Stability Analysis of Simplex Architecture Controlled Inverted Pendulum

Taylor Johnson
Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801
Email: johnso99@illinois.edu

***Abstract*—Switched controllers are being used frequently for control of complex systems. However, they introduce new challenges for verification of stability, of which a great deal of work in the hybrid systems domain has been formalizing recently. This work is a stability case study of the classical inverted pendulum, in this case controlled using the Simplex architecture of [1], combining to form a complete system as in [2]. The main result shown uses small-gain theorems to prove the stability of the system with regards to measurement delays.**

I. Introduction and Preliminaries

The physical system in Figure 1 consists of a DC-motor driven cart and a pendulum attached to the cart, with the control goal of keeping the angle θ of the pendulum at 0° measured from the vertical. There is an additional control goal of moving to a set point x_s. While in general this system is described by the nonlinear form

$$\dot{x} = f(x, u) \tag{1}$$

we work with a linearized model of the form

$$\dot{x} = Ax + Bu \tag{2}$$

The linearization is justified after the system description. There are four state variables, the cart position x, cart velocity $\dot{x}$, pendulum angle θ, and pendulum velocity $\dot{\theta}$ (we will now denote X as the state vector and x as the position, seen together in Equation 3). As the system has been implemented, it is subject to physical constraints. The range of x is between $[-0.7, 0.7]$ meters, $\dot{x}$ is between $[-1.0, 1.0]$ meters/second, θ is between $[-30, 30]^\circ$, and $\dot{\theta}$ is unconstrained.

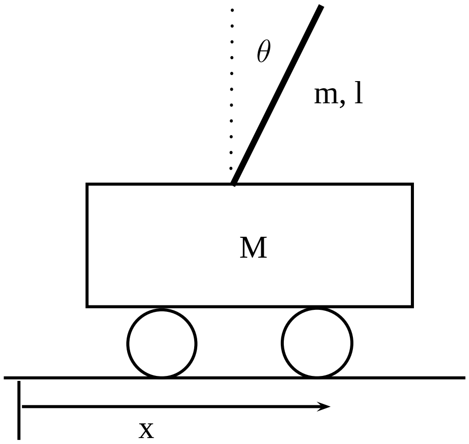

Fig. 1. Inverted Pendulum System

$$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} x - x_s \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} \tag{3}$$

The system is stabilized by linear state feedback of the form $\dot{X} = (A + BK)X$. The control input, u is the armature voltage of the DC-motor (V_a) and is constrained between $[-4.96, 4.96]$ volts.

The primary linearizations to note are that we ignore static friction and take the armature inductance ($L_a = 18$ millihenries) to be 0 henry hence reducing the order of the system by making the armature current state variable I_a a function of V_a. The linearization is justified since the control objective is to stabilize the system in a neighborhood of the vertical equilibrium.

For the remainder of the paper, we work towards proving stability of the system, first in the classical

Lyapunov sense, and then using newer small-gain theorem results to also discuss stability in the delayed system as in [3] and [4]. Recent results using small-gain theorems allow us to prove stability of systems with delay by analyzing the stability of a simpler system without delay. The idea is to treat the error introduced by the delay as a disturbance input with bounded-gain, allowing us to talk about input-to-state stability. Using this, we can find the maximum delay the system can tolerate. Despite the linearizations of the system model, the small-gain techniques apply to nonlinear systems and can also be used to analyze stability in the more general system model.

II. System and Controller Models

Figure 2 shows a high-level view of the entire control system. We now discuss each block individually. We are using hybrid input-output automata to describe the blocks, and we assume the reader is familiar with such notation, but if not, [5] or [6] can serve as references to the notation.

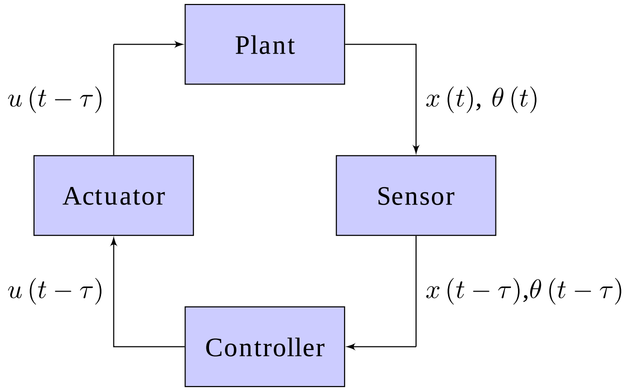
Plant
$u\left(t-\tau\right)$
$x\left(t\right)$, $\theta\left(t\right)$
Actuator
Sensor
$u\left(t-\tau\right)$
$x\left(t-\tau\right)$,$\theta\left(t-\tau\right)$
Controller

