_p)=\begin{bmatrix}v_{px}\;v_{py}\;v_{pz}\end{bmatrix}^T\in\mathcal{V}_p\subset\mathbb{R}^3, \tag{2}$$

$${}^{\mathcal{F}}\mathbf{V}_{o_c}=\overset{F}{\frac{d}{dt}}(\mathbf{r}_{\mathbf{o_c}})=\begin{bmatrix}v_{cx}\;v_{cy}\;v_{cz}\end{bmatrix}^T\in\mathcal{V}_c\subset\mathbb{R}^3. \tag{3}$$

Using Eqs. (1) - (3), the velocity of p as viewed by an observer in $\mathcal{C}$ is given by

$$\begin{aligned}{}^{\mathcal{C}}\mathbf{V}_{p/o_c}&=\overset{\mathcal{C}}{\frac{d}{dt}}(\mathbf{r}_p-\mathbf{r}_{\mathbf{o_c}})=\begin{bmatrix}\dot{X}(t)\;\dot{Y}(t)\;\dot{Z}(t)\end{bmatrix}^T,\\ {}^{\mathcal{C}}\mathbf{V}_{p/o_c}&={}^{\mathcal{C}}\mathbf{V}_p-{}^{\mathcal{C}}\mathbf{V}_{o_c}={}^{\mathcal{C}}\mathbf{V}_p-{}^{\mathcal{C}}\mathbf{V}_{o_c}+{}^{\mathcal{C}}\mathbf{w}^{\mathcal{F}}\times(\mathbf{r}_p-\mathbf{r}_{o_c})\end{aligned} \tag{4}$$

where ${}^{\mathcal{C}}\mathbf{w}^{\mathcal{F}}$ denotes the angular velocity of $\mathcal{F}$ as viewed by an observer in $\mathcal{C}$. The angular velocity of the camera relative to $\mathcal{F}$ is expressed as ${}^{\mathcal{F}}\mathbf{w}^{\mathcal{C}}=\begin{bmatrix}\omega_x\;\omega_y\;\omega_z\end{bmatrix}^T$. Since ${}^{\mathcal{C}}\mathbf{w}^{\mathcal{F}}$ and ${}^{\mathcal{F}}\mathbf{w}^{\mathcal{C}}$ are related as ${}^{\mathcal{C}}\mathbf{w}^{\mathcal{F}}=-{}^{\mathcal{F}}\mathbf{w}^{\mathcal{C}}$,

$${}^{\mathcal{C}}\mathbf{w}^{\mathcal{F}}=\begin{bmatrix}-\omega_x\;-\omega_y\;-\omega_z\end{bmatrix}^T. \tag{5}$$

Substituting Eqs. (1) - (3) into Eq. (4) yields

$$\begin{aligned}\begin{bmatrix}\dot{X}(t)\\ \dot{Y}(t)\\ \dot{Z}(t)\end{bmatrix}&=\begin{bmatrix}v_{px}-v_{cx}\\ v_{py}-v_{cy}\\ v_{pz}-v_{cz}\end{bmatrix}+\begin{bmatrix}0&\omega_z&-\omega_y\\ -\omega_z&0&\omega_x\\ \omega_y&-\omega_x&0\end{bmatrix}\begin{bmatrix}X(t)\\ Y(t)\\ Z(t)\end{bmatrix},\\ &=\begin{bmatrix}v_{px}-v_{cx}+\omega_zY(t)-\omega_yZ(t)\\ v_{py}-v_{cy}+\omega_xZ(t)-\omega_zX(t)\\ v_{pz}-v_{cz}+\omega_yX(t)-\omega_xY(t)\end{bmatrix}\end{aligned} \tag{6}$$

The inhomogeneous coordinates of Eq. (1), $\bar{m}(t)=\begin{bmatrix}\bar{m}_1(t)\;\bar{m}_2(t)\;1\end{bmatrix}^T\in\mathbb{R}^3$, are defined as

$$\bar{m}(t)\triangleq\begin{bmatrix}\frac{X(t)}{Z(t)}\;\frac{Y(t)}{Z(t)}\;1\end{bmatrix}^T. \tag{7}$$

Considering subsequent development, the state vector $x(t)=\begin{bmatrix}x_1(t)\;x_2(t)\;x_3(t)\end{bmatrix}^T\in\mathcal{Y}\subset\mathbb{R}^3$ is defined as

$$x(t)\triangleq\begin{bmatrix}\frac{X}{Z}\;\frac{Y}{Z}\;\frac{1}{Z}\end{bmatrix}^T. \tag{8}$$

