49th IEEE Conference on Decision and Control
December 15-17, 2010
Hilton Atlanta Hotel, Atlanta, GA, USA

Adaptive Control using Quantized Measurements with Application to Vision-only Landing Control

Yoav Sharon, Daniel Liberzon, Yi Ma

Abstract— **We consider a class of control systems where the plant model is unknown and the feedback contains only partial quantized measurements of the state. We use a nonlinear optimization that is taking place over both the model parameters and the state of the plant in order to estimate these quantities. We propose a computationally efficient algorithm for solving the optimization problem, and prove its convergence using tools from convex and non-smooth analysis. We demonstrate the importance of this class of control systems, and our method of solution, using the following application: having a fixed wing airplane follow a desired glide slope on approach to landing. The only feedback is from a camera mounted at the front of the airplane and looking at a runway of unknown dimensions. The quantization is due to the finite resolution of the camera. Using this application we also compare our method to the basic method prevalent in the literature, where the optimization is only taking place over the plant model parameters.**

I. Introduction

A quantizer is a device that converts a real-valued signal into a piecewise constant one taking a finite set of values. In this paper we consider fixed output quantization, for example due to sensors of limited resolution. This type of quantization had been studied, among many others, in [1], [2] and [3]. We remark that many of the recent results on quantization consider dynamic quantization, for example [4] and [5].

When the plant dynamics are unknown and system identification, but not stabilization, is desired, we mention [6], [7], and [8] among others which address the issue of quantization. Surprisingly, very few papers dealt with the problem of stabilization when the plant dynamics are unknown, despite the prevalence of this problem in many control applications. Two papers, [9] and [10], consider input quantization. The assumption taken by these papers, of input quantization with deterministic quantizers, makes a fundamental difference from the settings of output quantization: With the controller knowing the input acting on the plant, a certainty equivalence principle separates the estimation of the plant dynamics from the effects of quantization. In [11] robustness to variations in the plant dynamics is proved using a specific dynamic quantization scheme. As the actual plant dynamics are not estimated, this requires the variation of the plant dynamics from some nominal model to be sufficiently small. Using supervisory control, [12] switches between several controllers, finds the one that best approximates the actual plant dynamics, and uses that one to stabilize the system.

The authors are with the Coordinated Science Laboratory, Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA. Email addresses: {ysharon2,liberzon,yima}@illinois.edu. This work was supported by NSF ECCS-0701676 award.

To generate the stabilizing control input, or to identify system parameters, an estimate of the state needs to be extracted from the quantized measurements. The basic approach is to select the center of the quantization regions as the state estimate. This was the approached followed by all the references cited above, including proofs of convergence for example in [6]. However, as we show in this paper, the convergence may be too slow and a more careful treatment of the quantized measurements, with their special characteristics, needs to be employed in selecting the state estimate. Here we follow-up on the approached proposed in [13] in the context of system identification. The first novelty in our paper is an alternative computational approach for solving the optimization problem that selects the state estimate.

The second novelty is the development of a simulation, on which this new approach can be tested, of a vision-based control problem: controlling a fixed-wing airplane to follow a gliding path on approach to landing. The only available feedback for the landing controller is a camera mounted on the airplane and looking at the runway. The source of quantization in this problem is the pixelization of the image. We believe that the development of a simulation platform for this problem has merits of its own in exposing the issue of quantization in vision-based control, and in the ability to compare different approaches for addressing this issue. As an application, we expect the settings we consider here to be applicable to small unmanned aerial vehicle (UAV) where one may want to avoid installing gyroscopes due to cost and weight considerations. For a complete flight control system though, the addition of at least an airspeed indicator to our settings would be required to measure this critical quantity that cannot be observed by the camera.

For the sake of completeness we cite [14], [15] and [16] as some of the other works which address the problem of landing by vision only. As each of these studies uses different settings and a different set of assumptions, we cannot compare their results to ours. We do not cite other studies where vision is only used for guidance, and the control task is accomplished using inertial sensors (gyroscopes).

Outline: In §II we formulate the general problem we address in this paper and propose an algorithm for solving the problem; in §III we provide a convergence result for the proposed algorithm; starting from §IV we focus on the specific application; in §IV we recall the longitudinal dynamics of an airplane, and derive a reduced order model; in §V we make the connection between the camera input and the state of the airplane; in §VI we describe the controller and in particular the control law; and finally in §VII we present

simulation results. We conclude with final remarks in §VIII.

II. Problem Formulation and Estimation Method

Consider a control system which consists of a plant, a quantizer, and a controller. The plant is assumed to be linear time variant (LTV):

$$\begin{aligned} \dot{x}(t) &= A(t,a)\,x(t) + B(a)\,u(t) + D(a) \\ y(t) &= Cx(t) \qquad\qquad z(t) = E(t,a) \end{aligned} \tag{1}$$

where $x(t) \in \mathbb{R}^n$ is the state of the plant, $u(t) \in \mathbb{R}^m$ is the control input, $y(t) \in \mathbb{R}^p$ ($p \leq n$) is the output signal which is sampled by the quantizer, $z(t) \in \mathbb{R}^q$ is an uncontrollable signal that is sampled by the quantizer and assists in estimating the model parameters, and $a \in \mathbb{R}^l$ is the vector of (the constant) model parameters. The matrices $A(t,a)$, $B(a)$, C, $D(a)$ and $E(t,a)$ are of appropriate dimensions. Their structure, as a function of the model parameters, a, and possibly of the time, is known, but the model parameters themselves are unknown.

