Active Control of Resolution for Stable Visual Tracking

Nicola Ferrier

University of Wisconsin, Madison WI 53706, USA,
ferrier@engr.wisc.edu,
WWW home page: http://mechatron.me.wisc.edu

Abstract. Success of visual tracking typically relies on the ability to process visual information sufficiently fast. Often a dynamic system model of target motion is used to estimate the target location within the image and a region of interest (ROI) is used to reduce the amount of image data processing. This has proven effective, provided the ROI is sufficiently large to detect the target and sufficiently small to be processed quick enough. Here we formally consider the size of the ROI and the resolution of the ROI to ensure that tracking is stable. Dynamic system formulation of visual tracking usually requires specification of the dynamics of the target. We analyze motions which can be described by linear time-invariant dynamical systems (although the image motion may be highly non-linear). One can successfully analyze the required ROI size and resolution to ensure stable tracking.

1 Motivation

Planning and modeling of sensor utilization in intelligent systems typically covers topics such as how to fuse multiple sensor inputs, or how to utilize a particular sensor for a task such as navigation or map building. For visual control of motion, sensor planning typically involves determining which algorithms to use to process the visual data. A key competency required in order to use vision for motion control (grasping or manipulation) is to be able to locate and track a target. Conventional visual servo control requires continuous tracking of a target, at great computational expense. We are looking to relax the continuous tracking requirement.

Previous systems that use vision for control of manipulation or navigation typically must track the object of interest. Processing closely sampled images enables the use of temporal coherence to facilitate tracking. For advanced systems, the ability to simultaneously track multiple objects (such as a driver moving his eyes to objects/points on either side of the road) either forces the use of multiple visual sensors *or* one must relax the temporal coherence requirement. We are looking to do the later. The first step towards this is to be able to understand the relationship between the target dynamics, system and measurement noise, and time delay and their impact on successful tracking. This chapter develops a tracking model, based on traditional Kalman filter tracking systems, that can successfully track given non-fixed time delays between measurements. Initially we assume that the delay is based on computation time (although the delay could be due to other reasons such as “time sharing” the sensor between two or more targets). If there is noise in the system model (as there will be in any practical system), as delay time increases, the positional uncertainty of the target will also increase. The region in the image that must be searched to ensure successful tracking will also increase (thus more computation is required to process the data), and we have a negative feedback system - tracking could become unstable (or effectively non-observable as the delay between observations grows). This chapter presents an algorithm to actively control the resolution of the processed image window to avoid this growth in computation time. While the idea of using variable sized search windows has been used before, we develop the underlying relationships between the system parameters and the required window size and resolution. We show via a simulation and a simple system how active control of the resolution of the image data processed can lead to stable tracking.

2 Background on Visual Tracking

Visual tracking, that is, extracting geometric information of the motion of one or many objects, is a central problem in vision-based control and there has been considerable effort to produce real-time algorithms for tracking moving objects. Typically tracking seeks to determine the image position of a target as the target moves through the camera's field of view. Successive image frames (usually taken at closely spaced intervals) are processed to locate the features or target region. Difficulties in target recognition arise due to the variation in images from illumination, full or partial occlusion of the target, effects of pose on the projected image. Given that the target can be recognized, tracking difficulties arise due to noise (modeling uncertainty) and the time delay required to process the image data.

Visual control systems must take into account the delays associated with the visual computation. Tracking stability can only be achieved if the sensing delays are sufficiently small, or, if the dynamical model is sufficiently accurate. Various approaches have been taken to compensate for the delay of the visual system in visual servo control. Many the tracking systems use *dynamic* models of target motion, and use the Kalman-Bucy filter for improved tracking performance (see e.g. [4, 12, 16, 17, 22]). Successful visual tracking for servo control has been achieved by balancing the trade off between the accuracy of the sensing and the visual data processing time. Roberts *et al* [20] show that for systems for which the dynamical model is known with sufficiently accuracy, the search area (window size or resolution) can be kept small enough such that the processing time is sufficiently small for accurate tracking. Clark & Ferrier [3, 6] utilize feed-forward compensation for the processing delay. The dynamical model must be known with sufficient accuracy to produce stable tracking. Schnackertz & Grupen [21] also utilize a feed-forward term. Similar feed-forward, or predictive, techniques are employed in visual tracking techniques which incorporate the computation delay (expressed as the number of frames 'dropped' during the compute cycle) within the dynamical model to predict the tracker motion [2]. Nelson *et. al.* [15] assume commensurate delays and model the delay d within the system model and a new control law is developed to account for this delay. One feature uniform to these methods is that, after estimating the approximate image location of the feature, a region of the image must be searched in the next frame to locate the target (either to match the model for wire frame and/or solid models, or locate the image feature point(s), etc.). Given the uncertainty of the estimated position (the covariance), the size of the search window can be modified to reduce computation time. For sufficiently small delays, tracking is successful, however none of these systems have *quantitatively* analyzed the bounds on the accuracy of the performance, the computational delay and the system noise.

Assuming a recursive estimation scheme for tracking, this paper demonstrates that even for simple linear dynamic models, the tracking system can become effectively *unobservable* for particular dynamics, noise models, or processing algorithms. As the size of the search window grows, the measurement sample time increases. This time delay can cause the tracking system to become unstable. Olivier [18] develops some mathematical underpinnings for this analysis, however his work is primarily restricted to scalar stochastic systems.

In this paper we concentrate on the ability of Kalman-Bucy filter based tracking systems to successful track objects using a *search window based* algorithm. We will show that for *search window based* tracking: 1) If computation time is proportional to the estimated covariance then the sequence of measurement times $(t_1, t_2, \ldots, t_k, \ldots)$ can diverge (becomes unstable), and 2) variable resolution can be used to stabilize the tracking (bound the time delay). The dynamic system analysis for tracking commonly used is presented in section 3. We analyze search window based methods giving analytic and experiment results demonstrating our claim.

3 Dynamics System Formulation of Tracking

The ability to describe, or estimate, the motion of the object with respect to the camera enables the tracking system to restrict the search to a local region centered on the estimated position. Point based methods typically assume closely spaced image sequences in order to use correlation methods (and often dynamics are not used under the assumption that the images are closely enough spaced that the point motion is Brownian, so the search window is centered on the current location). The motion of the points, lines or region is typically described using a stochastic dynamic system model. The *state*, $x(t)$, depends on the representation used. For example, for point based tracking x may be a vector of point coordinates, while for b-spline based tracking x may be a vector of control points, and for region based tracking x may be a vector of coordinates of the centroid of the region.

