Dynamic Stereo Triangulation for Robot Juggling

A. A. Rizzi and D. E. Koditschek
University of Michigan, Artificial Intelligence Laboratory
Department of Electrical Engineering and Computer Science

Abstract
We have devised a nonlinear state estimator to recover the position and velocity of falling balls directly from the image plane measurements of a stereo camera pair. Avoiding an explicit triangulation step in the estimation procedure allows the continued use of one camera's data even when the other camera may be occluded. This paper presents a rudimentary analysis, some simulation results, and data from a working implementation that suggest the potential utility of the idea.

1. Introduction
Figure 1 depicts the system architecture underlying a juggling robot that we have discussed at length in previous publications [11, 9, 6]. This paper concerns a new algorithm we have introduced into our signal processing module, the block labeled “Linear Observer” in the figure, which filters the data issued by the stereo camera pair we use as “eyes.” Previously, this module incorporated a linear state estimator for Newtonian free flight in order to obtain reliable information concerning the position and velocity of the bodies to be juggled. For its input, this linear observer requires measurements in cartesian coordinates of the bodies’ positions. Thus, the “Vision Coordinator” block in our system architecture has included an algebraic triangulation step. In the present paper we discuss recent experiments with a new signal processing method that eliminates the algebraic triangulation procedure. Camera output is directly filtered by a nonlinear state estimator. Our empirical experience with this new scheme is sufficiently promising to motivate our further efforts — presently in progress — to prove its stability.

The immediate motivation in our lab for this revision in the signal processing architecture has a very practical origin. It is almost guaranteed that a camera will “lose a ball” during each flight phase of a juggle: balls temporarily travel out of one or the other the field of view; one ball temporarily occludes the view of the second. Such losses are generally recoverable, and we have discussed in previous publications certain “control of attention” mechanisms that significantly aid in the recovery [8, 7]. However, as matters stood until recently, once one camera had lost a ball, the valid data from the other was ignored, as there was no new stereo image pair input to triangulate. Thus there was no resulting cartesian estimation error input to the linear observer (which consequently ran as a pure predictor until the camera system had recovered again). Despite its frequent disregard of valid data, this system managed to work reasonably well. It juggled one and two balls for hours at a time [6]. But it was clear to us that a more rational data management scheme could be found. Why throw out both cameras’ outputs when only one is blinded?

As an alternative approach, consider a state estimation scheme that can work directly from the cameras’ image plane measurements. Now, the signal processing algorithm can continue to work with the information from one camera while substituting predicted...
data in place of the other camera's information\textsuperscript{1}. But the transformation from cartesian space to image plane pairs is nonlinear and this entails a nonlinear state estimator. Happily, because of the special algebraic structure of the projective transformation, we have been able to devise a working nonlinear estimator to do the job. Moreover, there seems good reason to hope that a formal proof of its efficacy can be provided.

These ideas extend naturally to situations wherein sporadic observations of a dynamical object from n-cameras must be integrated to form an evolving view of its cartesian motion\textsuperscript{2}. More generally, we presume that many instances of the sensor fusion problem — the problem of deriving from multiple sources of overlapping but independently distorted data more accurate information than may obtain from any one alone — may be recast as nonlinear observer problems. Thus, one might refer to this approach as on-line dynamical data-fusion. Here, we explore a particularly simple instance.

2. Juggling Apparatus

Our juggling system, pictured in Figure 1, consists of three major components: an environment (the ball); the robot; and an environmental sensor (the vision system). We now sketch the operation of this system sufficiently to place the current problem in a meaningful context.

2.1. The Need for Continuous Flight Information

Following Bühler et al. [9], we command a robot to "juggle" by forcing it to track a reference trajectory generated by a distorted reflection of the ball's continuous trajectory. This policy amounts to the choice of a map $m$ from the phase space of the body to the joint space of the robot. A robot reference trajectory,

$$r(t) = m(b(t), b'(t)),$$  

(1)

is generated by the geometry of the graph of $m$ and the dynamics of ball, $b(t)$. This reference trajectory (along with the induced velocity and acceleration signals) can then be directly passed to a robot joint controller.\textsuperscript{3}

In following the prescribed joint space trajectory, the robot's paddle pursues a trajectory periodically intersecting that of the ball. The impacts induced at these intersections result in the desired juggling behavior.