Using Eqs. (6) and (8), the dynamics of the state $x(t)$ can be expressed as

$$\begin{aligned}\dot{x}_1&=\Omega_1+f_1+v_{px}x_3-x_1v_{pz}x_3,\\ \dot{x}_2&=\Omega_2+f_2+v_{py}x_3-x_2v_{pz}x_3,\\ \dot{x}_3&=v_{cz}x_3^2+(x_2\omega_x-x_1\omega_2)x_3-v_{pz}x_3^2,\\ y&=\begin{bmatrix}x_1\;x_2\end{bmatrix}^T.\end{aligned} \tag{9}$$

where $\Omega_1(u,y),\Omega_2(u,y),f_1(u,x),f_2(u,x),f_3(u,x)\in\mathbb{R}$ are defined as

$$\begin{aligned}\Omega_1(u,y)&\triangleq x_1x_2\omega_x-\omega_y-x_1^2\omega_y+x_2\omega_z,\\ \Omega_2(u,y)&\triangleq\omega_x+x_2^2\omega_x-x_1x_2\omega_y-x_1\omega_z,\\ f_1(u,x)&\triangleq(x_1v_{cz}-v_{cx})x_3,\\ f_2(u,x)&\triangleq(x_2v_{cz}-v_{cy})x_3,\\ f_3(u,x)&\triangleq v_{cz}x_3^2+(x_2\omega_x-x_1\omega_y)x_3.\end{aligned}$$

Assumption 1: The velocities of the camera and object are assumed to be upper and lower bounded by constants.
Assumption 2: Since the states $x_1(t)$ and $x_2(t)$ are equivalent to

[1] In the entire development, it is assumed that $\{\cdot\}_e$ is omitted in vector representation where $\{\cdot\}_e$ denotes the column-vector representations of the vector in the basis $\{\mathbf{e}_x,\,\mathbf{e}_y,\,\mathbf{e}_z\}$ (i.e. ${}^{\mathcal{F}}\mathbf{V}_p=\{{}^{\mathcal{F}}\mathbf{V}_p\}_e$).

the pixel coordinates in the image plane, and the size of image plane is bounded by known constants, thus it can be assumed that $x_1(t)$ and $x_2(t)$ are also bounded by $\underline{W} \leq x_1(t) \leq \overline{W}$, $\underline{H} \leq x_2(t) \leq \overline{H}$ where $\underline{W}, \overline{W}$ and $\underline{H}, \overline{H}$ are obtained using the width and height of image plane.

Camera Model and Geometric Image Formation

In order to describe the image formation process, the geometric perspective projection is commonly used as depicted in Fig. 1. The projection model consists of an image plane Π, a center of projection o_c, a center of image plane o_I and the distance between Π and o_c (focal length). Two-dimensional pixel coordinates of the projected point q in the image plane Π is given by $\tilde{m}(t) = \begin{bmatrix} u(t) & v(t) & 1 \end{bmatrix}^T \in I \subset \mathbb{R}^3$. The three-dimensional coordinates $\bar{m}(t)$ are related to the pixel coordinates $\tilde{m}(t)$ by the following relationship [9]

$$\tilde{m}(t) = K_c \bar{m}(t) \tag{10}$$

where $K_c \in \mathbb{R}^{3\times3}$ is an invertible upper-triangular form of intrinsic camera matrix given by

$$K_c = \begin{bmatrix} f & 0 & c_x \\ 0 & \alpha f & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

where α is the image aspect ratio, f is the focal length and (c_x, c_y) denotes the optical center o_I expressed in pixel coordinates. To simplify the derivation of the perspective projection matrix, the projection center is assumed to coincide with the origin of the camera reference frame *(Assumption 3)* and the optical axis is aligned with the z-axis of the coordinate system fixed in the camera (*Assumption 4*).

DESIGN OF AN UNKNOWN INPUT OBSERVER

An unknown input observer for a class of nonlinear system is designed to estimate the position of tracked object in [8]. Based on Eqs. (6) and (8), the following nonlinear system can be constructed

$$\begin{aligned} \dot{x} &= f(x,u) + g(y,u) + Dd \\ y &= Cx \end{aligned} \tag{11}$$

where $x(t) \in \mathbb{R}^3$ is a state of the system, $y(t) \in \mathbb{R}^2$ is an output of the system, $u(t) \in \mathbb{R}^6$ is a measurable input, $d(t) \in \mathbb{R}$ is an unmeasurable input, $g(y,u) = \begin{bmatrix} \Omega_1 & \Omega_2 & 0 \end{bmatrix}^T$ is nonlinear in $y(t)$ and $u(t)$, and $f(x,u) = \begin{bmatrix} f_1 & f_2 & f_3 \end{bmatrix}^T$ is nonlinear in $x(t)$ and $u(t)$ satisfying the Lipschitz condition $\|f(x,u) - f(\hat{x},u)\| \leq \gamma_1 \|x - \hat{x}\|$ where $\gamma_1 \in \mathbb{R}^+$. A full row rank of matrix $C \in \mathbb{R}^{2\times3}$ is selected as

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix},$$

and a full column rank of $D \in \mathbb{R}^{3\times1}$ is chosen as $D = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T$ or $\begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T$. The system in Eq. (11) can be written in the following form :

$$\begin{aligned} \dot{x} &= Ax + \bar{f}(x,u) + g(y,u) + Dd \\ y &= Cx \end{aligned} \tag{12}$$

where $\bar{f}(x,u) = f(x,u) - Ax$ and $A \in \mathbb{R}^{3\times3}$. The function $\bar{f}(x,u)$ satisfies the Lipschitz condition [10, 11]

$$\|f(x,u) - f(\hat{x},u) - A(x - \hat{x})\| \leq (\gamma_1 + \gamma_2) \|x - \hat{x}\| \tag{13}$$

where $\gamma_2 \in \mathbb{R}^+$. A nonlinear unknown input observer for Eq. (12) is designed as