The quantizer samples the signals y and z once every τ seconds, and for each individual element, y_i and z_j, sends the controller an interval which contains this element. Thus up to time t, assuming the sampling started at $t = 0$, the information available to the controller is the lower and upper bounds, $\underline{y}_i(k\tau) < \overline{y}_i(k\tau)$, $i = 1,\ldots,p$, $k = 0,\ldots,\lfloor t/\tau \rfloor$ as well as $\underline{z}_j(k\tau) < \overline{z}_j(k\tau)$, $j = 1,\ldots,q$, $k = 0,\ldots,\lfloor t/\tau \rfloor$, such that for each i, j, and k: $\underline{y}_i(k\tau) \leq y_i(k\tau) \leq \overline{y}_i(k\tau)$ and $\underline{z}_i(k\tau) \leq z_i(k\tau) \leq \overline{z}_i(k\tau)$. The controller estimates the state and the model parameters based on this information, and then uses that estimate to generate the control input that will drive the system to some desired steady-state value. The estimation, independent of the law by which the control input is generated, will be discussed in this section. In §VI we will provide an example of a control law.

By restricting to piecewise constant control input such that $\forall k \in \mathbb{N}$: $u(t) = u(k\tau)\ \forall t \in [k\tau, (k+1)\tau)$, the continuous plant dynamics (1) can be written in discrete form as:

$$\begin{aligned} x((k+1)\tau) &= A_k(a)\,x(k\tau) + B_k(a)\,u(k\tau) + D_k(a) \\ y(k\tau) &= Cx(k\tau) \qquad\qquad z(k\tau) = E_k(a). \end{aligned} \tag{2}$$

While it is possible to compute A_k, B_k, D_k and E_k such that (2) tracks (1) exactly, taking $A_k(a) = e^{\int_{k\tau}^{k\tau+\tau} A(s,a)ds}$ for example, in many cases an approximation as the one we use in §VI will be sufficient. Define

$$\begin{aligned} U_{k+1} &\doteq \left[u^T(0), u^T(\tau), \ldots, u^T((k-1)\tau)\right]^T \\ Y_{k+1} &\doteq \left[y^T(0), z^T(0), y^T(\tau), \ldots, y^T(k\tau), z^T(k\tau)\right]^T, \end{aligned}$$

and consider

$$\begin{aligned} \mathscr{E}(k, Y_k, U_k, a) = \mathfrak{A}(k\tau, a) \begin{bmatrix} y((k-r)\tau) \\ \vdots \\ y((k-1)\tau) \end{bmatrix} + \\ \mathfrak{B}(k\tau, a) \begin{bmatrix} u((k-r)\tau) \\ \vdots \\ u((k-1)\tau) \end{bmatrix} + \mathfrak{D}(k\tau, a) - y(k\tau) \end{aligned} \tag{3}$$

where r is the observability index (assumed to be constant) for the system (A_k, C), and $\mathfrak{A}$, $\mathfrak{B}$ and $\mathfrak{D}$ are such that when the values in Y_k and U_k follow the dynamics in (2) we get $\mathscr{E}(k, Y_k, U_k, a) = 0$. The existence of such $\mathfrak{A}$, $\mathfrak{B}$ and $\mathfrak{D}$ is a standard result of discrete-time systems. We refer to $\mathscr{E}(k, Y_k, U_k, a)$ as the modeling error.

We define the cost function,

$$f(Y_N, a) \doteq \sum_{k=r}^{N} \left\|\mathscr{E}(k, Y_N, U_N, a)\right\|_2^2 + \sum_{k=0}^{N-1} \left\|E(k\tau, a) - z(k\tau)\right\|_2^2,$$

and propose the following minimization problem in order to estimate both the state and the model parameters:

$$\begin{aligned} &\min_{a, Y_N} f(Y_N, a) \\ &\text{subject to } \forall k \in [0,\ldots,N-1]: \\ &\qquad \underline{y}_i(k\tau) \leq y_i(k\tau) \leq \overline{y}_i(k\tau) \quad \forall i \in [1,\ldots,p] \\ &\qquad \underline{z}_i(k\tau) \leq z_i(k\tau) \leq \overline{z}_i(k\tau) \quad \forall j \in [1,\ldots,q], \end{aligned} \tag{4}$$

where N is the number of quantized output measurements we collected. Problem (4) is a constrained nonlinear minimization problem. We solve it using the following iterative process: In each iteration fix Y_N and find a that minimizes the cost function; then with the new a, find Y_N that minimizes the cost function and satisfies the constraints. When a is fixed, minimizing over Y_N becomes a constrained quadratic programming problem for which there exist computationally efficient solvers. Minimizing over a when Y_N is fixed can still be a nonlinear minimization problem, but it is now unconstrained, and it has a fixed (small) number of variables that does not grow with N. In many cases, as in the problem we address below, we can derive the second derivative, the Jacobian, explicitly and solve it efficiently using general purpose nonlinear solvers. The algorithm can be initialized with any Y_N that satisfies the constraints. We remark that the convergence of this algorithm is performed online, such that with every iteration more measurements are added.