Central to this juggling strategy is a sensory system capable of "keeping its eyes on the ball." We require that the vision system produce a 1 KHz signal containing estimates of the ball's spatial position and velocity (six measurements). Denote this "robot reference rate" by the symbol $r_f = 10^{-3}$sec. Two R5-170 CCD television cameras constitute the "eyes" of the juggling system and deliver a frame consisting of a pair of interlaced fields at 60 Hz, so that a new field of data is available every $r_f = 16.67 	imes 10^{-3}$sec. The CYCLOPS vision system, described in [9, 4], provides the hardware platform upon which the data in these fields are used to form the input signal to the mirror law, (1). The remainder of this section describes how this is done.

2.2. Triangulation and Flight Models

We work with the simple projective stereo camera model,

$$p : IR^3 \rightarrow IR^5,$$

that maps positions in affine 3-space to a pair of image plane projections in the standard manner. Knowledge of the cameras' relative positions and orientations together with knowledge of each camera's lens characteristics (at present we model only the focal length, and image center) permits the selection of a pseudo-inverse or "triangulation function",

$$p^1 : IR^5 \rightarrow IR^3,$$

such that $p^1 op = id_{IR^3}$. We have discussed our calibration procedure and choice of pseudo-inverse at length in previous publications [10, 9].

For simplicity, we have chosen to model the ball's flight dynamics as a point mass under the influence of gravity. A position-time-sampled measurement of this dynamical system will be described by the discrete dynamics,

$$w_{j+1} = F^*(w_j) \triangleq A_s w_j + a_s;$$

$$A_s \triangleq \begin{bmatrix} I & s \tau \cr 0 & I \end{bmatrix}; \quad a_s \triangleq \begin{bmatrix} \frac{1}{2} s^2 a \cr \frac{1}{2} s^2 \end{bmatrix}$$

$$b_j = C w_j; \quad C = [I, 0],$$

\textsuperscript{3}In the case of a one degree of freedom arm we found that a simple PD controller worked quite effectively [2]. In the present setting, we have found it necessary to introduce a nonlinear inverse dynamics based controller [12]. The high performance properties of this controller notwithstanding, our present success in achieving a spatial two-juggle has required some additional "smoothing" of the output of the mirror law as described in [6].
where $s$ denotes the sampling period, $\bar{a}$ is the gravitational acceleration vector, and $u_{ij} \in \mathbb{R}^d$.

2.3. Signal Processing: A Linear Observer

Following Andersson's experience in real-time visual servoing [1] we apply a first order moment computation to a small window of a threshold-sampled (thus, binary valued) image of each field. Thresholding is the only "early vision" strategy required in a visually structured environment, and we presently illuminate white ping-pong balls with halogen lamps while putting black matte cloth covering on the robot, floor, and curtaining off any background scene. Thus, the "world" as seen by the cameras contains only one or more white balls against a black background. When only one white ball is presented, the camera system reports a pair of pixel addresses, $v \in \mathbb{R}^2$, containing the centroid of the single illuminated region seen by each camera. A field from each camera is acquired with period $\tau_f$, and centroid computations over a small sub-window of 30 by 40 pixels follow during the next $\tau_f$ seconds. Finally triangulation is performed to extract the ball's spatial position from the centroid measurements.