For continuous linear systems we describe the evolution of the state with the equation

$$dx(t) = A(t)x(t)dt + \mathit{\Gamma}(t)dw_t, \quad t \geq t_o \tag{1}$$

where the state $x(t)$ is an n-vector, A is an $n \times n$ continuous matrix time function, $\mathit{\Gamma}$ is an $n \times r$ continuous matrix time function, w_t is a r-vector Brownian motion process with expected value $\mathcal{E}[dw_t dw_t^T] = Q(t)dt$. At time instants t_k discrete linear measurements are taken:

$$y(t_k) = C(t_k)x_k + v_k; \quad k = 1, 2, \ldots; \quad t_{k+1} > t_k \geq t_o \tag{2}$$

where y_k is the m-vector observation and C is an $m \times n$ nonrandom bounded matrix function. v_t is a m-vector white, Gaussian sequence, $v_k \sim N(0, R_k)$, $R_k > 0$. The initial state is assumed to be a Gaussian random variable, $x_{t_o} \sim N(x^*_{t_o}, P_{t_o})$. The random variables x_{t_o}, v_k and w_t are assumed independent. The covariance of the state is given by $P(t) = \mathcal{E}[(x(t) - x^*(t))(x(t) - x^*(t))^T]$. The *actual* (true) state is denoted $x^*(t)$. Here we are using a continuous-discrete system. The system (motion of an object) is a continuous process, whereas measurements are taken at discrete times. Jazwinski [13] points out that analysis of the continuous-continuous system and the continuous-discrete system are essentially equivalent.

Under these assumptions, estimation is often performed using the Kalman-Bucy filter (see e.g. [4, 12, 16, 22]). Under the assumed statistical assumptions, the Kalman-Bucy filter provides a state estimate that is an optimal trade-off between the extrapolated state under the system dynamical model and the measured state.

Most tracking systems assume *equally spaced* measurements. Here we *do not* make that assumption in order to evaluate the effect of the search size on the tracking performance. Measurements are taken at discrete instances $(t_1, t_2, \ldots t_k, t_{k+1}, \ldots)$. The time of the measurements effects tracking stability. Previous work has considered the optimal measurement schedule where the growth of the covariance is used to determine *when* to take measurements [14]. Due to the time taken for the processing of image data, the problem with visual tracking is not one of determining *how long* to wait before taking a measurement, but determining how to measure *often enough* to ensure stability. Here we assume one finishes processing previous measurements before taking the next measurement. If the time between measurements $|t_{k+1} - t_k|$ grows, then tracking fails (we call this situation unstable tracking).

For linear time invariant systems, the estimate of the state can be found from the state transition matrix on the interval $[t_k, t_{k+1}]$,

$$\hat{x}(t_{k+1}) \; = \; \mathit{\Phi}(t_{k+1}, t_k)x(t_k) \; = \; e^{A(t_{k+1} - t_k)} \; \hat{x}(t_k) \tag{3}$$

and the covariance is determined (before measurement) from the Lyapanov equation [1]:

$$P(t^-_{k+1}) = e^{A(t_{k+1} - t_k)} P(t^+_k) e^{A^T(t_{k+1} - t_k)} + Q(t_{k+1}, t_k) \tag{4}$$

[1] We adopt the notation $P(t^-_k)$ to denote the covariance at time t prior to any measurement. In other notation this may be denoted $P(t_k|t_{k-1})$ and $P(t^+_k)$ denoted $P(t_k|t_k)$

where

$$P(t_k^+) = \left(P^{-1}(t_k^-) + C^T(t_k)R_k^{-1}C(t_k)\right)^{-1} \tag{5}$$

and

$$Q(t_{k+1}, t_k) = \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-\tau)}Q(\tau)e^{A^T(t_{k+1}-\tau)}d\tau$$

is the integrated noise over the interval (t_k, t_{k+1}). The covariance immediately before a measurement at time t_{k+1} is given by:

$$P(t_{k+1}^-) = e^{A(t_{k+1}-t_k)}\left(P^{-1}(t_k^-) + C^T(t_k)R_k^{-1}C(t_k)\right)^{-1}e^{A^T(t_{k+1}-t_k)} + Q(t_{k+1}, t_k) \tag{6}$$

We will utilize this expression later in determining the computational delay for tracking. The ability to determine the growth of the covariance requires solving this Riccati-equation. For certain cases the non-linear coupled differential equations in the elements of P can be solved (see e.g. Gelb [10], chapter 4, example 1 presents four methods to solve the Riccati equation for a second-order integrator with no system noise).

4 Computation Time and Covariance

In this section we explore the relationship between the covariance and the computation time.

Suppose the location of a target is a pixel location within a frame (e.g. point coordinate or centroid of a region). If the entire image frame is searched, the computation time is a function of image size. Often one utilizes a model of object motion then a search window or region of interest (ROI) can be centered on the predicted target location. The covariance of the estimated target location can be used to determine the size of the search window. Although there have been suggestions to utilize the covariance to determine ROI size [2], in practice the ROI is a fixed size or a limited set of sizes [19, 20]). If the ROI is chosen to be proportional to the covariance (typically a 2σ or 3σ window size) then under the assumed Gaussian noise models, with high probability the target should be located within the given ROI. If we let the ROI size vary then the size of the ROI affects the processing time. If the algorithm must "visit" each pixel a certain number of times (assumed at least once), then the processing time is proportional to the ROI size:

$$t_{k+1} = t_k + \beta\|P(t_k^-)\|$$

where the *size* of the covariance, denoted $\|P(t_k^-)\|$, represents the number of pixels to be processed. If the ROI size (and hence the covariance) grows, so does the processing time and hence the system can become unstable. Figure 1 graphically depicts this idea. The solid line represents the covariance in a target position *before* a measurement has been taken. The size of this region determines the amount of data to be processed and hence effects the time of the next possible measurement. The dashed ellipse represents the uncertainty in a target position