III. Proof of Convergence

Because $f(Y_N, a)$ is quadratic in Y_N, we can rewrite (4), for fixed N and U_N, as

$$\begin{aligned} &\min_{a, Y} \left\|Q(a)\,Y - r(a)\right\|_2^2 \\ &\text{subject to} \quad \underline{Y}_i \leq Y_i \leq \overline{Y}_i \quad \forall i \in [1,\ldots,n] \end{aligned} \tag{5}$$

where $Y \in \mathbb{R}^n$, $Q : \mathbb{R}^l \to \mathbb{R}^{m \times n}$, $r : \mathbb{R}^l \to \mathbb{R}^m$ (the n and m defined in this section are different from the n and m defined in the previous section). Note that $m < n$. In proving convergence of our proposed iterative algorithm, we will refer to (5) as the problem being minimized. We define $\mathcal{Y}$ to be the set of Y's satisfying the inequality constraints. We say that a is a critical point of f for a given Y if

$$\frac{\partial f(Y, a)}{\partial a_i} = 0, \qquad \forall i \in \{1,\ldots,l\}. \tag{6}$$

We say that Y is a critical point of f for a given a if

$$\frac{\partial f(Y,a)}{\partial Y_i} \leq 0 \text{ if } Y_i > \underline{Y}_i, \quad \frac{\partial f(Y,a)}{\partial Y_i} \geq 0 \text{ if } Y_i < \overline{Y}_i, \quad \forall i \in \{1,\ldots,n\}. \tag{7}$$

And we say that (Y,a) is a critical point of f if both (6) and (7) hold. We define σ as the function that maps each $Y \in \mathcal{Y}$ to the set of critical points of f given Y.

Consider the following assumptions:

1) The functions $Q(\cdot)$ and $r(\cdot)$ are continuous.
2) Let (Y_k, a_k) and (Y_{k+1}, a_{k+1}) be the estimated values before and after iteration k of the algorithm. Then for every k: $a_{k+1} \in \sigma(Y_k)$, $f(Y_k, a_{k+1}) < f(Y_k, a_k)$ if $a_{k+1} \neq a_k$, Y_{k+1} is a critical point of f given a_{k+1}, and $f(Y_{k+1}, a_{k+1}) < f(Y_k, a_{k+1})$ if $Y_{k+1} \neq Y_k$.
3) There exists $K \in \mathbb{N}$ such that the number of critical points of f given Y, $\sigma(Y)$, is smaller than K for every $Y \in \mathcal{Y}$, and furthermore, $\sigma(\cdot)$ is continuous.

We now can state the following convergence result:

Proposition 3.1: Given that assumptions 1-3 hold, the iterative algorithm described above converges to a set M of critical points of f.

The proof is based on the compactness of $\mathcal{Y}$; on the continuity of $\sigma(Y)$, due to assumption 3, which also implies compactness of $\sigma(\mathcal{Y})$; on the fact that $\arg\min_{Y\in\mathcal{Y}} f(a,Y)$ is closed as a function of a due to a result from [17]; and finally on the Discrete-time LaSalle Invariance Principle for set-valued maps, [18, Theorem C.1]. The complete proof, omitted here, can be found in the unabridged version of this paper available online on the first author's website.

We find appropriate to report here an additional result, even though its implication is still under investigation. Set $n_Q = \operatorname{rank} Q$. We say that $Q \in \mathbb{R}^{m\times n}$ is *in general directions* if every n_Q columns of Q are linearly independent. We also define $V(a) \doteq \min_{Y\in\mathcal{Y}} f(Y,a)$, and $\mathcal{P}_0 \doteq \{a \in \mathbb{R}^l \,|\, V(a) = 0\}$ ($\mathcal{P}_0$ might be an empty set).

Proposition 3.2: With the additional assumption that the functions $Q(\cdot)$ and $r(\cdot)$ are C^2 (twice continuously differentiable), the vector of model parameters a in the iterative algorithms described above converges to a set $M \subset \mathbb{R}^l$ for which one of the following holds: (1) $M \subseteq \mathcal{P}_0$; (2) $Q(M)$ contains matrices not in general directions; (3) for every $a \in M$, the directional derivative $V'(a;v) \doteq \lim_{\varepsilon \searrow 0^+} \frac{V(a+\varepsilon v) - V(a)}{\varepsilon}$ exists and satisfies $V'(a;v) \geq 0$, $\forall v \in \mathbb{R}^l \setminus 0$.

This proposition is mainly due to a result from [19]. Due to disturbances and non linearity in the system, and the zero measure of the set of matrices not in general directions in $\mathbb{R}^{m\times n}$, we expect the third case to be the prevailing one.

