On a Control Lyapunov Function based Anytime Algorithm for Control of Nonlinear Processes

Vijay Gupta[1] Daniel E. Quevedo[2]

[1]Department of Electrical Engineering, University of Notre Dame, Notre Dame, USA
[2]School of Electrical Engineering & Computer Science, The University of Newcastle, Australia

1. Introduction

- In networked and embedded systems, the computation resources available for calculating the control input may be time-varying.
- Therefore, the implicit assumption often made about the processor always being able to execute the desired control algorithm may break down.
- We study control in the presence of random availability of processing power.
- We present an anytime algorithm, which provides a more refined control input as more processing resources become available.
- The basic idea is to utilize the extra processing time to refine the control input to decrease the Lyapunov function as compared to the value at as many time steps in the past as possible.
- Thus, the effect of not being able to compute an input at a previous time step can be mitigated.
- We analyse stochastic stability of the closed loop system and indicate through numerical simulations that the performance gains provided by the proposed algorithm can be significant.

2. System Setup

Plant Model

- We consider a nonlinear plant model with state $x_k \in \mathbb{R}^n$ and input $u_k \in \mathbb{R}^p$:

$$x_{k+1} = f(x_k, u_k), \quad f(0,0) = 0.$$

- There exists a stabilising state-feedback control policy κ with Lyapunov function $V(x) \geq |x|$, which gives

$$V(x_{k+1}) < \epsilon V(x_k), \quad \text{for some} \quad \epsilon \in (0,1),$$

when

$$x_{k+1} = f(x_k, \kappa(x_k)).$$

- With zero input, we have

$$V(f(x,0)) \leq \alpha V(x), \qquad \forall x \in \mathbb{R}^n, \qquad (1)$$

for some $\alpha > 0$.
- We suppose that the initial state x_0 is such that

$$V(x_0) < \infty.$$

Random Processor Time Availability

- The amount of processor time available for control, say τ_k, is time-varying and random.
- To encompass situations where the controller task can be preempted by other computational tasks, we will assume that the controller
  1. does not know the distribution of $\{\tau_k\}$,
  2. has no advance knowledge of the value of τ_k.

3. Baseline Algorithm

- We denote the probability that the controller is unable to calculate any control input via p_0.
- When using the baseline algorithm, we have:

$$u_k = \begin{cases} \kappa(x_k) & \text{with probability } 1 - p_0, \\ 0 & \text{with probability } p_0. \end{cases}$$

- The resulting system is akin to a networked control system with random packet loss.

Stability via the Baseline Algorithm

Consider the baseline algorithm and α as in (1).
We then have that, if

$$\alpha\, p_0 < 1 - \epsilon(1 - p_0),$$

then:

$$\sum_{k=0}^{\infty} \mathcal{E}\{|x_k| \mid x_0\} < \infty.$$

4. Anytime Control Algorithm

Basic Idea

- The control input u_k aims at decreasing $V(x_{k+1})$.
- At some instances, the execution time may be insufficient giving $u_k = 0$ and $V(x_{k+1}) > V(x_k)$.
- The algorithm aims at countering the effect of such increases whenever possible.

Algorithm Description

At every time k and given x_k:

Step 1. Set $N_k = 0$ and $u_k = 0$.
Step 2. Calculate u_k such that

$$V(f(x_k, u_k)) < \epsilon V(x_{k+1-N_k}).$$

Step 3. Set

$$N_k \leftarrow N_k + 1$$

and go to Step 2.

Remarks

- Whenever the algorithm is interrupted, the current value u_k is used as the control input.
- The algorithm is anytime in the sense that a value for u_k is available at any time. As more execution time becomes available, the quality of u_k is refined.
- The algorithm does not require knowledge of the distribution of the processor time availability.
- The algorithm seeks to calculate a control input which reduces the Lyapunov function over several past time steps. This requires the existence of a common control Lyapunov function going back N_k time steps. This requirement may impose an upper bound on the possible values of N_k.

5. Stability via the Anytime Algorithm

We denote the time instants where $N_k \geq 1$ by

$$\{k_i\} \subseteq \{0, 1, 2, \dots\}, \quad k_{i+1} > k_i$$

and assume that $k_0 = 0$.

Assumptions

A1. The process $\{N_k\}$ is independent and identically distributed with:

$$\text{Prob}\{N_k = \eta\} = p_\eta, \quad \eta \in \{0, 1, 2, \dots\}. \qquad (2)$$

A2. The bound α in (1) satisfies

$$\alpha < 1/p_0.$$

Lemma (Conditional Expectation of $V(x_{k_1+1})$):

Consider the anytime algorithm and suppose that Assumptions A1 and A2 are satisfied.
We then have:

$$\mathcal{E}\{V(x_{k_1+1}) \mid x(k_1)\} = \Omega V(x_1),$$

where

$$\Omega \triangleq \epsilon(1 - p_0)\left(1 + \frac{\alpha}{1 - p_0\alpha} \sum_{\eta=1}^{\infty} p_\eta p_0^{\eta - 1}\right) \geq 0.$$