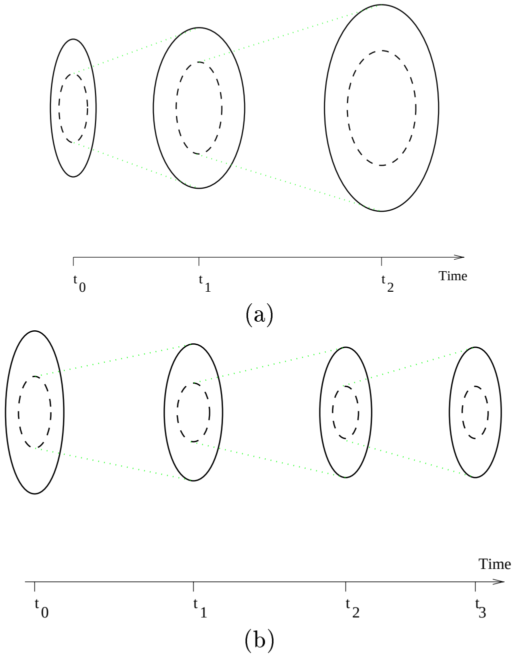


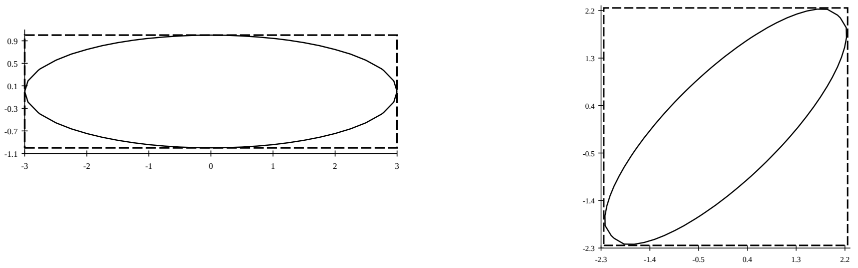
Fig. 1. Graphic representation of growth in covariance. The solid line represents the covariance before a measurement, $P(t_k^-)$. The dashed line represents the covariance after the measurement has been processed, $P(t_k^+)$. The time between measurements depends on the size of the covariance. (a) The covariance shown here increases in size and so the time delay $t_{k+1} - t_k$ increases. This tracking becomes effectively unobservable. (b) In this instance, the covariance decreases in size to a fixed size and so the time delay $t_{k+1} - t_k$ converges to a fixed interval.

after a measurement has been taken. The size of this ellipse grows over the time interval before the next measurement can be taken.

If the covariance is expressed with respect to image coordinates (for image based systems) then the covariance can be used to describe an ellipsoidal uncertainty region. For example a 2σ ellipse centered about $(\hat{x}, \hat{y})$ is given by:

$$\left\{ (x, y) \mid [x - \hat{x}, y - \hat{y}] \; P^{-1} \; \begin{bmatrix} x - \hat{x} \\ y - \hat{y} \end{bmatrix} \leq 2 \right\}$$

In order to ensure that the target is found, the entire ellipse must be searched. The exact elliptical region could be searched, for example, fast region filling algorithms from computer graphics [8, 11] can be adapted to extract the elliptical region from the image. Alternatively, a bounding rectangle of the uncertainty region can be used. Based on current technology with linear scanning of CCDs images using frame-grabbers, the latter is often used. The major and minor axes of the ellipse are determined by the eigenvalues of the covariance matrix and the desired search scale (2σ). For an ellipse with major/minor axis lengths of



(a) (b)

Fig. 2. The bounding rectangles (aligned with the image coordinate frame). In (a) the ellipse is optimally aligned with the image axes. In (b) the worst case alignment occurs with the ellipse rotated 45^0

a and b, in the worst case, the bounding rectangle will have area $2(a^2 + b^2)$. The bounding rectangle in the the optimal case (the major and minor axes are aligned with the image CCD/frame-grabber axes) has area $4ab$ (and note that the ellipse has area πab). The major and minor axes of the ellipse are given by the maximum and minimum eigenvalues of the covariance. For a σ search region the area of the bounding box for the elliptical ROI is given by

$$4(\lambda_{max})(\lambda_{min}) \leq Area(ROI) \leq 2\left(\lambda_{max}^2 + \lambda_{min}^2\right)$$

where λ denotes the eigenvalues of P. Thus we can obtain bounds for the "size-of" operator, $\|P(t_k^-)\|$, and hence on the computational delay. In general the axes of the uncertainty ellipse will not be aligned with the axes of the image (the lower bound).

5 Covariance Growth and Resolution Control

Tracking with a variable sized ROI will succeed or fail based on the size of the ROI and the computation performed on the ROI. Here the ROI size depends on the growth of the covariance $P(t_k^-)$. P evolves according to the Riccati equation given in equation (6). We noted earlier that the growth of the covariance depends on the dynamics, A, the time interval $(t_{k+1} - t_k)$, and the noise models and that solving the Riccati equation analytically can only be performed in special cases.

We note that the measurement resolution, R, will be determined by the chosen search procedure (algorithm). Centroids can be located to sub-pixel accuracy, thus R will typically be on the order of a pixel squared. This accuracy is not unrealistic for window search approaches. For SSD tracking [1, 19] give estimates for the uncertainty of the computed target position.

If the covariance is too large, one can decrease the resolution of the processing within the image by sub-sampling the ROI or using image pyramid algorithms [15, 23]. Modifying the resolution of the processing will decrease the accuracy of the measurement but it will also decrease the time delay. A scalar system analysis (from Olivier [18]) is summarized in the appendix. Below we consider a simple dynamical system. We will demonstrate the growth of the covariance for this simple dynamical system and show how modification of the measurement resolution can lead to stable tracking, at the expense of the accuracy of the target estimate. While others (e.g. Nelson [15]) have used multi-scale approaches, no analysis was given to appropriately determine the correct resolution or window-size. In these works the window-size and resolution were determined *a priori* and thus were fixed parameters in the system. Only targets with particular dynamics or noise models would be track-able with fixed window and resolution sizes. Here we show that procedures that control (modify) resolution can indeed stabilize the tracking system. The intuition in previous tracking systems with multi-resolution searches can be formalized by considering both the covariance size and the resolution of search.

If the ROI is sub-sampled then the measurement resolution R is inversely proportional to this sub-sampling factor. As mentioned above many window based routines can achieve sub-pixel accuracy. The resolution will grow with the sub-sampling factor. In the scalar case (see appendix), one can explicitly derive the relationship between the resolution factor, the time delay and the system dynamics. For higher dimensions close-formed analysis requires solution of equation (6), hence we analyze some simple cases, then resort to simulation and experimentation to demonstrate the utility of resolution control.