We now address the assumptions we made. Assumption 1 holds with many models. Most optimization tools satisfy assumption 2 (ignoring numerical errors). For assumption 3 to hold we need that the number of locally minimizing a's be finite for any $Y \in \mathcal{Y}$. This requires Y to be sufficiently exciting in some sense (depending on the specific model and the quantization). We are still investigating what other conditions need to be satisfied in order to guarantee that assumption 3 holds.

IV. Airplane Dynamics

We consider only the longitudinal dynamics in the vertical plane, and we make the following assumptions. There is no difference in elevation between the two ends of the runway. The aircraft has static stability — the control system consisting of only the pitch and the pitch rate, with all other signals considered as external input, is open-loop stable. The unknown wind velocity has only a fixed (independent of height) horizontal component. There is no thrust (power off landing). The lift, drag and gravitational forces, associated with the pitch angle required to follow the desired glide slope at the initial velocity, are at a balance such that the velocity remains relatively steady. And finally, we assume the airplane starts relatively close to the desired glide slope angle. In the last section we will test our method on the dynamics of a Cessna 172.

This section is divided into two subsections. In the first subsection we state the true dynamics of the airplane. In the second subsection we approximate the true dynamics using an LTV model. We emphasize that while we use the LTV model to design the implementation of our method, we use the true dynamics to test it in simulation.

A. True Dynamics

By considering only longitudinal dynamics in the vertical plane, we are left with six degrees of freedom describing the motion of the airplane[1]:

$$\left.\begin{matrix} p_x \\ p_z \end{matrix}\right\} \text{— position} \qquad \left.\begin{matrix} v_x \\ v_z \end{matrix}\right\} \text{— velocity}$$
$$\left.\begin{matrix} \theta \\ \dot{\theta} \end{matrix}\right\} \text{— pitch angle and pitch rate.} \tag{8}$$

All the quantities above are defined in the frame of reference whose origin is fixed at the beginning of the runway. The x-axis positive direction is defined such that the runway is on the positive side of this axis. The z-axis positive direction is up. The control input is the elevator deflection, δ_e, which measures the angular displacement of the elevator from its trim position.

Using the six quantities in (8) and several aerodynamics constants we can derive the equations of motion. These equations can be found in any textbook on flight dynamics, [20] for example. We will use the following rotation matrix in deriving the equations: $R(\phi) \doteq \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix}$. A standard way of computing the forces acting on the airplane is to first compute the lift and the drag. The lift is the aerodynamic force perpendicular to the relative wind, and the drag is the aerodynamic force parallel to the relative wind. Both forces are assumed to act on the center of lift.

The complete derivation of the equations of motion is as follows:

1) Angle of relative wind, $\varphi = \tan^{-1}(v_z/(v_x + v_{\text{wind}}))$.

[1]The standard state used in the literature on automatic landing includes: true airspeed v_T, angle of attack α, pitch θ, pitch rate $q = \dot{\theta}$, height $h = p_z$, and deviation from the glide slope d [20, p.341]. Knowing the wind one can translate between this standard state and (8).

2) Angle of attack, $\alpha = \theta - \varphi$.
3) Airspeed, $V_T = \left\| [v_x + v_{\text{wind}}, v_z]^T \right\|$.
4) Lift coefficient, $C_L = C_{L_\alpha}(\alpha) + C_{L_q}\dot{\theta}\bar{c}/(2V_T) + C_{L_{\delta_e}}\delta_e$.
5) Drag coefficient, $C_D = C_{D_\alpha}(\alpha) + \left|C_{D_{\delta_e}}\delta_e\right|$.
6) Pitch moment coefficient, $C_m = C_{m_\alpha}(\alpha) + C_{m_{\dot{\alpha}}}\dot{\alpha}\bar{c}/(2V_T) + C_{m_q}\dot{\theta}\bar{c}/(2V_T) + C_{m_{\delta_e}}\delta_e$.

Remark: Above C_{L_q}, $C_{L_{\delta_e}}$, $C_{D_{\delta_e}}$, $C_{m_{\dot{\alpha}}}$, C_{m_q} and $C_{m_{\delta_e}}$ are airframe dependent empirically obtained constants; C_{L_α}, C_{D_α}, C_{m_α} are airframe dependent functions of the angle of attack, and are derived from tables of empirically obtained values. Below ρ is the air density which we assume to be constant, and S is the wing area.

7) Lift, $L = 0.5C_L\rho V_T^2 S$.
8) Drag, $D = 0.5C_D\rho V_T^2 S$.
9) Linear aerodynamic forces, $F = R(\varphi)[-D, L]^T$
10) Pitch moment, $M = 0.5C_m\rho V_T^2 S\bar{c} + (R(\theta)D_{cg}) \times_y F$ where $[x_1, z_1]^T \times_y [x_2, z_2]^T \doteq z_1 x_2 - x_1 z_2$.
11) Linear accelerations, $\dot{v}_x = F_x/W$, $\dot{v}_z = F_z/W - g$, where W is the weight of the aircraft, and g is the acceleration due to gravity.
12) Angular acceleration, $\ddot{\theta} = M/I_{yy}$ where I_{yy} is the moment of inertia around the pitch axis.
13) Linear velocity, $\dot{p}_x = v_x$, $\dot{p}_z = v_z$.