Fig. 2. System Model

A. Plant Model

The plant model of the inverted pendulum is described in Figure 3, where the two input parameters to the automaton plant, $A : \texttt{Real}^{4\times 4}$ and $B : \texttt{Real}^{4\times 1}$, are defined as

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -a_{22} & -a_{23} & a_{24} \\ 0 & 0 & 0 & 1 \\ 0 & a_{42} & a_{43} & -a_{44} \end{bmatrix} \tag{4}$$

and

$$B = \begin{bmatrix} 0 \\ b_2 \\ 0 \\ -b_4 \end{bmatrix} \tag{5}$$

where $a_{22} = \frac{4\bar{B}}{D_l}$, $a_{23} = \frac{3mg}{D_l}$, $a_{24} = \frac{6B_\theta}{lD_l}$, $a_{42} = \frac{6\bar{B}}{lD_l}$, $a_{43} = \frac{6\bar{M}g}{lD_l}$, $a_{44} = \frac{12\bar{M}B_\theta}{ml^2D_l}$, $b_2 = \frac{4B_l}{D_l}$, and $b_4 = \frac{6B_1}{lD_l}$, for $D_l = 4\bar{M} - 3m$, $\bar{B} = \frac{K_gB_m}{r^2} + \frac{K_g^2K_iK_b}{r^2R_a}$, $B_l = \frac{K_gK_i}{rR_a}$, $\bar{M} = \frac{m+M+(K_g*J_m)}{r^2}$, and where g is gravity, R_a is armature resistance, r is driving wheel radius, J_m is motor rotor inertia, B_m is coefficient of viscous friction, B_θ is coefficient of viscous friction, K_i is motor torque constant, K_b is motor back-e.m.f. constant, K_g is gear ratio, M is cart mass, m is pendulum mass, and l is pendulum length.

The system is then described as

$$\dot{X} = \left(A + BK_\sigma\right)X \tag{6}$$

where K_σ is one of the safety, baseline, or experimental controller gains and the solution to this is

$$X = e^{(A+BK_\sigma)t}X_0 \tag{7}$$

where X_0 is an initial condition within the stabilizable region.

```
**automaton** Plant($A$ : Real$^{4\times 4}$, $B$ : Real$^{4\times 1}$)
**variables**
  **input** $u$ : Real;
  **output** $\theta$ : Real; $x$ : Real;
  **internal** $\theta_h$ : Real := 0; $\dot{\theta}_h$ : Real := 0;
    $x_h$ : Real := 0; $\dot{x}_h$ : Real := 0;

**trajectories**
  **trajdef** plantDynamics
    **evolve** $d(x_h) = \dot{x}_h$;
      $d(\dot{x}_h) = -a_{22}\dot{x}_h - a_{23}\theta_h + a_{24}\dot{\theta}_h + b_2u$;
      $d(\theta_h) = \dot{\theta}_h$;
      $d(\dot{\theta}_h) = a_{42}\dot{x}_h + a_{43}\theta_h - a_{34}\dot{\theta}_h - b_4u$;
      $\theta = \theta_h$; $x = x_h$;
```

Fig. 3. Linearized Plant Model

B. Sensor Model

The sensor model can be seen in Figure 4, and is an analog-to-digital converter (ADC). This is where the delay is being introduced to the system in the form of measurement delay, primarily due to filtering. Between the sensor and the controller there exists a low-pass filter which will introduce a delay between one and two sampling periods ($T_s = 20$ milliseconds) long. There is also nondeterministic delay introduced due to the ADC and controller not being synchronized. The ADC is utilizing a sample-and-hold, which must be finished before the controller reads the value, but the time difference between when the sample-and-hold finishes and the controller reads the value is unknown, but assumed to be less than one sampling period (T_s) long.

```
automaton Sensor($T_s$ : Real) where $T_s > 0$
  signature
    output sample($\theta', x'$ : Real)

  variables
    input $\theta$ : Real; $x$ : Real;
    internal $\theta_s$ : Real; $x_s$ : Real;
      $now_s$ : Real := 0;
      $next\_sample$ : AugmentedReal := $T_s$;
      let $time\_left := next\_sample - now_s$

  transitions
    output sample($\theta', x'$)
      pre $now_s = next\_sample$
          $\wedge\, \theta' = \theta_s$
          $\wedge\, x' = x_s$;
      eff $next\_sample := next\_sample + T_s$;

  trajectories
    trajdef periodicSample
      stop when $now_s = next\_sample$
      evolve $d(now_s) = 1$; $\theta_s = \theta$; $x_s = x$;
```