5.1 Tracking with Simple Linear Dynamics

To demonstrate the potential instability in tracking and subsequent improvement with resolution control we consider a simple system with constant velocity motion. We analyze a first order integrator with measurement of the position (which is a frequently encountered system in tracking). The continuous-discrete linear system has state $\mathbf{x}^T = \begin{bmatrix} \theta, \dot{\theta} \end{bmatrix}$, and the state dynamics is given by

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ w \end{bmatrix} \tag{7}$$

where $w \sim N[0, q]$ is the system noise and the discrete measurement process is

$$y(t_k) = \begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{x}(t_k) + v_k, \quad v_k \sim N[0, r] \quad t_k \in [t_o, t_1, \ldots] \text{ where } t_{i+1} > t_i$$

The transition matrix for the system matrix given above is

$$e^{A(t_{k+1}-t_k)} = \begin{bmatrix} 1 & (t_{k+1} - t_k) \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & \delta_k \\ 0 & 1 \end{bmatrix}$$

where $\delta_k = (t_{k+1} - t_k)$, and

$$C^T R^{-1} C = \begin{bmatrix} \frac{1}{r_k} & 0 \\ 0 & 0 \end{bmatrix}.$$

The covariances immediately before measurements at times t_k and t_{k+1} are denoted

$$P(t_k^-) = \begin{bmatrix} p_{11}(t_k^-) & p_{12}(t_k^-) \\ - & p_{22}(t_k^-) \end{bmatrix} = \begin{bmatrix} p_{11_k} & p_{12_k} \\ - & p_{22_k} \end{bmatrix}$$

and

$$P(t_{k+1}^-) = \begin{bmatrix} p_{11}(t_{k+1}^-) & p_{12}(t_{k+1}^-) \\ - & p_{22}(t_{k+1}^-) \end{bmatrix} = \begin{bmatrix} p_{11_{k+1}} & p_{12_{k+1}} \\ - & p_{22_{k+1}} \end{bmatrix}.$$

With this system equation (5) becomes

$$P(t_k^+) = \frac{1}{p_{11_k} + r_k} \begin{bmatrix} r_k p_{11_k} & r_k p_{12_k} \\ - & p_{22_k}(p_{11_k} + r_k) - p_{12_k}^2 \end{bmatrix}$$

Using equation (6) and the notation above we solve for $P(t_{k+1}^-)$ and obtain three non-linear difference equations:

$$\begin{aligned} p_{11_{k+1}} &= \frac{1}{p_{11_k} + r_k} \left(r_k p_{11_k} + 2\delta_k r_k p_{12_k} + \delta_k^2 (p_{22_k}(p_{11_k} + r_k) - p_{12_k}^2) \right) + q_{11_k} \\ p_{12_{k+1}} &= \frac{1}{p_{11_k} + r_k} \left(r_k p_{12_k} + \delta_k \left(p_{22_k}(p_{11_k} + r_k) - p_{12_k}^2 \right) \right) + q_{12_k} \qquad (8) \\ p_{22_{k+1}} &= \frac{1}{p_{11_k} + r_k} \left(p_{22_k}(p_{11_k} + r_k) - p_{12_k}^2 \right) + q_{22_k} \end{aligned}$$

where

$$Q(t_{k+1}, t_k) = \int_{t_k}^{t_{k+1}} \begin{bmatrix} (t_{k+1} - \tau)^2 & (t_{k+1} - \tau) \\ (t_{k+1} - \tau) & 1 \end{bmatrix} q(\tau) d\tau = \begin{bmatrix} \frac{1}{3}\delta_k^3 q & \frac{1}{2}\delta_k^2 q \\ - & \delta_k q \end{bmatrix}$$

for constant system noise $dw_\tau^2 = q(\tau) = q$. Note that the evolution of the covariance depends on the initial conditions $P(0)$, the time delay δ_k, the noise covariances q_{ij} and r. The dynamics enter via the state transition matrix. The initial speed and/or position do not enter into the above equations.

To reduce the number of subscripts, we will write $a_k = p_{11_k}$, $b_k = p_{12_k}$, and $c_k = p_{22_k}$ in the following.

5.2 Variable Time and Fixed Resolution Tracking

Here we consider search methods which search at a fixed resolution within the ROI. The size of the ROI is assumed to be proportional to the target positional uncertainty (i.e. δ_k is a function of the size of p_{11}, r_k is constant). We take $\delta_k = 4 * \beta\sqrt{p_{11}}$. The positional covariance is obtained from the 11 element of the covariance matrix, $p_{11} = {\sigma_p}^2$. To search at a fixed resolution within a 2-σ_p region of the estimated position we must process $4\sqrt{p_{11}}$ pixels. Thus $4\sqrt{p_{11}}$ is the

[2] Note that p_{22} represents the velocity variance, and p_{12} is the covariance term between position and velocity. The search window depends only on the current value of the positional uncertainty p_{11}. The other terms will affect the growth of this term, however in determining the search window only the value of p_{11} is considered.

number of pixels to be processed and β is the processing time per pixel giving a time delay due to computation of δ_k.

The difference equations (equations (8)) for this time delay become

$$\begin{aligned} a_{k+1} &= \frac{1}{a_k + r}\left(ra_k + 2\beta a_k^{1/2} rb_k + \beta^2 a_k(c_k(a_k + r) - b_k^2)\right) + q_{11_k} \\ b_{k+1} &= \frac{1}{a_k + r}\left((rb_k + \beta a_k^{1/2}(c_k(a_k + r) - b_k^2)\right) + q_{12_k} \\ c_{k+1} &= \frac{1}{a_k + r}\left(c_k(a_k + r) - b_k^2\right) + q_{22_k} \end{aligned}$$

We solve these numerically. Figure 3 uses default values of $a_o = 25$ (i.e. initial uncertainty in object position is 25 pixels squared), and $b_o = 0$, $c_o = 4$, $r = 1\text{pixel}^2$, $q = 1(\text{pixel}/s)^2$ and $\beta = 0.01s/\text{pixel}$. These plots show that for various combinations of parameters, the time delay, δ_k, can increase with k (and hence the tracking system fails). In figure 3(b) the values of β are varied. For $\beta > 0.127$, the time delay diverges (and hence only the first few measurements for $\beta = 0.13$ were plotted). The interaction of the various parameters is difficult to observe. From Figure 3 one may be led to believe that, say, increasing the system noise cannot cause the system to fail. In Figure 4 the same parameters are used except that β is increased from $\beta = 0.01$ second/pixel to $\beta = 0.025$ second/pixel. The system diverges for much smaller system noise values. (In figure 3(d), had we increased q sufficiently, the time delay diverges).

5.3 Fixed Time Delay with Resolution Control

A more typical case assumes a fixed delay (typically a multiple of the camera frame rate). In order to have a fixed delay, the covariance must be kept small *or*, as we demonstrate here, the resolution must be adjusted.