Proof (Main Points):
1. The variable k_1 satisfies $\text{Prob}\{k_1 = \ell\} = (1 - p_0)p_0^\ell$.
2. Use conditioning upon k_1 and N_k. □

Theorem (Stochastic Stability):

Consider the anytime algorithm. Suppose that Assumptions A1 and A2 are satisfied, and that $\Omega < 1$. Then

$$\sum_{k=0}^{\infty} \mathcal{E}\{|x_k| \mid x_0\} < \infty$$

Proof (Main Points):
1. $V(x_{k_i+1})$ is a stochastic Lyapunov function for $\{x_{k_i}\}$.
2. We have $\mathcal{E}\{|x_{k_i+1}| \mid x_0\} \leq \mathcal{E}\{V(x_{k_i+1}) \mid x_0\} \leq \Omega^i V(x_0)$.
3. For the intermediate instants, $\mathcal{E}\{|x_k|\}$ can be bounded by conditioning upon $\Delta_i \triangleq k_{i+1} - k_i$. □

6. Numerical Examples

- We assume that the processor time available for control calculations is uniformly distributed in the interval $[0, 1]$
- To evaluate performance, we consider

$$\mathcal{E}\left\{\sum_{k=0}^{99} 10x_k^2 + u_k^2\right\}$$

by carrying out 500 Monte-Carlo simulations.

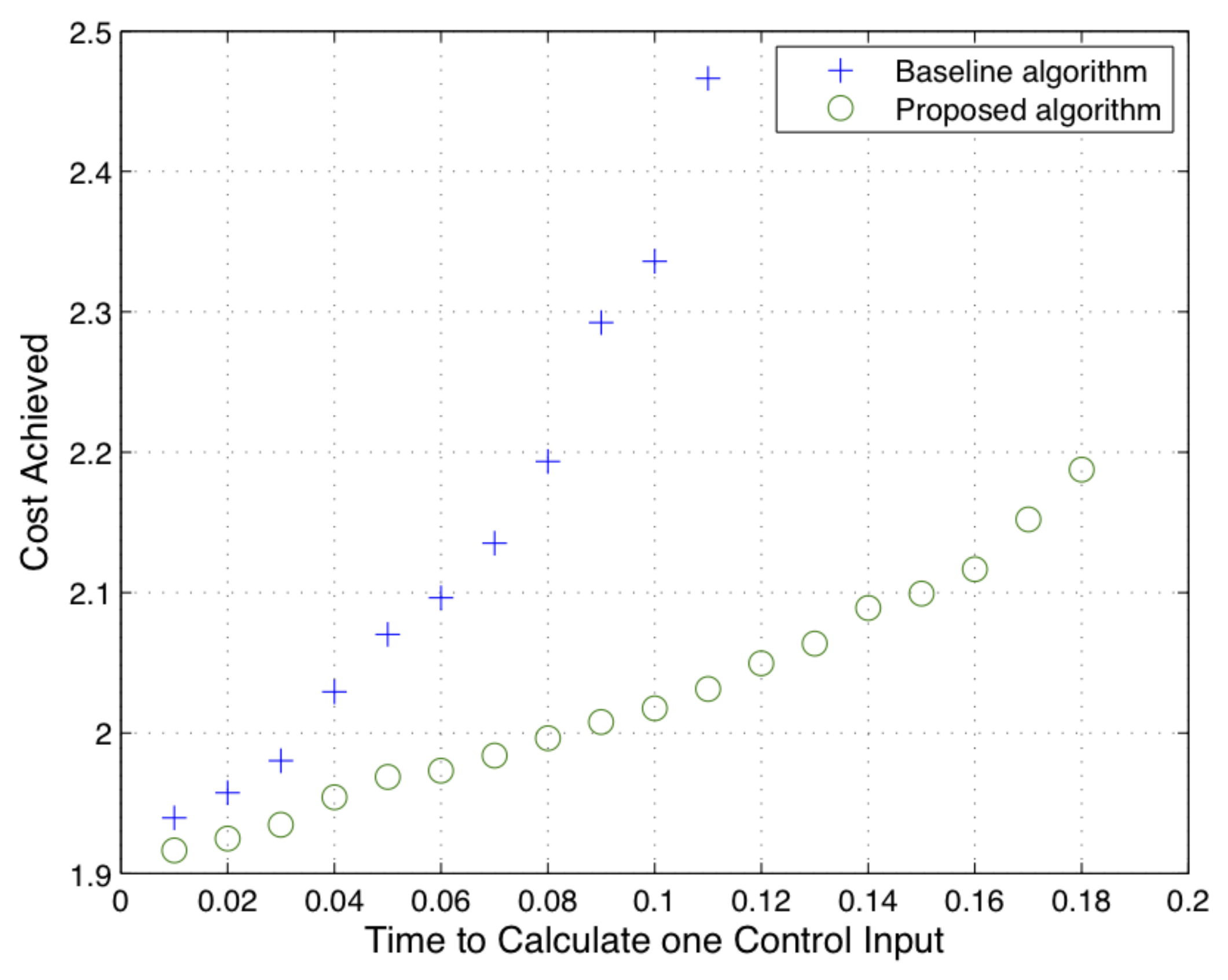
Nonlinear System

We first consider the following process:

$$x_{k+1} = x_k + 0.2(x_k^3 + u_k),$$

with associated control law and Lyapunov function

$$\kappa(x) = -x^3 - x, \quad V(x_k) = |x_k|.$$



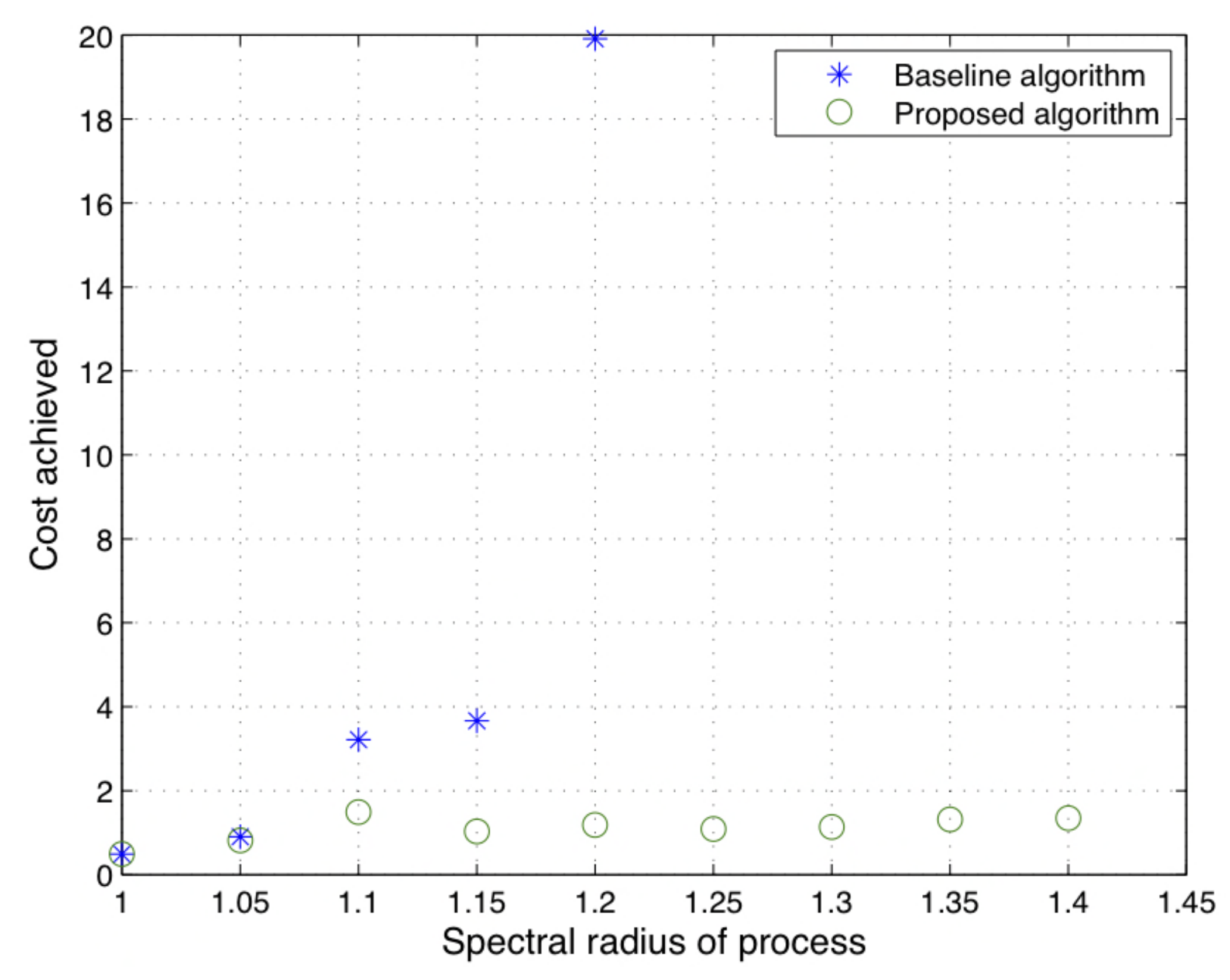
The figure indicates that the proposed algorithm may gives significant performance gains.

Linear System

We next consider the linear process

$$x_{k+1} = \alpha x_k + u_k$$

and assume that the time to calculate one control input (i.e., of carrying out Step 2) is equal to 0.2.



As the plant becomes more unstable, the proposed algorithm gives better performance compared to the baseline algorithm.

7. Conclusions

- We proposed an anytime control algorithm for situations where process resources are random.
- Stability of the resulting closed loop system can be analysed using stochastic Lyapunov functions.
- Simple numerical examples illustrated the performance gain with the proposed algorithm.
- This is but a first step towards a more complete theory of anytime control algorithms.
- In particular, further work may include:
  - considering the effect of model imperfections,
  - carrying out a joint design of the anytime algorithm and a processor-scheduler.