$$\begin{aligned} \dot{z} &= Nz + Ly + M\bar{f}(\hat{x},u) + Mg(y,u) \\ \hat{x} &= z - Ey \end{aligned} \tag{14}$$

where $\hat{x}(t) \in \mathbb{R}^3$ is an estimate of the unknown state $x(t)$, $z(t) \in \mathbb{R}^3$ is an auxiliary signal, the matrices $N \in \mathbb{R}^{3\times3}$, $L \in \mathbb{R}^{3\times2}$, $E \in \mathbb{R}^{3\times2}$, $M \in \mathbb{R}^{3\times3}$ are designed as [12]

$$\begin{aligned} M &= I_3 + EC \\ N &= MA - KC \\ L &= K(I_2 + CE) - MAE \\ E &= -D(CD)^+ + Y\left(I_2 - (CD)(CD)^\dagger\right) \end{aligned} \tag{15}$$

where $(CD)^\dagger$ denotes the generalized pseudo inverse of the matrix CD. The gain matrix $K \in \mathbb{R}^{3\times2}$ and matrix $Y \in \mathbb{R}^{3\times2}$ are selected such that

$$Q \triangleq N^T P + PN + \left(\gamma_1^2 + \gamma_2^2\right) PMM^T P + 2I_3 < 0 \tag{16}$$

where $P \in \mathbb{R}^{3\times3}$ is a positive definite, symmetric matrix. The stability result of the observer is stated using the following theorem.
Theorem: The nonlinear unknown input observer in Eq. (14) is exponentially stable if Eq. (16) is satisfied.

Proof: See theorem 1 of [8].

To find E, K and P, Eq. (16) is reformulated in terms of a linear matrix inequality (LMI) as [8]

$$\begin{bmatrix} X_{11} & \beta X_{12} \\ \beta X_{12}^T & -I_3 \end{bmatrix} < 0 \tag{17}$$

where

$$
\begin{aligned}
X_{11} &= A^T (I_3 + FC)^T P + P(I_3 + FC)A + A^T C^T G^T P_Y^T \\
&\quad + P_Y GCA - C^T P_K^T - P_K C + 2I_3 \\
X_{12} &= P + PFC + PFC + P_Y GC \\
P_Y &= PY \\
P_K &= PK \\
\beta &= \sqrt{\gamma_1^2 + \gamma_2^2}.
\end{aligned}
$$

To solve LMI feasibility problem in Eq. (17), the CVX toolbox in MATLAB is used [13]. Using P, P_K and P_Y obtained from Eq. (17), K and Y are computed using $K = P^{-1}P_K$ and $Y = P^{-1}P_Y$.

If the number of outputs n_y is equal to the number of unknown inputs n_d, then the unknown disturbance can be represented as $Dd(t) = D_1 d_1(t) + D_2 d_2(t)$ where $d_1(t)$ includes $(n_d - 1)$ number of unknown inputs, $d_2(t)$ includes remaining unknown input and $D_1, D_2 \in \mathbb{R}^{3\times 1}$ are full column rank. In this case, the estimation error will be uniformly ultimately bounded with the bound proportional to the norm of the disturbance $d_2(t)$. The gain K and Y can be selected optimally to minimize the effects of sensor noise and the external disturbance by solving the following LMI.

$$
\begin{aligned}
&\min \quad \delta\gamma^2 - (1-\delta)\lambda_{min}(P), \\
&\text{subject to} \quad P > 0, \quad \gamma \geq 0, \\
&\begin{bmatrix} X_{11} & \beta X_{12} & X_{13} \\ \beta X_{12}^T & -I_3 & 0 \\ X_{13}^T & 0 & -\gamma^2 \end{bmatrix} < 0
\end{aligned} \tag{18}
$$

where $X_{13} = PD_2$, $\delta \in [0,1]$, λ_{min} denotes the minimum eigenvalue of a matrix, and γ denotes the $\mathcal{L}_2$ norm of the estimation error with respect to the disturbance $d_2(t)$.

EXPERIMENTS AND RESULTS

To verify the designed unknown input observer [8] for real-time implementation, two sets of experiments are conducted on a PUMA 560 serial manipulator and a two-link planar robot. The first set is performed for the relative position estimation of a static object using a moving camera. The second set is performed for position estimation of moving object. A schematic overview of experimental configuration is illustrated in Fig. 2.

Testbed Setup

The testbed consists of five components: (1) robot manipulators, (2) camera, (3) image processing workstation (main), (4) robot control workstation (PUMA and two-link), and (5) serial communication. Figure 3 shows the experimental platforms. A camera is rigidly fixed to the end-effector of the PUMA 560. The

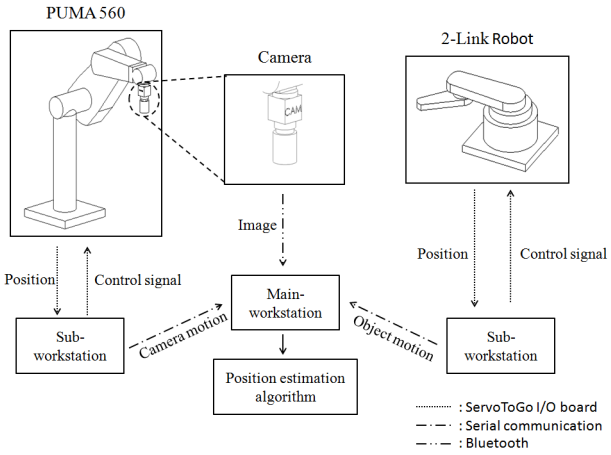

Figure 2. AN OVERVIEW OF EXPERIMENTAL CONFIGURATION.