A linear observer can recover from these discrete time measurements an estimate of the full state (positions and velocities). As described above, the window operates on pixel data that is at least one field old,

$$p_k = F^{-\tau_f} (w_k),$$

to produce a centroid. We use $p_k$ as an "extra" state variable to denote this delayed image of the ball's state. Denote by $W_k$ the function that takes a white ball against a black background into a pair of thresholded image plane regions and then into a pair of first order moments at the $k^{th}$ field. The data from the triangulator may now be written as

$$\tilde{b}_k = p^i \circ W_k \circ p(Cp_k). \tag{3}$$

Thus, the observer operates on the delayed data,

$$\hat{p}_{k+1} = F^{\tau_f} (\tilde{p}_k) - G(C\hat{p}_k - \tilde{b}_k), \tag{4}$$

where the gain matrix, $G \in \mathbb{R}^{6 \times 3}$, is chosen so that $A_{\tau_f} + GC$ is asymptotically stable — that is, if the true delayed data, $Cp_k$, were available then it would be guaranteed that $\hat{p}_k \rightarrow p_k$.

In principle, one might choose an optimal set of gains, $G^*$, resulting from an infinite horizon quadratic cost functional, or an optimal sequence of gains, $\{G_t\}_{t=0}^\infty$, resulting from a $k$-stage horizon quadratic cost functional (probably a better choice in the present context), according to the standard Kalman filtering methodology. Of course, this presumes rather strong assumptions and a significant amount of a priori statistical information about the nature of disturbances in both the free flight model (2) as well as in the production of $\bar{b}$ from $\bar{a}$ via the moment generation process. To date we have obtained sufficiently good results with a common sense choice of gains $G$ that recourse to optimal filtering seems more artificial than helpful. We provide our juggling algorithm with an appropriately extrapolated and interpolated version of these estimates as follows. The known latency is corrected by the prediction,

$$\hat{w}_k = F^{\tau_f + \epsilon_k} (\hat{p}_k),$$

where $\epsilon_k$ denotes the time required by the centroid computation at the $k^{th}$ field. Subsequently, the mirror law is passed the next entry in the sequence,

$$F^{\tau_f} (\hat{w}_i), (i = 1, ..., \tau_f - \epsilon_{k+1})$$

until the next estimate, $\hat{p}_{k+1}$ is ready.

2.4. Sensing Issues Arising From the Two-Juggle

Initial attempts to implement the spatial two-juggle (simultaneously batting of two balls), demonstrate that the simple system described above required $\varepsilon$ number of modifications. Central to juggling two balls is the ability for the sensor system to withstand transient losses of data for one or both cameras. These events arise from balls passing outside the field of view of one or both cameras and from balls passing arbitrarily close together in the image planes of the cameras.

For reasons reported earlier [7, 8] we have chosen to rely on the observer's prediction of the balls flight across such events, rather than resorting to more complicated algorithms or fundamentally different strategies. Expanding the moment computations performed on a window to include the zeroth and second order moments affords a reliable measure of the "ballness" of an object, and allows for selective rejection of images with unreliable first order moments (caused either by having either two or no balls present in the window)$^4$. Unfortunately, a loss of data from one camera precludes triangulation and, thus, update of the ball's state by means other than simple integration of the flight model (2). This amounts to setting $G = 0$ in (4) whenever there is an occlusion event.

$^4$As reported in [7] the potential for skipping images and dealing with prolonged out-of-frame events has necessitated that the output of the observer be used to locate the windows in successive images, as opposed to simply centering the windows on the previous measurements. Thus the observer is being used both to drive the juggling algorithm and locate the subwindows used for centroid computations mentioned above.
3. Dynamic Triangulation

Underlying our new estimation technique is the simple idea of augmenting the standard (linear) Newtonian flight model, \( \tilde{b} = \hat{a} \), with a nonlinear “output map,” \( v = p(\tilde{b}) \), where \( p \) denotes the camera transformation introduced in Section 2.1. We now present this procedure, a substitute for (4), that promotes the use of each camera’s data when available.

In Section 3.1, we will note a useful algebraic property of rational linear transformations that relates image plane errors to cartesian errors through a scaled version of the camera jacobian. In Section 3.2 we introduce a “toy” first order observer theory for purposes of illustrating the utility of the previous algebraic fact. The convergence properties of this version of the procedure are simple to establish. Of course, the physical plant of interest is characterized by second order dynamics, and in Section 3.3 we offer an extension of the simple first order procedure to this setting. Arguments establishing the favorable convergence properties of this algorithm are presently in progress.