B. Reduced Order Model

We define the reduced order state: $x \doteq \left(\theta_p, \theta_v, \theta, \dot{\theta}\right)^T$, where $\theta_p \doteq \tan^{-1}(p_z/p_x) - \gamma_R$, $\theta_v \doteq \tan^{-1}(v_z/v_x) - \gamma_R$, and γ_R is the (negative) desired glide slope angle. In the literature on airplane dynamics, $\gamma_R + \theta_v$ is referred to as the flight path angle, γ. The motivation for this choice of state is that our main goal is to drive θ_p to zero. The dynamics of θ_p depend strongly on all the signals in x, thus we also need to drive these signals to some appropriate values. As the dynamics of θ_p depend on the remaining signals from the full order model, $d = \sqrt{p_x^2 + p_y^2}$ and $V_E = \sqrt{v_x^2 + v_z^2}$, only through multiplications with the states already in x, we exclude explicit reference to these states in the system model that we attempt to control. Note that by our convention,

$$\begin{pmatrix} p_x \\ p_z \end{pmatrix} = d\begin{pmatrix} -\cos(\theta_p + \gamma_R) \\ -\sin(\theta_p + \gamma_R) \end{pmatrix}, \quad \begin{pmatrix} v_x \\ v_z \end{pmatrix} = V_E\begin{pmatrix} \cos(\theta_v + \gamma_R) \\ \sin(\theta_v + \gamma_R) \end{pmatrix}.$$

We now rewrite the dynamics for this reduced order state. We start with the θ_p dynamics:

$$\begin{aligned}\dot{\theta}_p &= \frac{1}{1 + (p_z/p_x)^2}\frac{v_z p_x - v_x p_z}{p_x^2} \\ &= \frac{V_E}{d}\left(\sin(\theta_p)\cos(\theta_v) - \sin(\theta_v)\cos(\theta_p)\right)\end{aligned}$$

If we assume that throughout the approach maneuver, θ_v and θ_p remain relatively small and V_E stays relatively fixed, then we can approximate the θ_p dynamics with

$$\dot{\theta}_p \approx \frac{1}{t_f - t}\theta_p - \frac{1}{t_f - t}\theta_v \tag{9}$$

where t_f is the time we expect to reach the beginning of the runway ($d/V_E = t_f - t$).

We continue with the θ_v dynamics. We approximate the factors which depend on the angle of attack, α, in the lift and drag coefficients with linear functions, and neglect the remaining factors, due to their relatively small contribution. This results in

$$C_L \approx C_{L,1}\alpha + C_{L,0} \qquad C_D \approx C_{D,1}\alpha + C_{D,0}.$$

With that we have (using $\phi \doteq \theta_v + \gamma_R - \varphi$),

$$\begin{aligned}\dot{\theta}_v =& \frac{1}{V_E}\left[-\sin(\theta_v + \gamma_R), \cos(\theta_v + \gamma_R)\right] \times \\ & \left(R(\varphi)\frac{\rho V_T^2 S}{2W}\begin{bmatrix} -C_D \\ C_L \end{bmatrix} - \begin{bmatrix} 0 \\ g \end{bmatrix}\right) \\ \approx& \frac{\rho V_T^2 S}{2V_E W}\left(-\sin(\phi)C_{D,1} + \cos(\phi)C_{L,1}\right)(\theta - \varphi) + \\ & \frac{\rho V_T^2 S}{2V_E W}\left(-\sin(\phi)C_{D,0} + \cos(\phi)C_{L,0}\right) - \frac{g\cos(\gamma_R)}{V_E},\end{aligned}$$

where we also used the approximation $\cos(\theta_v + \gamma_R) \approx \cos(\gamma_R)$ assuming θ_v is relatively small. In the windless case, $\phi = 0$ as $\varphi = \theta_v + \gamma_R$, and it is easy to see how a linear model can be derived:

$$\dot{\theta}_v \approx -C_{v\to v}\theta_v + C_{p\to v}(\theta - \theta_0) \tag{10}$$

where $C_{v\to v}$, $C_{p\to v}$ and θ_0 are considered constants. They do depend on V_T as well as on other environmental variables such as the air pressure and the weight of the aircraft, but we assume that all of these variables (including, in particular, V_T) change only slightly throughout the approach maneuver. We claim that the linear model (10) is a good approximation even when there is a wind, where now the three constants just mentioned also depend on the wind speed.

We finish with the angular acceleration, $\ddot{\theta}$, dynamics. We see that M is the sum of two terms, one which depends on the pitch moment coefficient, and one due to the linear aerodynamic forces. Since we found that the second term is small compared to the first, we approximate the angular acceleration dynamics without it. If we further approximate $C_{m/\alpha}(\alpha) \approx C_{m,1}\alpha + C_{m,0}$, then its not hard to see that in the windless case the angular acceleration can be written as:

$$\ddot{\theta} \approx C_{v\to\dot{\theta}}\theta_v + C_{\theta\to\dot{\theta}}\theta + C_{\dot{\theta}\to\dot{\theta}}\dot{\theta} + C_{\delta_e\to\dot{\theta}}(\delta_e - \delta_0). \tag{11}$$

And again, we claim that the linear model (11), with different constants, is a good approximation even when there is wind.