Figure 3. PLATFORMS.

PUMA and the two-link robot are rigidly attached to a work table. Experiments are conducted to estimate the position of the static as well as the moving object. A fiduciary marker is used as an object in all the experiments. For experiments involving a static object, the object is fixed to the work table. For experiments involving a moving object, the object is fixed to the end-effector of the two-link robot which follows a desired trajectory. The PUMA 560 is used to move the camera while observing the static or moving object.

A mvBlueFox-120a color USB camera is used to capture images. The camera is calibrated using the MATLAB camera calibration toolbox [14]. A Core2-Duo 2.53 GHz laptop (main-workstation) operating under Windows 7 is used to carry out the image processing and to store data transmitted from the PUMA-workstation. The image processing algorithms are written in C/C++, and developed in Microsoft Visual Studio 2008. The OpenCV and MATRIX-VISION API libraries are used to capture the images and to implement a KLT feature point tracker [15, 16]. Sub-workstations (PUMA and two-link) are composed of two Pentium 2.8 GHz PCs operating under QNX. These two

computers are used to host control algorithms for the PUMA 560 and the two-link robot via Qmotor 3.0 [17]. A PID controller is employed to control the six joints of the PUMA 560. A RISE-based controller [18] is applied to control the two-link robot. Control implementation and data acquisition for the two robots are operated at 1.0 kHz frequency using the ServoToGo I/O board. The forward velocity kinematics [19, 20] are used to obtain the position and velocity of the camera and tracked point. The camera velocities computed on the PUMA-workstation are transmitted to the main-workstation via serial communication at 30 Hz. The pose (position and orientation) of the tracked point and the camera are computed and stored in the sub-workstations at 1.0 KHz. The position of the camera and the point are used to compute the ground truth distance between the camera and object as $\{r_{obj/cam}\}_e = \left[\{R\}_e^E\right]^{-1} \{r_{obj} - r_{cam}\}_E$ where $\{R\}_e^E \in \mathbb{R}^{3\times 3}$ is the rotation matrix of the camera with respect to the inertial reference frame.

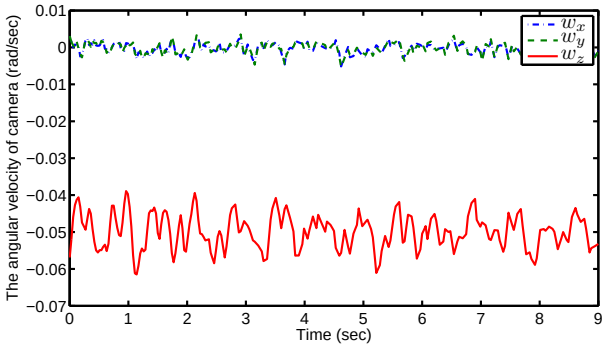

Figure 4. CAMERA ANGULAR VELOCITY.

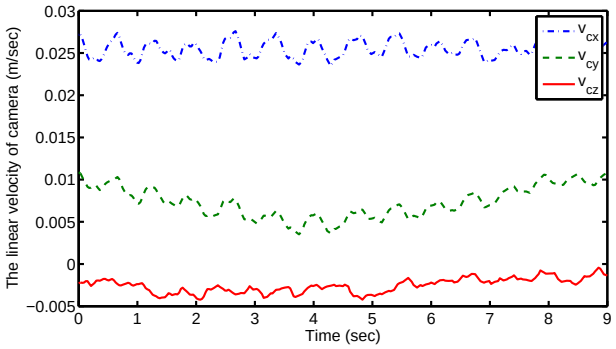

Figure 5. CAMERA LINEAR VELOCITY.

Experiment I : Moving camera with a static point

In this section, the structure estimation algorithm is implemented for a static object observed using a moving camera. A tracked point on the static object is observed by a downward-looking camera. Since v_{px}, v_{py} and v_{pz} are zero for a static object, the unmeasurable disturbance input $d(t)$ is zero. An experiment set is designed to test the observer with time-varying velocities of camera in the Y and Z direction. Figures 4 and 5 show the linear and angular camera velocities. The matrices A, C and D are selected to be

$$A = \begin{bmatrix} 0.00 & -0.05 & -1.0 \\ 0.05 & 0.00 & 0.0 \\ 0.00 & 0.00 & 0.00 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, D = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}.$$

The matrix Y and gain matrix K are computed using the CVX toolbox in MATLAB [13] as

$$K = \begin{bmatrix} 1.3149 & 0.0000 \\ 0.0000 & 1.3149 \\ 0.0590 & -0.1256 \end{bmatrix}, Y = \begin{bmatrix} -1.0000 & 0.0000 \\ 0.0000 & 0.0000 \\ 2.5125 & 0.0000 \end{bmatrix}.$$

The estimation result is illustrated in Figs. 6 and 7. The steady-state RMS errors in the position estimation in X, Y and Z coordinates are 0.0029 m, 0.0210 m and 0.0412 m, respectively.