Using the same notation as above, δ_k is constant but r_k depends on the size of p_{11}. If we take β to be the processing time per pixel, and $\delta_k = \delta$ is the processing time, then δ/β is the number of pixels that can be processed in the given time interval. If $4\sqrt{p_{11}}$ is the window size (max number of pixels) then we can only check every pixel if

$$4\sqrt{p_{11}} < \frac{\delta}{\beta}$$

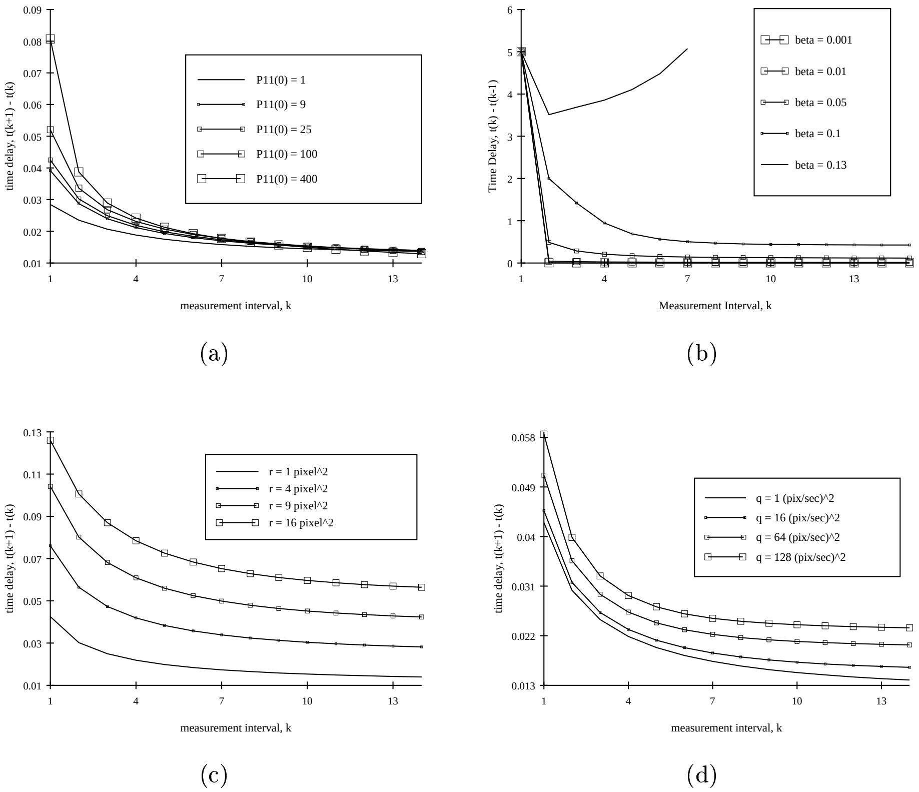
If this condition does not hold then we must process at a lower resolution (subsample). We find n such that

$$\frac{1}{2^n} 4\sqrt{p_{11}} < \frac{\delta}{\beta} \tag{9}$$

i.e.

$$n = \lceil \log_2 \frac{4\beta\sqrt{p_{11}}}{\delta} \rceil \tag{10}$$

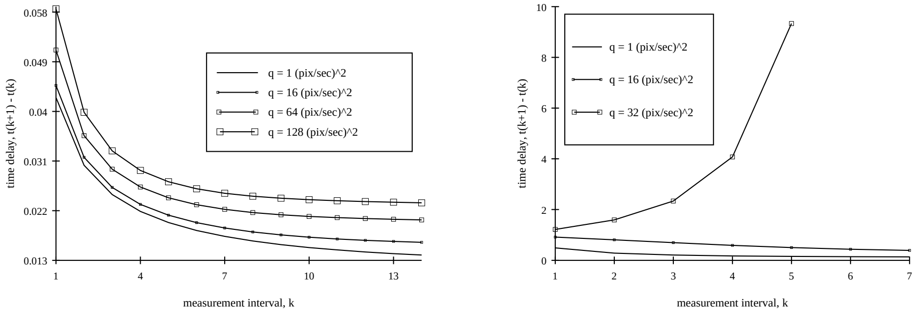
where $\lceil\ \rceil$ denotes the ceiling function and we adjust r accordingly (will increase by a factor of 2^n). Note that we could have used an arbitrary scaling factor



(a) (b)

(c) (d)

Fig. 3. Graphs showing the time delay (δ_k) for variation in system parameters. The plots show the effect of increasing (a) initial positional covariance, P_o, (b) computation time, β, (c) measurement noise, r, and (d) system noise, q. The default values for parameters used are $P_{11}(0) = 25 pixel^2$, $R = 1 pixel^2$, $Q = 1(pixel/sec)^2$, and $\beta = 0.01 sec/pixel$.



(a) (b)

Fig. 4. The time delay (δ_k) diverges at much lower system noise levels when the computation time increases. (a) For $\beta = 0.01$, the time delay did not diverge for $q < 512(pix/s)^2$ (b) $\beta = 0.025$, the time delay diverges for much smaller q.

instead of $\frac{1}{2^n}$, however vision hardware often supports sub-sampling by factors of 2.

Using the above criteria (equation (9)) to set the value of r_k, one must then solve the difference equations (8):

$$
\begin{aligned}
a_{k+1} &= \frac{1}{a_k + r_k}\left(r_k a_k + 2\delta r_k b_k + \delta^2(c_k(a_k + r_k) - b_k^2)\right) + q_{11_k} \\
b_{k+1} &= \frac{1}{a_k + r_k}\left(r_k b_k + \delta\left(c_k(a_k + r_k) - b_k^2\right)\right) + q_{12_k} \\
c_{k+1} &= \frac{1}{a_k + r_k}\left(c_k(a_k + r_k) - b_k^2\right) + q_{22_k}
\end{aligned}
$$

Figure 3 showed various divergent behavior for certain combinations of parameters. Given a fixed delay of $0.05s$, these unstable systems can be stabilized by controlling the resolution of the ROI. The positional covariance (and hence the time delay) in figure 3 (b) and figure 4 (b) diverged. Figure 5 shows these same dynamical system parameters, however the resolution is determined as described above. The positional covariance converges to steady state.