V. Camera Feedback

We assume a runway recognition algorithm provides the information about the rows corresponding to different points on the runway, and the width of the runway at these points. All the information is given in pixels, but knowing the parameters of the camera and the angle at which it is installed on the aircraft, we can easily translate the rows into angles in the vertical plane from any reference axis fixed to the aircraft. The reference axis we use is the longitudinal axis, which is also used to define the pitch angle as the angle between this

axis and the plane tangent to Earth's surface. Points below the reference axis will be associated with a negative angle. Summarizing, we assume we have the following information (all the widths are given in pixel units):

$\underline{\phi}_b$	$\overline{\phi}_b$	—	the angle in the vertical plane at which the runway begins
$\underline{w}_b$	$\overline{w}_b$	—	the width of the runway where it begins
$\underline{w}_{b'}$	$\overline{w}_{b'}$	—	the width of the runway in the first row of pixels above ϕ_b
$\underline{\phi}_e$	$\overline{\phi}_e$	—	the angle in the vertical plane to an arbitrary point on the runway
$\underline{w}_e$	$\overline{w}_e$	—	the width of the runway at ϕ_e

Each quantity comes with a lower and upper bound, denoted by the underline and the overline respectively, which is the result of the pixelization.

We now discuss the physical quantities we can derive from these measured quantities. First, the angle at which the runway begins, ϕ_b, relates to our state variables as

$$\underline{\phi}_b \leq \gamma_R + \theta_p - \theta \leq \overline{\phi}_b. \tag{12}$$

Second, the width in pixels of the runway where it begins, w_b, relates to the distance to the runway as $w_b = \mu/d$, where

$$\mu \doteq \frac{\text{runway width (meters)}}{\tan\left(\frac{\text{horizontal field of view (degrees)}}{2}\right)} \left(\text{number of pixels on the horizontal axis}\right)$$

Last, a pixel at a vertical angle ϕ corresponds to a point on the surface which is at an angle of $-\phi-\theta$ below the horizon from the aircraft point of view. The distance to that point, assuming a planar terrain, is $\frac{\sin(-\phi_0-\theta)d_0}{\sin(-\phi-\theta)}$ where d_0 is the distance to another point on the surface which appears at an angle ϕ_0. We just showed that the distance to any object on the surface is inverse proportional to the width in pixels of that object. Thus we have that

$$\frac{\underline{w}_{b'}}{\overline{w}_e} \leq \frac{\sin\left(-\overline{\phi}_b - \theta\right)}{\sin\left(-\underline{\phi}_e - \theta\right)} \leq \frac{\overline{w}_{b'}}{\underline{w}_e} \tag{13}$$

from which we can derive bounds for possible values of θ. There is not a closed form solution to derive these bounds, but they can be easily calculated using simple iterative methods.

Although w_b is not controllable, estimating it helps us to estimate the time we expect to reach the runway, since

$$w_b = \frac{\mu}{V_E\left(t_f - t\right)} \doteq \frac{C_w}{t_f - t}. \tag{14}$$

VI. Controller

In (9), (10), (11), (14) we have established that (1) is applicable to our system, where $u\left(t\right) = \delta_e\left(t\right)$, $z\left(t\right) = w_b\left(t\right)$, $a \doteq \left[t_f, C_{v\to v}, C_{v\to\dot{\theta}}, C_{\theta\to v}, \theta_0, C_{\theta\to\dot{\theta}}, C_{\dot{\theta}\to\dot{\theta}}, C_{\delta_e\to\dot{\theta}}, \delta_0, C_w\right]$. Defining $x'\left(k\right) = x\left(k\tau\right)$, $u'\left(k\right) = u\left(k\tau\right)$, we approximate the continuous dynamics with the following discrete version: $x'\left(k+1\right) = x'\left(k\right) + \tau\left(A\left(k\tau\right)x'\left(k\right) + Bu'\left(k\right) + D\right)$. We then use the algorithm from §II to estimate the state and the model parameters as the system evolves.