3.1. Camera Model Properties

Recall that the stereo camera transformation, \( p \), is formed by stacking together the perspective projections due to the two individual cameras,

\[
v \triangleq p(b) = \begin{bmatrix} \frac{1}{\Pi_3} \frac{1}{\Pi_6} \Pi_{1,2} 1 H_0 b \\ \frac{1}{\Pi_3} \frac{1}{\Pi_6} \Pi_{1,2} 2 H_0 b \end{bmatrix}.
\]

(5)

Here, \( 1 H_0 \) is the the homogeneous-matrix representation of the base frame in the \( i \)th camera’s frame, \( f_i \) is the \( i \)th camera’s focal length, \( \Pi_i \) denotes projection onto the \( i \)th component, and \( \Pi_{1,2} \triangleq \frac{\Pi_1}{\Pi_2} \).

In this section we note that

\[
p(\tilde{b}) - p(b) = \Lambda(b, \tilde{b}) P(b) (\tilde{b} - b)
\]

(6)

where \( P(b) \) is the jacobian of \( p \) evaluated at \( b \) and

\[
\Lambda(b_1, b_2) \triangleq \begin{bmatrix} \frac{1}{\Pi_3} \frac{1}{\Pi_6} H_0 b & 0 \\ 0 & \frac{1}{\Pi_3} \frac{1}{\Pi_6} H_0 b \end{bmatrix}.
\]

To see this, begin by defining

\[
d_1 = \Pi_3 1 H_0 b \\
d_2 = \Pi_3 2 H_0 b
\]

for \( i \in \{1, 2\} \). Then with \( v \) and \( \tilde{v} \) the projections of \( b \) and \( \tilde{b} \) respectively we form

\[
\tilde{v} \triangleq \tilde{v} - v = \begin{bmatrix} \frac{1}{d_1} \Pi_{1,2} 1 H_0 \left( d_1 \tilde{b} - d_1 b \right) \\ \frac{1}{d_2} \Pi_{1,2} 2 H_0 \left( d_2 \tilde{b} - d_2 b \right) \end{bmatrix}.
\]

(7)

Substituting for the \( d_i \) terms inside the parenthesis yields

\[
\tilde{v} = \begin{bmatrix} \frac{1}{d_1} \Pi_{1,2} 1 H_0 b \begin{bmatrix} J_1 \end{bmatrix} \begin{bmatrix} 1 H_0 b \end{bmatrix} \\ \frac{1}{d_2} \Pi_{1,2} 2 H_0 b \begin{bmatrix} J_2 \end{bmatrix} \begin{bmatrix} 2 H_0 b \end{bmatrix} \end{bmatrix},
\]

(8)

where

\[
J_1 = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, J_2 = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}
\]

Finally taking the jacobian of (5) results in

\[
P(\tilde{b}) \triangleq D_3 P = \begin{bmatrix} \frac{1}{d_1} \Pi_{1,2} 1 H_0 \begin{bmatrix} J_1 \end{bmatrix} \begin{bmatrix} 1 H_0 b \end{bmatrix} \\ \frac{1}{d_2} \Pi_{1,2} 2 H_0 \begin{bmatrix} J_2 \end{bmatrix} \begin{bmatrix} 2 H_0 b \end{bmatrix} \end{bmatrix},
\]

(9)

which confirms the assertion of (6).

Throughout the remainder of this paper we will assume that \( A \) is positive definite. Geometrically this requires that both \( b \) and \( \tilde{b} \) always lie on the same side (front/back) of all the cameras at all times. In practice this is not an unrealistic assumption since it merely requires that neither the actual object cross the singularity in \( p(\tilde{b}) \) nor that the initial errors in the observer system become so large as to cause the estimated object location to cross this same singularity. Formally, this assumption implies

\[
(\tilde{b} - b)^T P(\tilde{b})(\tilde{v} - v) > 0.
\]

(10)

3.2. A First Order Observer

Consider the dynamical system described by

\[
\begin{align*}
\dot{b} &= Ab + u \\
v &= p(\tilde{b})
\end{align*}
\]