6 Linear Dynamics - Simple Experiments

In this section rather than solve the difference equations for the positional covariance, we perform resolution control during tracking: that is, we actively modify the resolution of the image processed to ensure that the time delay does not diverge. To utilize the methods developed, we consider simple systems that can again be described by linear time-invariant dynamical systems. Suppose we have

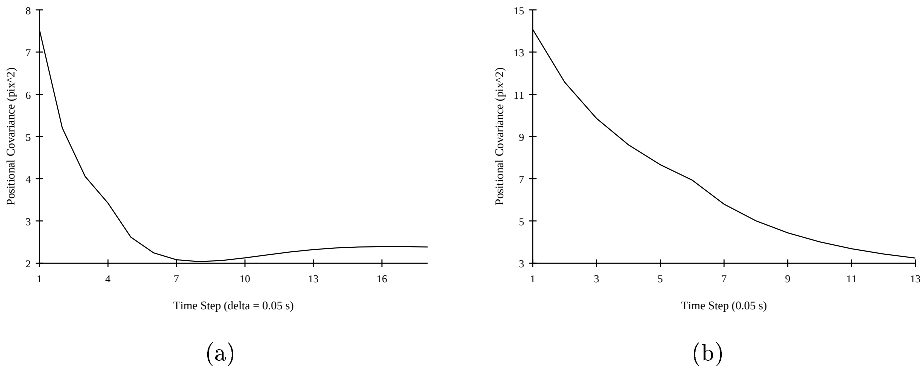


Fig. 5. (a) The parameters used in figure 4(b) caused the covariance (and hence the time delay) to diverge. For fixed time step, using the same parameters but incorporating resolution control we show a convergent positional covariance. The resolution (not shown) is on the order of 8-16 $pixel^2$. (b) For $\beta = 0.13$, tracking failed for the default parameters and fixed resolution. Here we use these same system parameters with resolution control and show a convergent positional covariance.

a circular track on the ground plane described by the path

$$p(\theta(t)) = (r\cos(\theta(t)), r\sin(\theta(t)) = (r\cos(\omega t + \theta_o), r\sin(\omega t + \theta_o)$$

with $p_o = (r, 0)$ and where $\theta(t) = \omega t + \theta_o$.

If we are viewing the motion with an **overhead camera**, the image of the track will also be a circular path[3]. We can view all of the track within the field of view and the path in the image is a circle centered at pixel (u_c, v_c) with radius ρ. The image path is described by

$$(u(t), v(t)) = (u_c + \rho\cos\phi(t), v_c + \rho\sin\phi(t))$$

where

$$\theta(t) = \phi(t) + \phi_o$$

(the angular measure in the image and on the circular path may be offset by a fixed amount ϕ_o). The target centroid is located at pixel $(\hat{u}, \hat{v})$ Assuming that the actual target location is on the path, and we project this detected location to the closest path point, we have a measurement of ϕ of the form:

$$\hat{\phi} = \tan^{-1}\left(\frac{\hat{v} - v_c}{\hat{u} - u_c}\right)$$

The accuracy of the estimated angle depends on the accuracy of the calibrated terms (center of path), and on the accuracy of the extracted target

[3] We comment later on how to relax this assumption

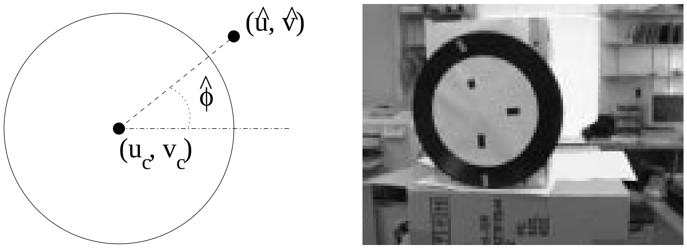


Fig. 6. Measurement of target position in the image. Example of object tracked while moving along a circular path.

location $(\hat{u}, \hat{v})$ A sensitivity analysis of the computation of $\hat{\phi}$ yields

$$\delta\hat{\phi} = \frac{(\hat{u} - u_c)d(\hat{v} - v_c) - (\hat{v} - v_c)d(\hat{u} - u_c)}{(\hat{v} - v_c)^2 + (\hat{u} - u_c)^2}$$

and assuming that the center has been calibrated with high accuracy

$$\delta\hat{\phi} = \frac{1}{(\hat{v} - v_c)^2 + (\hat{u} - u_c)^2} \left((\hat{u} - u_c)\delta\hat{v} - (\hat{v} - v_c)\delta\hat{u}\right).$$

If the target is close to the path then $(\hat{v} - v_c)^2 + (\hat{u} - u_c)^2 \approx \rho^2$ so we have

$$\delta\hat{\phi} = \frac{1}{\rho^2} \left((\hat{u} - u_c)\delta\hat{v} - (\hat{v} - v_c)\delta\hat{u}\right)$$

Assuming a Gaussian distribution for the target location with $\mathcal{E}[\hat{u}] = u_T$, $\mathcal{E}[\hat{v}] = v_T$ where (u_T, v_T) is the actual target location in the image. $\mathcal{E}[\delta\hat{u}^2] = \sigma_u^2$ and $\mathcal{E}[\delta\hat{v}^2] = \sigma_v^2$ depend on the target geometry and image resolution. For symmetric targets and image sampling we can assume $\sigma_u^2 = \sigma_v^2 = \sigma^2$ (i.e. the uncertainty in image target centroid computation) and thus the covariance of the measurement is given by

$$\begin{aligned}\sigma_\phi^2 = \mathcal{E}[\delta\hat{\phi}^2] &= \frac{1}{\rho^4}\left((u_T - u_c)^2\sigma_v^2 + (v_T - v_c)^2\sigma_u^2\right) \\ &= \frac{1}{\rho^4}\left((u_T - u_c)^2 + (v_T - v_c)^2\right)\sigma^2 = \frac{\sigma^2}{\rho^2}\end{aligned}$$

From this we see that it is desirable to have the image of the circular path as large as possible in the field of view (maximize ρ). Note, in practice, ϕ is function of the ratio of two Gaussian random variables. The true distribution will be a Cauchy distribution [7, 9]. The Gaussian assumption for errors in ϕ is only approximate.

The examples given in figure 7 show a system with slow processing. The time delays increase and eventually the tracking is unstable. Figure 8 shows tracking of the same dynamic system using fixed time intervals and resolution control. By sub-sampling the ROI the resolution of processing is controlled to stabilize

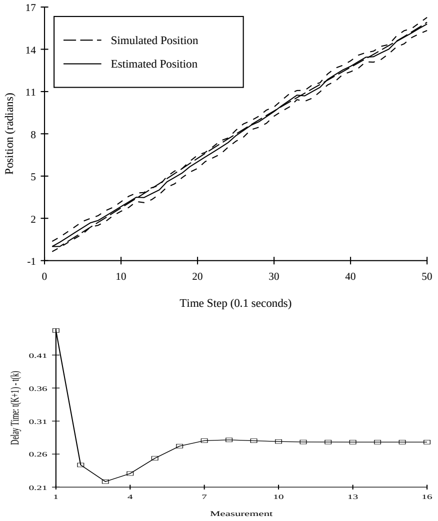
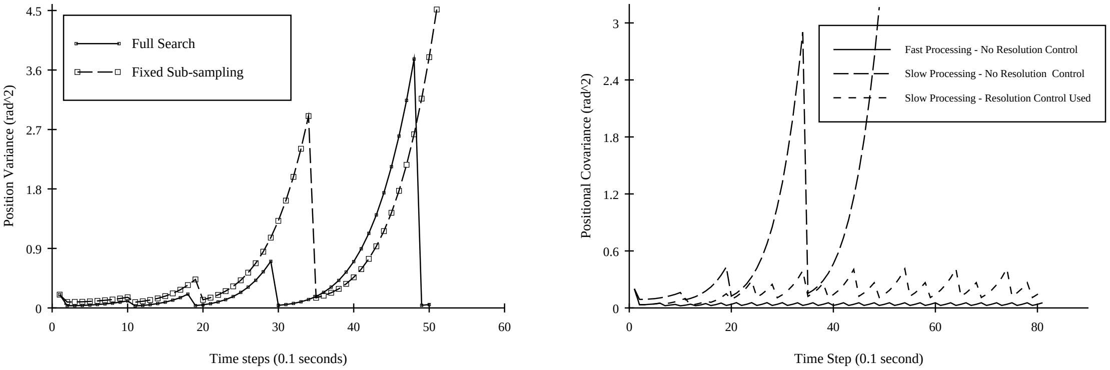