Since we estimate a discrete linear model, and we have constraints on the control input, we chose to use Model Predictive Control (MPC) [21]. In order to use the MPC in the standard settings, we transform our LTV system to an LTI (linear time invariant) system by using the following state variables:

$$\tilde{x}_k\left(k\right) = \begin{bmatrix} \left(t_f - \left(k-1\right)\tau\right)x_1\left(k\right) \\ x_3\left(k\right) - p_0 \\ \left(t_f - \left(k-2\right)\tau\right)x_1\left(k-1\right) \\ x_3\left(k-1\right) - p_0 \end{bmatrix}.$$

The transformed state variable follows $\tilde{x}_{k+1} = \tilde{A}\tilde{x}_k + \tilde{B}\left(u - \tilde{e}_0\right)\ \forall k$ for some constant matrices $\tilde{A}$ and $\tilde{B}$, where $\tilde{e}_0 = e_0 - C_{\theta\to\dot{\theta}}p_0/C_{\delta_e\to\dot{\theta}}$. The control horizon of the MPC is denoted by H. In order to satisfy conditions A1–A4 in [21, §3.3], which ensure closed-loop asymptotic stability in the non-adaptive case, we use the terminal constraint $\tilde{x}_{k+H} = 0$ and $u_{k+H} = \tilde{e}_0$. We use a quadratic cost where we only penalize the change in the control action: $\sum_{i=k+1}^{k+H}\left(u\left(i\right) - u\left(i-1\right)\right)^2$.

VII. Simulation Results

We used a Cessna 172 model for our simulation. The numerical values were taken from [22] that lists several dynamical models to be used with the FlightGear Flight Simulator [23]. We positioned the airplane 250 m from the runway on the desired 1 to 9 ratio glide slope. The initial velocity was set to 36 m/s (∼70 knots) true air speed, moving parallel to the ground, no flaps configuration. The pitch and pitch rate were initialized to zero. Wind was set to 5 m/s headwind, ISA atmospheric conditions. The airplane weight was set to 2405 lb. The camera's field of view was 32° (horizontal) by 24° (vertical), its resolution was 640 by 480, and it took 50 frames per seconds (fps). The width of the runway was 20 m and ϕ_e was associated with a point on the runway that was distanced 200 m from the beginning of the runway. The simulation time constant (the time interval between each update of the dynamics) was 0.01 s. We started the simulation using an open-loop control consisting of several step functions. After 2 seconds we started running the estimator, and for every new measurement received we ran two iterations of the estimator. After 3 seconds we closed the loop by engaging the model predictive controller. The control input was limited to ±12°. The control horizon was set to $H = 150$ (or 3 s). Once the field of view was too small to cover the whole width of the runway where it begins, the controller was disengaged and the elevator deflection remained constant. We did not simulate ground effect.

We show here two runs of our simulation. In the first run we used the approximated dynamics which we derived in §IV-B. The reason is that this way we know exactly to which model parameters the estimator should converge. In the second run we used the true airplanes dynamics from §IV-A. In the first run, in addition to the quantized estimator from §II, we also used a basic estimator for comparison. The basic estimator does not take into account the special characteristics of quantized measurements, and attempts to fit a model to measurements which are the center of each quantization range. Essentially the basic estimator minimizes

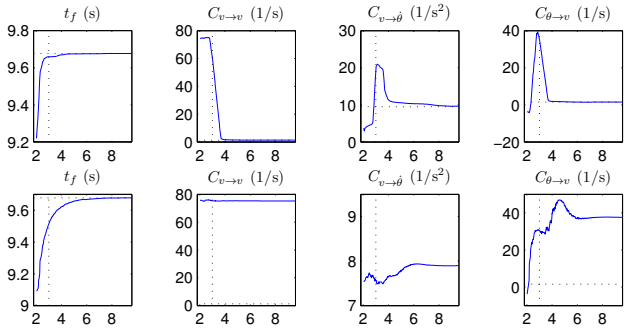

Fig. 1. Comparison between the quantized estimator and the basic estimator using simulated linearized airplane dynamics. The first row of figures shows some of the estimates of the quantized estimator. The second row of figures shows the same estimates of the basic estimator. The x-axis in all the figures represents time (seconds). The horizontal dotted black lines in both sets of figures represent the true model parameters. Note the estimation error of the basic estimator compared to the quantized estimator especially in the estimation of $C_{v\to v}$ and of $C_{\theta\to v}$. The vertical dotted black lines in all the figures represent the time when the model predictive controller was engaged.

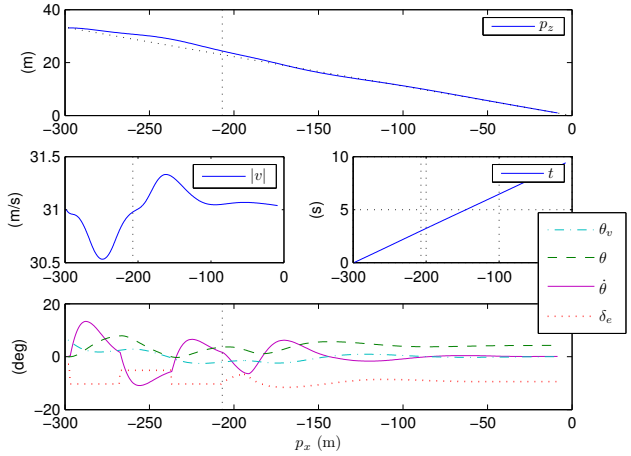

Fig. 2. Simulation of true airplane dynamics using the quantized estimator. The 4 figures show the 6 degrees of freedom state of the airplane and the control input. The slanted dotted black line in the top figure indicates the desired glide slope.

the same cost function from (4) but only over the model parameters.