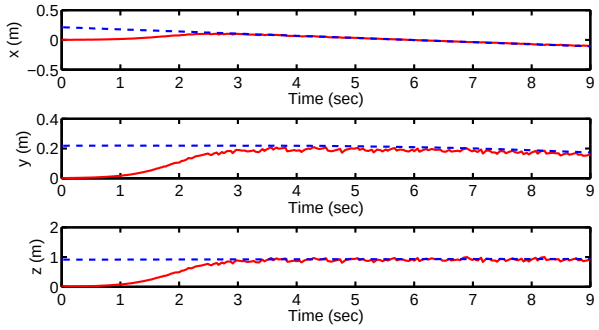

Figure 6. COMPARISON OF THE ACTUAL (DASH) AND ESTIMATED (SOLID) POSITION OF A STATIC POINT WITH RESPECT TO A MOVING CAMERA.

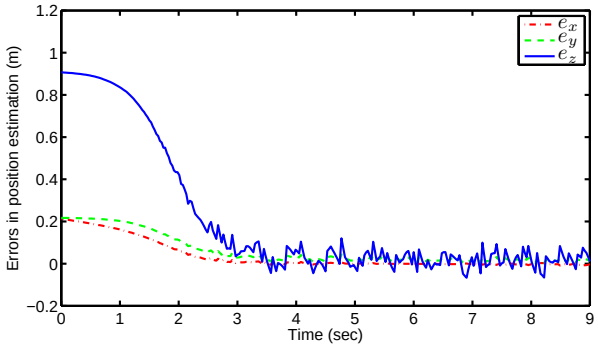

Figure 7. POSITION ESTIMATION ERROR FOR A STATIC POINT.

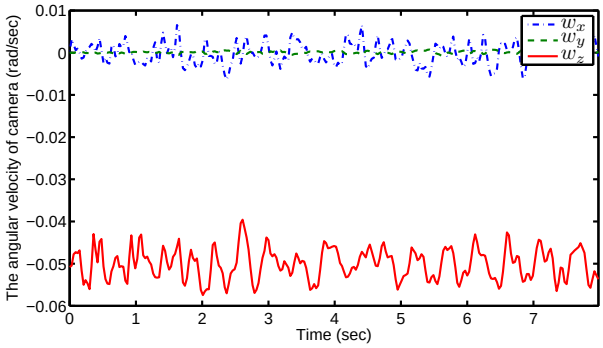

Figure 8. CAMERA ANGULAR VELOCITY.

Experiment II : Moving camera with moving object

In this section, the observer is used to estimate the position of a moving object using a moving camera. A downward-looking camera observes a moving point fixed to the moving two-link robot arm. In this case, the object is moving in the $X-Y$ plane with unknown velocities $v_{px}(t)$ and $v_{py}(t)$. The estimation result using the observer gain obtained from the method in [8] and from the convex optimization method are compared. In the experiment Set 2, the linear velocity of camera has two time-varying velocities to test the observer with more generalized trajectory of the moving camera.

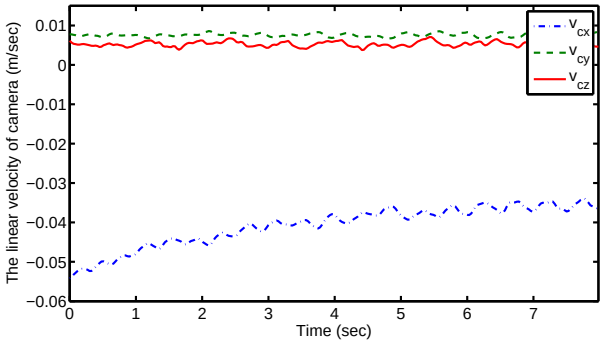

Figure 9. CAMERA LINEAR VELOCITY.

Set 1 In this experiment set, the observer is tested with a time-varying linear velocity of camera along the X direction. The camera velocities are shown in Figs. 8 and 9. The matrices A, C and D are given by

$$A = \begin{bmatrix} 0.00 & -0.05 & 0.00 \\ 0.05 & 0.00 & -0.30 \\ 0.00 & 0.00 & 0.00 \end{bmatrix}, \; C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \; D = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}.$$

The matrix Y and gain matrix K are computed using the CVX toolbox in MATLAB and are given as

$$K = \begin{bmatrix} 1.2204 & 0.0000 \\ 0.0000 & 1.2204 \\ 0.3731 & 0.0000 \end{bmatrix}, \; Y = \begin{bmatrix} 0.0000 & 0.0000 \\ 0.0000 & -1.0000 \\ 0.0000 & 7.4618 \end{bmatrix}.$$

The estimation result is illustrated in Figs. 10 and 11. The RMS errors and peak errors of the estimated position in the steady-state are given in Tab. 1.

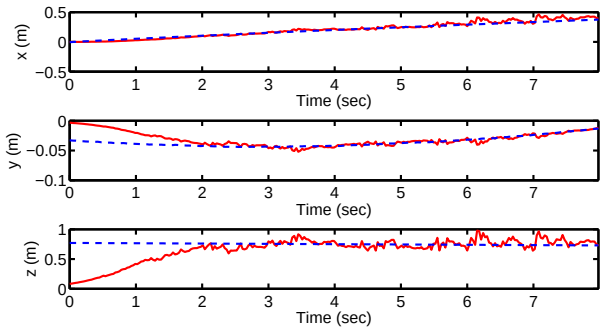

Figure 10. COMPARISON OF THE ACTUAL (DASH) AND ESTIMATED (SOLID) POSITION OF A MOVING POINT WITH RESPECT TO A MOVING CAMERA.