Fig. 7. Estimated position of target with dashed lines showing the position covariance. Right hand side shows doubled processing time. (Initial speed is 33 RPM, system noise is $N[0, 0.04 rad/s^2]$).



(a) (b)

Fig. 8. (a) The covariance for target position shown for full search and sub-sampling. Even with sub-sampling the covariance can grow monotonically. (b) Graph comparing target position covariance for a fast algorithm, a slow algorithm with no resolution control and a slow algorithm using resolution control. Notice that the slow algorithm with resolution control reaches a "steady state" (it oscillates between two covariance values). The "steady-state" covariance is larger than the equivalent dynamic system with fast processing.

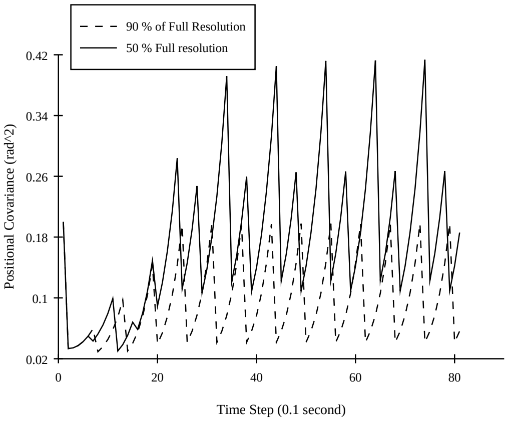


Fig. 9. Comparison of resolution control for two different sub-sampling factors. The dashed line shows tracking at close to full resolution, the solid line shows tracking with 50% resolution

the tracking. Thus in order to process the images as they arrive (say at 30 Hz) the either the ROI size must be adjusted (risking tracking failure) or the ROI must be sub-sampled. Figure 9 shows the resulting positional covariance for two different measurement resolutions. Lower resolution can facilitate or stabilize tracking, at the loss of accuracy of the tracking system.

In steady state in both figure 8 and 9, the positional covariance oscillates between two values. The size of the covariance determines the resolution to use. For particular values of system parameters, the resolution factor n in equation (10) can jump between two values.

6.1 Extensions to other Motion

The examples presented were simple to demonstrate feasibility of actively controlling resolution. There are some simple extensions to the dynamic systems presented.

It is simple to extend the circular motion to a non-fronto parallel camera. If the camera is not directly overhead, but views the ground plane at an oblique angle, the image of the circular path will be an ellipse centered at pixel (u_c, v_c) with major and minor axes of length a and b. On rotated coordinates we have

$$(u'(t), v'(t)) = (u'_c + a\cos\phi(t), v'_c + b\sin\phi(t))$$

which in the image are

$$\begin{bmatrix} u(t) \\ v(t) \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} u'(t) \\ v'(t) \end{bmatrix}$$

The target centroid is located at pixel $(\hat{u}, \hat{v})$ Assuming that the actual target location is on the path, and we project this detected location to the closest path point, we have a measurement of ϕ of the form:

$$\hat{\phi} = \tan^{-1}\left(\frac{a(\hat{v} - v_c)}{b(\hat{u} - u_c)}\right)$$

and

$$\theta(t) = \phi(t) + \phi_o$$

The measured angle provides an estimate of the actual angle given that the path has been calibrated within the image.

Viewing motion on known paths For object motion along known paths, often a linear time-invariant dynamical system can be used to describe the motion (for example, vehicle motion along roads). The path can be determined *a priori* by calibration or from known "maps" (for motion along roads). A special case is motion on the ground plane. The mapping of the ground plane to the image plane is a collineation (or projectivity) In homogeneous coordinates, points on the ground plane $(x, y, 1)$ are mapped to points on the image (u, v, s) by a 3x3 matrix, known up to a scale [5]. The coefficients of this matrix can be determined using calibration techniques. The path can be described by a curve (parameterized by arc-length) $p(s) = [x(s), y(s)]^T$, we can formulate the *state* as $S(t) = [s(t), \dot{s}(t), \ddot{s}(t)]^T$ and then dynamics (up to 2nd order) along the path are thus described by the simple linear system used earlier (2nd order integrator).

Thus if the path geometry is known, or available from prior computation or calibration, the dynamic system formulation of motion along that path can still be described by a linear system. The measurement, however, involves a non-linear mapping from the path to the image.

7 Discussion

While we have demonstrated that active control of resolution can yield stable tracking performance, the situations where we have applied it are very simple. Generally some form of calibration must be performed to enable the motion to be described by a linear system. We monitor the size of the covariance and/or the delay time to assess the performance of the tracking system (and determine whether to employ resolution control). Other metrics, such as the Fisher Information matrix are being considered for possible use in assessing tracking performance. Without control of the ROI size (and resolution of processing within the ROI), one cannot guarantee stability of the tracking systems. Previous tracking systems have intuitively followed these bounds. We have tried to quantify the relationships between the system parameters and tracking performance. Future work aims to provide a mechanism to evaluate tracking performance online. Monitoring either the condition number of the information matrix or the sequence of delay times between measurements allows the tracker to perform on-line modification of resolution to maintain stability. Currently the positional covariance size is used to instantiate resolution control. This method works as long as the modified resolution does not effect the target recognition process. We implicitly assumed that the target was still recognizable at the computed resolution. If the resolution employed is "larger" than the target, tracking will fail.

We have presented dynamical systems which can be described with simple linear models. Certain examples, (e.g. motion of vehicles along curvilinear paths) produce motion which is highly non-linear in the image, yet the tracking can be formulated with a linear system. While the linear dynamic system may seem restrictive, the ability to separate path geometry from path dynamics enables tracking of complex motion such as vehicles on known roads and landmark tracking for navigation. Many tracking applications can fit within this framework.