The results are shown in Figures 1 and 2. The results show significant improvement in the estimation error between the basic estimator and the quantized estimator. They also show that with the quantized estimator we were able to stabilize the system. In an additional run, not shown here, we used the basic estimator with which the system failed to stabilize.

VIII. Conclusion

We formulated an adaptive control problem using quantized measurements. We used an engineering application to establish the importance of this formulation, and demonstrated the benefits that can be gained by a more careful selection of the measurements within the quantization ranges. We acknowledge that the results here are preliminary and we hope that this work will lead to more research on this issue. The most important step would be to provide proof of stability for the closed loop control system, as well as to show that it is robust to small variations to the dynamics from the linear model.

References

[1] R. K. Miller, M. S. Mousa, and A. N. Michel, "Quantization and overflow effects in digital implementations of linear dynamic controllers," *IEEE Trans. Automat. Control*, vol. 33, pp. 698–704, 1988.

[2] D. F. Delchamps, "Stabilizing a linear system with quantized state feedback," *IEEE Trans. Automat. Control*, vol. 35, no. 8, pp. 916–924, 1990.

[3] N. Elia and S. K. Mitter, "Stabilization of linear systems with limited information," *IEEE Trans. Automat. Control*, vol. 46, no. 9, pp. 1384–1400, 2001.

[4] G. N. Nair, R. J. Evans, I. M. Y. Mareels, and W. Moran, "Topological feedback entropy and nonlinear stabilization," *IEEE Trans. Automat. Control*, vol. 49, no. 9, pp. 1585–1597, 2004.

[5] S. Tatikonda and S. Mitter, "Control under communication constraints," *IEEE Trans. Automat. Control*, vol. 49, no. 7, pp. 1056–1068, 2004.

[6] H. Suzuki and T. Sugie, "System identification based on quantized i/o data corrupted with noises and its performance improvement," in *Proc. 45th IEEE Conf. on Decision and Control*, 2006, pp. 3684–3689.

[7] H. Ishii and T. Başar, "Quantization in $\mathcal{H}^\infty$ parameter identification," *IEEE Trans. Automat. Control*, vol. 53, no. 9, pp. 2186–2192, 2008.

[8] K. Tsumura, "Optimal quantization of signals for system identification," *preprint, http://arxiv.org/abs/0905.1862v1*, 2009.

[9] T. Hayakawa, H. Ishii, and K. Tsumura, "Adaptive quantized control for nonlinear uncertain systems," *Systems Control Lett.*, vol. 58, pp. 625–632, 2009.

[10] H. Sun, N. Hovakimyan, and T. Başar, "$\mathcal{L}_1$ adaptive controller for systems with input quantization," in *Proc. 2010 American Control Conference*, 2010, pp. 253–258.

[11] Y. Sharon and D. Liberzon, "Input to state stabilizing controller for systems with coarse quantization," *IEEE Trans. Automat. Control*, 2010, to appear. [Online]. Available: http://www.ysharon.info/papers/quantization.pdf

[12] L. Vu and D. Liberzon, "Stabilizing uncertain systems with dynamic quantization," in *Proc. 47th IEEE Conf. on Decision and Control*, 2008, pp. 4681–4686.

[13] A. Okao, M. Ikeda, and R. Takahashi, "System identification for nano control: A finite wordlength problem," in *Proc. 2003 IEEE Conf. on Control Applications*, 2003, pp. 49–53.

[14] O. Bourquardez and F. Chaumette, "Visual servoing of an airplane for alignment with respect to a runway," in *Proc. 2007 IEEE Int'l Conf. on Robotics and Automation*, 2007, pp. 1330–1335.

[15] A. Miller, M. Shah, and D. Harper, "Landing a UAV on a runway using image registration," in *Proc. IEEE International Conference on Robotics and Automation*, 2008.

[16] A. A. Proctor and E. N. Johnson, "Vision-only approach and landing," in *Proc. AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2005.

[17] M. J. Best and B. Ding, "On the continuity of the minimum in parametric quadratic programs," *J. Optimization Theory and Applications*, vol. 86, no. 1, pp. 245–250, 1995.

[18] J. Cortés, S. Martínez, and F. Bullo, "Spatially-distributed coverage optimization and control with limited-range interactions," *ESAIM: Control, Optimization and Calculus of Variation*, vol. 11, pp. 691–719, 2005.

[19] D. Ralph and S. Dempe, "Directional derivatives of the solutions of a parametric nonlinear program," *Mathematical Programming*, vol. 70, pp. 159–172, 1995.

[20] B. L. Stevens and F. L. Lewis, *Aircraft control and simulation, 2nd edition*. Hoboken, NJ: John Wiley & Sons, Inc., 2003.

[21] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained model predictive control: stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[22] M. S. Selig, R. Deters, and G. Dimock. (2002) Aircraft dynamics models for use with flightgear. University of Illinois at Urbana-Champaign. [Online]. Available: http://www.ae.illinois.edu/m-selig/apasim/Aircraft-uiuc.html

[23] Flightgear, an open-source flight simulator. FlightGear.org. [Online]. Available: http://www.flightgear.org