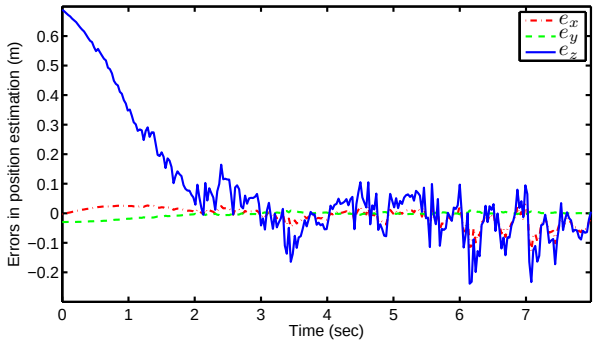

Figure 11. POSITION ESTIMATION ERROR FOR A MOVING POINT.

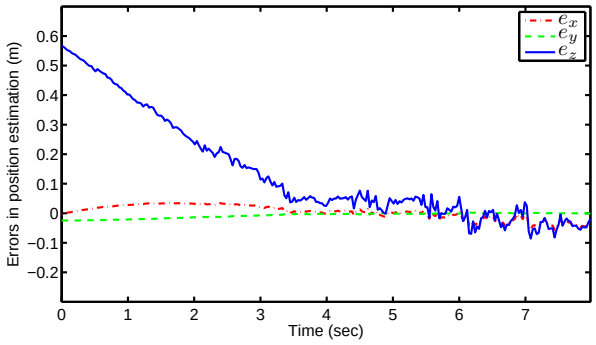

Figure 13. POSITION ESTIMATION ERROR FOR A MOVING POINT.

Set 1 with convex optimization approach In this experiment, a comparison between the gain computation using LMIs in Eqs. (17) and (18) is given. The matrix Y and gain matrix K for convex optimization are computed using the CVX toolbox as

$$K = \begin{bmatrix} 0.0003 & 0.0000 \\ 0.0000 & 0.3886 \\ 0.0000 & 0.0000 \end{bmatrix}, \; Y = \begin{bmatrix} 0.0000 & 0.0000 \\ 0.0000 & -1.0000 \\ 0.0000 & 3.0000 \end{bmatrix}.$$

The estimation result is illustrated in Figs. 13 and 12. Table 1 shows the steady-state RMS error and peak error of the position estimates using the convex optimization approach.

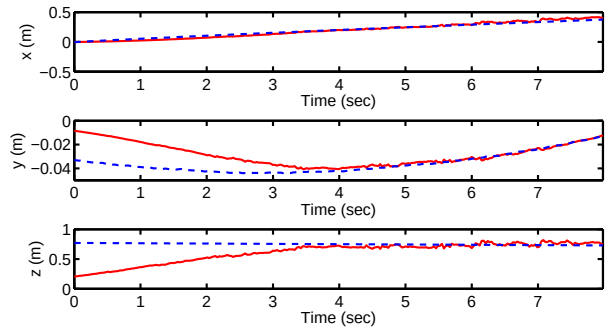

Figure 12. COMPARISON OF THE ACTUAL (DASH) AND ESTIMATED (SOLID) POSITION OF A MOVING POINT WITH RESPECT TO A MOVING CAMERA.

Table 1. POSITION ESTIMATION ERRORS.

	w/o optimization	w/ optimization
RMS in x (m)	0.0345	0.0227
RMS in y (m)	0.0031	0.0016
RMS in z (m)	0.0740	0.0416
peak error in x (m)	0.1273	0.0566
peak error in y (m)	0.0093	0.0036
peak error in z (m)	0.2385	0.0857

Set 2 In this experiment set, the linear camera velocities along the X and Y direction are time-varying, and the camera angular camera velocity is constant. The camera velocities are depicted in Figs. 14 and 15. The matrices A, C and D are given by

$$A = \begin{bmatrix} 0.00 & -0.05 & -1.00 \\ 0.05 & 0.00 & -0.30 \\ 0.00 & 0.00 & 0.00 \end{bmatrix}, \; C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \; D = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}.$$

The matrix Y and gain matrix K are computed as

$$K = \begin{bmatrix} 1.2892 & 0.0000 \\ 0.0000 & 1.2892 \\ 0.2313 & 0.0000 \end{bmatrix}, \; Y = \begin{bmatrix} 0.0000 & 0.0000 \\ 0.0000 & -1.0000 \\ 0.0000 & 4.6261 \end{bmatrix}.$$

The estimation result is depicted in Figs. 16 and 17. The steady-state RMS errors and peak errors of the position estimation in X, Y and Z coordinates are given in Tab. 2.