References

1. P. Anandan. A computational framework and an algorithm for measurement of visual motion. *International Journal of Computer Vision*, 2:283–310, 1989.
2. A. Blake, R. Curwen, and A. Zisserman. A framework for spatio-temporal control in the tracking of visual contours. *International Journal of Computer Vision*, 1993.
3. J. Clark and N. Ferrier. Modal control of visual attention. In *Proc. of the Int'l Conf. on Computer Vision*, pages 514–531, Tarpon Springs, Florida, 1988.
4. A. Cretual and F. Chaumette. Image-based visual servoing by integration of dynamic measurements. In *International Conf. on Robotics and Automation*, pages 1994–2001, 1998.
5. O. Faugeras. *Three Dimensional Vision.* MIT Press, 1993.
6. N. Ferrier and J. Clark. The Harvard Binocular Head. *International Journal of Pattern Recognition and AI*, pages 9–32, March 1993.
7. E. Fieller. The distribution of the index in a normal bivariate population. *Biometrika*, 24:428–440, 1932.
8. J. D. Foley and A. van Dam. *Fundamentals of interactive computer graphics.* Addison-Wesley, 1982.
9. R. Geary. The frequency distribution of the quotient of two normal variables. *Royal Statistical Society Series A*, 93:442–446, 1930.
10. A. Gelb and the Technical Staff, Analytic Sciences Corporation. *Applied optimal estimation.* MIT Press, Cambridge, MA, 1974.
11. A. S. Glassner, editor. *Graphics gems (Volumes I-V).* Academic Press, 1990-1995.
12. C. Harris and C. Stennett. Rapid – a video-rate object tracker. In *Proc. 1st British Machine Vision Conference*, pages 73–78, 1990.
13. A. Jazwinski. *Stochastic processes and filtering theory.* Academic Press, 1970.
14. R. Mehra. Optimization of measurement schedules and sensor designs for linear dynam ic systems. *IEEE Transactions on Automatic Control*, 21(1):55–64, 1976.
15. B. Nelson, N. Papanikolopoulos, and P. Khosla. Visual servoing for robotic assembly. In *Visual Servoing-Real-Time Control of Robot Manipulators Based on Visu al Sensory Feedback*, pages 139–164. World Scientific Press, 1993.
16. K. Nichols and S. Hutchinson. Weighting observations: the use of kinematic models in object tracking. In *Proc. 1998 IEEE International Conf. on Robotics and Automation*, 1998.

17. D. Okhotsimksy, A.K.Platonov, I. Belousov, A. Boguslavsky, S. Emelianov, V. Sazonov, and S. Sokolov. Real time hand-eye system: Interaction with moving objects. In *International Conf. on Robotics and Automation*, pages 1683–1688, 1998.
18. C. Olivier. Real-time observatiblity of targets with constrained processing power. *IEEE Transactions on Automatic Control*, 41(5):689–701, 1996.
19. N. Papnikolopoulos, P. Khosla, and T. Kanade. Visual tracking of a moving target by a camera mounted on a robot: A comp bination of control and vision. *IEEE Transactions on Robotics and Automation*, 9(1), 1993.
20. J. Roberts and D. Charnley. Parallel attentive visual tracking. *Engineering Applications of Artificial Intelligence*, 7(2):205–215, 1994.
21. T. Schnackertz and R. Grupen. A control basis for visual servoing tasks. In *Proc. 1995 IEEE Conf. on Robotics and Automation*, Nagoya, Japan, 1995.
22. G. Sullivan. Visual interpretation of known objects in constrained scenes. *Phil. Trans. R. Soc. Lond. B.*, B(337):109–118, 1992.
23. M. Vincze and C. Weiman. On optimising tracking performance for visual servoing. In *International Conf. on Robotics and Automation*, pages 2856–2861, 1997.

A. 1D Analytic Results

We present a 1D example analytically. The following example follows closely from that given in [18]). The one dimensional linear system:

$$dx = axdt + dw \ ; \ E[dw^2] = \sigma^2 dt \tag{11}$$

$$y(t_k) = x(t_k) + v \ ; \ v(t_k) \sim N[0, R_k] \tag{12}$$

For this scalar system equation 6 is

$$P(t_{k+1}^-) = e^{2a(t_{k+1}-t_k)} \frac{P(t_k^-)R_k}{P(t_k^-) + R_k} + \frac{\sigma^2}{2a}\left(e^{2a(t_{k+1}-t_k)} - 1\right) \tag{13}$$

This general form of $P(t_{k+1}^-)$ can be shown to grow for particular dynamics and measurement noise. Note that if

$$t_{k+1} - t_k > \frac{1}{2a}\log\left(\frac{P+R}{R}\right)$$

(where we abbreviate $P = P(t_k^+)$ and $R = R(t_k)$) then

$$\begin{aligned} P(t_{k+1}^-) &> \left(\frac{P+R}{R}\right)\frac{PR}{P+R} + \frac{\sigma^2}{2a}\left(\frac{PR}{R} - 1\right) \\ &= P(t_k^-)\left(1 + \frac{\sigma^2}{2aR}\right) - \frac{\sigma^2}{2a} \\ &= P(t_k^-)\left(1 + \frac{\sigma^2}{4aR}\right) + P(t_k^-)\frac{\sigma^2}{4aR} - \frac{\sigma^2}{2a} \\ &> P(t_k^-)\left(1 + \frac{\sigma^2}{4aR}\right) \quad \text{for } P(t_k^-) > \max[2R, a] \end{aligned}$$

Hence the covariance grows at a rate greater than unity.

If we choose the resolution to be some fraction of the covariance, $R = \alpha P$, where $\alpha << 1$ then equation 13 becomes

$$P(t_{k+1}^-) = e^{2a(t_{k+1}-t_k)}\frac{\alpha}{1+\alpha}P(t_k^-) + \frac{\sigma^2}{2a}\left(e^{2a(t_{k+1}-t_k)} - 1\right)$$

and the covariance evolves as a geometric sequence

$$P(t_{k+1}^-) = \gamma P(t_k^-) + \zeta.$$

For $\gamma < 1$ the covariance decreases and hence the target can be tracked. The expression

$$\gamma < 1 \qquad \text{or} \qquad e^{2a(t_{k+1}-t_k)}\frac{\alpha}{1+\alpha} \ < \ 1$$

makes explicit the trade off between the sample interval $(t_{k+1} - t_k)$, the resolution sub-sampling α, and the system dynamics, a. If the system is sufficiently slow for the time scale used then we can pick a resolution to ensure tracking.