(11)

This system is “fully-measurable” – algebraic triangulation can fully reconstruct it’s state \( b \). But it seems helpful to underscore the utility of (6) by considering an observer for this system to both filter the resultant state estimates and allow for estimation during loss of measurements. The observer takes the form

\[
\begin{align*}
\dot{\tilde{b}} &= A \tilde{b} + u - KP(\tilde{b})(\tilde{v} - v) \\
\dot{\tilde{v}} &= p(\tilde{b}),
\end{align*}
\]

(12)
with $K \in IR^{3 \times 4}$. Forming the error dynamics for $\hat{\eta} \triangleq \hat{v} - v$ we have
\[
\dot{\hat{b}} = A\hat{v} - K P_T(\hat{b})(\hat{v} - v).
\] (13)

Making use of (6) allows us to substitute for $(\hat{v} - v)$, and results in
\[
\dot{\hat{b}} = \left(A - K P_T(\hat{b})A(\hat{b}, b)P(\hat{b})\right)\hat{b}.
\] (14)

From (10) we know $P^T(\hat{b})A(\hat{b}, b)P(\hat{b})$ is positive definite from which it follows that there exists a $K$ such that\footnote{Note, due to the time-varying nature of (14), the choice of stabilizing $K$ is necessarily influenced by the initial conditions. In practice a reasonable bound could be placed on the initial errors such that a suitably large fixed $K$ might readily be chosen.} $\lim_{t \to \infty} \hat{b} = 0$.

### 3.3. An Observer for Mechanical Systems with Linear Dynamics

In contrast to the completely measurable system presented above, let us now reconsider the system, $\hat{b} = \hat{a}$, written more generally as
\[
\begin{align*}
\dot{b}_1 &= b_2 \\
\dot{b}_2 &= A_1b_1 + A_2b_2 + u \\
v &= p(b_1),
\end{align*}
\] (15)

where $b_1$ and $b_2$ represent the position and velocity of the object respectively. The associated observer now takes the form
\[
\begin{align*}
\dot{\hat{b}}_1 &= \hat{b}_2 - \Gamma_1 P_T(\hat{b}_1)(\hat{v} - v) \\
\dot{\hat{b}}_2 &= A_1\hat{b}_1 + A_2\hat{b}_2 - \Gamma_2 P_T(\hat{b}_1)(\hat{v} - v) \\
v &= p(\hat{b}_1),
\end{align*}
\] (16)

with gain matrices $\Gamma_1$ and $\Gamma_2$ free to be chosen. Proceeding as above we take differences to determine the error dynamics
\[
\begin{align*}
\dot{\hat{b}}_1 &= \hat{b}_1 - \Gamma_1 P_T(\hat{b}_1)\hat{v} - v) \\
\dot{\hat{b}}_2 &= A_1\hat{b}_1 + A_2\hat{b}_2 - \Gamma_2 P_T(\hat{b}_1)(\hat{v} - v),
\end{align*}
\] (17)

which simplifies to
\[
\begin{align*}
\dot{\hat{b}}_1 &= \left(I - \Gamma_1 P_T(\hat{b}_1)A(\hat{b}_1, b_1)P(\hat{b}_1)\right)\hat{b}_1 \\
\dot{\hat{b}}_2 &= \left(A_1 - \Gamma_2 P_T(\hat{b}_1)A(\hat{b}_1, b_1)P(\hat{b}_1)\right)\hat{b}_1 + A_2\hat{b}_2.
\end{align*}
\] (18)

A proof of convergence for this system is in progress.

### 4. Implementation

Although the analysis of the previous section is at best in its infancy we have forged ahead with a number of simulations and a functional implementation of the class of observer described here. As usual, the real world departs from the assumptions underlying these models in certain important regards. What follows is a brief discussion of the differences between the previous section and the actual system, along with both experimental and simulation results demonstrating the utility and pitfalls for this type of observer.