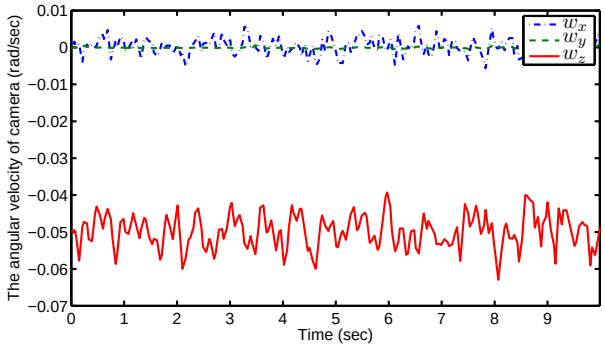

Figure 14. CAMERA ANGULAR VELOCITY.

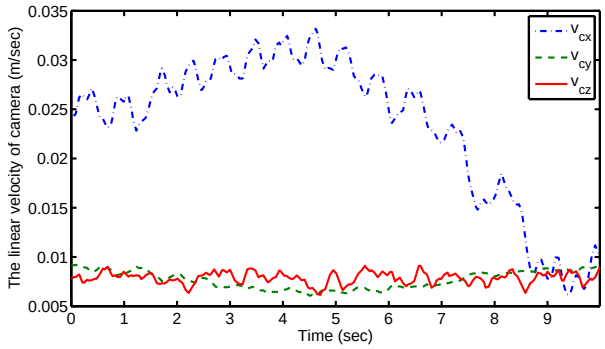

Figure 15. CAMERA LINEAR VELOCITY.

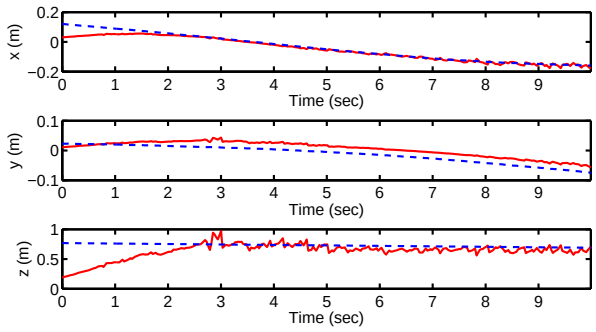

Figure 16. COMPARISON OF THE ACTUAL (DASH) AND ESTIMATED (SOLID) POSITION OF A MOVING POINT WITH RESPECT TO A MOVING CAMERA.

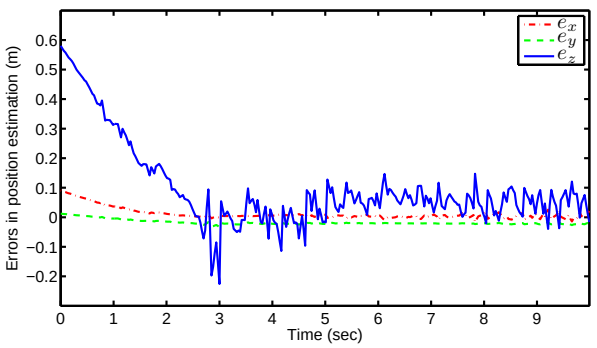

Figure 17. POSITION ESTIMATION ERROR FOR A MOVING POINT.

Set 2 with convex optimization approach In this experiment, a comparison between the gain computation using LMIs in Eqs. (17) and (18) is given. The matrix Y and gain matrix K for convex optimization are computed using the CVX toolbox as

$$K = \begin{bmatrix} 0.0003 & 0.0000 \\ 0.0000 & 0.3886 \\ 0.0000 & 0.0000 \end{bmatrix}, \; Y = \begin{bmatrix} 0.0000 & 0.0000 \\ 0.0000 & -1.0000 \\ 0.0000 & 3.0000 \end{bmatrix}.$$

The estimation result is illustrated in Figs. 19 and 18. Table 2 shows the RMS error and peak error of the position estimates using the convex optimization approach.

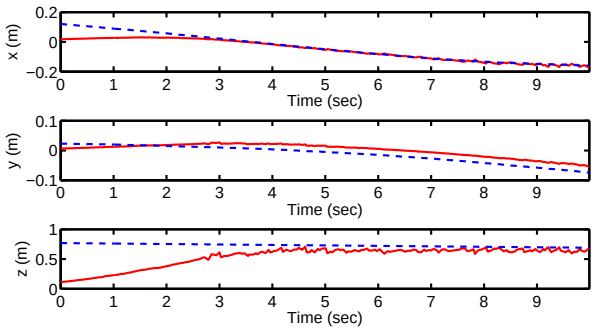

Figure 18. COMPARISON OF THE ACTUAL (DASH) AND ESTIMATED (SOLID) POSITION OF A MOVING POINT WITH RESPECT TO A MOVING CAMERA.

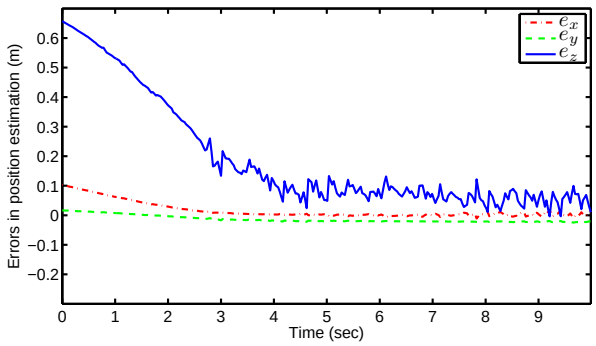

Figure 19. POSITION ESTIMATION ERROR FOR A MOVING POINT.

Table 2. POSITION ESTIMATION ERRORS.

	w/o optimization	w/ optimization
RMS in x (m)	0.0067	0.0044
RMS in y (m)	0.0216	0.0208
RMS in z (m)	0.0645	0.0730
peak error in x (m)	0.0258	0.0172
peak error in y (m)	0.0334	0.0249
peak error in z (m)	0.2257	0.1330

CONCLUSION

In this work, the unknown input observer in [8] is implemented to estimate the position of a static and a moving object. For the static object case, the number of unknown inputs are less than the number of measured outputs. Thus, the observer exponentially converges to the true state. For the moving object case, when the number of disturbance inputs is equal to the number of outputs, the observer yields an uniformly ultimately bounded result (cf., Set 2 of Experiment 2). Yet, the observer yields RMS error within 0.075 m precision in the presence of sensor noise in feature tracking and camera velocities. The observer yields good performance for detecting the coordinates for the static as well as moving objects as seen from the RMS errors in each experiment. As illustrated in Tabs. 1 and 2, the convex optimization approach reduces the effect of noise in the state estimation while the transient time increases. The convex optimization method can be used to obtain an optimal observer gain for more general camera and object motions.

ACKNOWLEDGMENT

This research is supported in part by NSF award numbers 0547448, 0901491, 1161260, and a contract with the Air Force Research Laboratory, Munitions Directorate at Eglin AFB. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the sponsoring agency.

References

[1] Avidan, S., and Shashua, A., 2000. "Trajectory triangulation: 3D reconstruction of moving points from a monocular image sequence". *IEEE Trans. Pattern Anal. Mach. Intell.,* **22**(4), Apr, pp. 348–357.

[2] Kaminski, J., and Teicher, M., 2004. "A general framework for trajectory triangulation". *J. Math. Imag. Vis.,* **21**(1), pp. 27–41.

[3] Han, M., and Kanade, T., 2004. "Reconstruction of a scene with multiple linearly moving objects". *Int. J. Comput. Vision,* **59**(3), pp. 285–300.

[4] Vidal, R., Ma, Y., Soatto, S., and Sastry, S., 2006. "Two-view multibody structure from motion". *Int. J. Comput. Vision,* **68**(1), pp. 7–25.

[5] Yuan, C., and Medioni, G., 2006. "3D reconstruction of background and objects moving on ground plane viewed from a moving camera". In Comput. Vision Pattern Recognit., Vol. 2, pp. 2261 – 2268.

[6] Park, H., Shiratori, T., Matthews, I., and Sheikh, Y., 2010. "3D reconstruction of a moving point from a series of 2D projections". In Euro. Conf. on Comp. Vision, Vol. 6313, pp. 158–171.

[7] Dani, A., Kan, Z., Fischer, N. R., and Dixon, W. E., 2010. "Structure and motion estimation of a moving object using a moving camera". In Proc. Am. Control Conf., pp. 6962–6967.

[8] Dani, A. P., Kan, Z., Fischer, N. R., and Dixon, W. E., 2011. "Structure estimation of a moving object using a moving camera: An unknown input observer approach". *IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, pp. 5005–5010.

[9] Szeliski, R., 2010. *Computer Vision: Algorithms and Applications*. Springer.

[10] Yaz, E., and Azemi, A., 1993. "Observer design for discrete and continuous nonlinear stochastic systems". *Int. J. Syst. Sci.,* **24**(12), pp. 2289–2302.

[11] Xie, L., and Khargonekar, P. P., 2010. "Lyapunov-based adaptive state estimation for a class of nonlinear stochastic systems". In Proc. of American Controls Conf., pp. 6071–6076.

[12] Darouach, M., Zasadzinski, M., and Xu, S., 1994. "Full-order observers for linear systems with unknown inputs". *IEEE Trans. Autom. Control,* **39**(3), Mar., pp. 606–609.

[13] Grant, M., and Boyd, S., 2005. Cvx: Matlab software for

disciplined convex programming. On the WWW. URL http://cvxr.com/cvx/.

[14] Bouguet, J., 2010. Camera calibration toolbox for matlab. On the WWW. URL http://www.vision.caltech.edu/bouguetj/.

[15] Tomasi, C., and Kanade, T., 1991. Detection and tracking of point features. Tech. rep., Carnegie Mellon University.

[16] Shi, J., and Tomasi, C., 1994. "Good features to track". In Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pp. 593–600.

[17] Loffler, M., Costescu, N., and Dawson, D., 2002. "Qmotor 3.0 and the qmotor robotic toolkit - an advanced PC-based real-time control platform". *IEEE Contr. Syst. Mag.,* **22**(3), pp. 12–26.

[18] Patre, P. M., Dupree, K., MacKunis, W., and Dixon, W. E., 2008. "A new class of modular adaptive controllers, part II: Neural network extension for non-LP systems". In Proc. Am. Control Conf., pp. 1214–1219.

[19] Spong, M., and Vidyasagar, M., 1989. *Robot Dynamics and Control*. John Wiley & Sons Inc., New York.

[20] Crane, C. D., and Duffy, J., 1998. *Kinematic Analysis of Robot Manipulators*. Cambridge.