#### 4.1. Choice of Observer Gains

Having no immediate insight at the outset concerning choice of the gain matrices $\Gamma_1$ and $\Gamma_2$ in (16), we chose to use the same gains as for the linear observers. Poor convergence in our first simulations demonstrated that this simple choice was inadequate. The primary cause for this effect was that the spatial dependence of $P(\hat{b})$ leads to widely differing effective gains depending on the ball's location in space. We were able to successfully compensate for this by making use of non-linear gain matrices of the form
\[
\Gamma = \Gamma_0 \left(P^T(\hat{b})P(\hat{b})\right)^{-1}.
\]

This essentially amounts to performing local triangulation (i.e. $\Gamma P_T(\hat{b})$ is the linear approximation to $P$ at $\hat{b}$), and dramatically improved the convergence behavior of the observer.

![Figure 2. Simulation: Convergence of the continuous and discrete time observers for small initial error.](image)
affords a reliable observer (4). However, no analogous theory is available for our new nonlinear dynamic triangulator.

In the absence of any theory, we have numerically studied both the continuous and discrete time systems. Figures 2 and 3 demonstrate how a change in the initial conditions can result in instability for the discrete system, while the continuous version remains well-behaved. Figure 2 depicts a case where the discrete and continuous systems demonstrate comparable behavior for identical gains and small initial errors. Figure 3 demonstrates that the same systems can display markedly different behavior for different initial conditions. In this particular example, the continuous system converges reasonably quickly, while the discrete version initially behaves reasonably well, then slowly begins to fail until 5.5 seconds, when it “explodes”.

4.3. The Dynamic-Triangulator and Juggling

Although we are forced to implement the discrete time version of this observer for the juggling system, we are fortunate in that it is relatively easy to control the initial errors. In particular, the observer is started only after a ball comes into the view of both cameras. This allows us to triangulate at start-up and correctly initialize the position estimate. Since the balls are always manually presented, we can safely initialize the velocity estimate to zero.

The primary benefit of this observer scheme is its ability to make use of data from a camera even if there is no data from the other camera. This has ramifications for the larger juggling system, since a ball will often pass outside the field of view of one or both cameras for periods of time in excess of 0.25 sec. Figure 4 demonstrates the difference between this observer and the triangulator/linear-observer system in just such a situation. Figure 4(a) shows the overall flight of the ball as estimated by both observers, and measured by the triangulator (absence of the solid line indicates that the ball was out of frame). In this example, the ball travels out of frame for approximately 0.2 sec. As can be seen in Figure 4(b) (a blowup of the ball returning into the field of view), the dynamical triangulator is capable of updating its estimate while the triangulator/observer pair are forced to simply predict the trajectory (note the differing behavior from 1.05 to 1.10 seconds). Significant reduction in tracking error then results as the ball reappears in both camera’s fields of view at 1.10 seconds. This anecdotal picture is confirmed by experimental statistics. Figure 5 shows
Figure 5. Experimental Data: Mean and standard deviation for the spatial observer errors immediately after recovery from out of frame, averaged over 102 events.

The mean and standard deviation of the norm squared tracking errors (position only) for the first four frames after recovery from an out of frame event for 102 typical events.

5. Conclusion
In this working paper we have
- discussed certain algebraic properties of camera maps that suggest a new class of state estimators for cartesian dynamical systems
- explored the nature of these estimators through various simulations
- successfully implemented a particular estimator for this class in our laboratory juggling robot.

Note that we have not proposed an “image-plane observer” as discussed in [5], since our estimates reside in cartesian space not the image planes. Yet, as an immediate benefit, we need no longer throw away data from both cameras when one of them is occluded. In the longer term, these ideas may provide a framework for turning more general problems in sensor fusion into nonlinear observer problems.

The price we pay for these advantages is manifest at once in
- the need for a new observer theory for linear systems with projective output; maps to aid in choice of gain settings and provide convergence guarantees;
- the greater practical risks of implementing in discrete time a nonlinear continuous-time dynamical system (for which we cannot foresee the development of any principled sampling theory).

Further study — theoretical, simulation, and experimental — seems needed before a verdict on the attractiveness of this scheme can be fairly rendered.

References