Fig. 4. Sensor

C. Actuator Model

The actuator model serves little purpose at the present time, but is included for completeness as seen in Figure 5, and may be used in the future to model the quantization error and delay introduced by this block. The actuator is a digital-to-analog converter (DAC), followed by a higher power voltage source to drive the DC motor.

```
automaton Actuator($T_a$ : Real) where $T_a > 0$
  signature
    input controllerOutput($u'$ : Real)

  variables
    output $u$ : Real;
    internal $u_a$ : Discrete Real := 0;
      $ready_a$ : Bool := $false$;
      $now_a$ : Real := 0;

  transitions
    input switchingOutput($u'$)
      eff $u_a = u'$;
        $ready_a := true$;

  trajectories
    trajdef hold
      evolve $d(now_a) = 1$; $u = u_a$;
```

Fig. 5. Actuator

D. Controller Model

The controller is implemented following the Simplex architecture of [1]. The Simplex architecture is built on the concept of analytic redundancy of [7], in this case, that several controllers implement the same control objective with different performance. There are three controllers in this architecture, a safety controller that can stabilize the system from the largest set of initial conditions but with poor performance, a baseline controller that has better performance than the safety controller but cannot stabilize from such a wide set of initial conditions, and an experimental controller that has the best performance but the smallest set of stabilizable initial conditions. The usefulness of such an architecture could be in upgrades of real-time systems that cannot afford downtime even for maintenance, such as critical infrastructure. In our case, the architecture is used to differentiate between the different sets of recoverable initial conditions and performance. Each controller uses linear state feedback for stabilization, with the only difference being higher gains in the baseline and experimental controllers to stabilize faster, with the downside that they cannot recover from certain initial conditions that the safety controller can.

Note that, as only θ and x are measurable, $\dot{\theta}$ and $\dot{x}$ are constructed by the first-order approximations $\dot{\theta}(t) = \frac{[\theta(t)-\theta(t-mT_s)]}{mT_s}$ and $\dot{x}(t) = \frac{[x(t)-x(t-mT_s)]}{mT_s}$, where m is an integer greater than one (chosen as 2 by experimentation). In the safety, baseline, and experimental controller automata, this first-order approximation is accomplished by storing a buffer of previous sampled values. It is more logical to do this computation here than in the sensor automaton as this calculation is done in the controllers of the implemented system.

1) Discretization: As the system is implemented in embedded software, the system model needs to be discretized. In continuous-time the system solution is of Equation 7, the discrete solution is of the form

$$X\left(t_0\right) = FX\left(t_0 - T_s\right) + Gu\left(t_0 - T_s\right) \qquad (8)$$

where

$$F = e^{AT_s} \qquad (9)$$

and

$$G = \int_0^{T_s} e^{A\tau} d\tau \qquad (10)$$

To get back to linear state feedback form, take the control as

$$u\left(t_0\right) = KX\left(t_0 - T_s\right) \tag{11}$$

yielding the full form

$$X\left(t_0\right) = FX\left(t_0 - T_s\right) + GBKX\left(t_0 - 2T_s\right) \tag{12}$$

This allows easy state projection of the form in Equation 13, where n is the number of sampling periods to project forward.

$$X\left(t_0 + nT_s\right) = \tag{13}$$
$$FX\left(t_0 + \left(n-1\right)T_s\right) + GBKX\left(t_0 - \left(n-2\right)T_s\right)$$

This projection is done to reduce the error introduced from the various delays in the system, primarily the digital implementation delay (that we are measuring the present state at t_0 and controlling the next state at $t_0 + T_s$) and the aforementioned filter delay.

```
automaton SafetyController($K_{sc}$ : Real$^{4\times1}$, $T_s, T_{safety}$ : Real, $m$ : Int)
  signature
    input sample($\theta', x'$ : Real)
    output safetyOutput($u'_{sc}$ : Real)

  variables
    internal $\theta_{sc}$ : Real := 0; $\dot{\theta}_{sc}$ : Real := 0;
      $x_{sc}$ : Real := 0; $\dot{x}_{sc}$ : Real := 0;
      $u_{sc}$ : Real := 0; $rt$ : Real := 0;
      $next\_cycle$ : AugmentedReal := $T_{safety}$;
      buffer: Seq[$prevTheta$ : Real, $prevX$ : Real] := {};
      let $time\_left := next\_cycle - rt$;
      let $length := length(buffer)$

  transitions
    input sample($\theta', x'$)
    eff buffer := buffer ⊢ [$\theta_{sc}, x_{sc}$]
      $\theta_{sc} := \theta'$; $x_{sc} := x'$;

    output safetyOutput($u'_{sc}$)
      pre $rt = next\_cycle \wedge u'_{sc} = u_{sc}$
      eff $next\_cycle := next\_cycle + T_{safety}$;
        $\dot{\theta}_{sc} := [\theta_{sc} - head(buffer).prevTheta]/(mT_s)$;
        $\dot{x}_{sc} := [x_{sc} - head(buffer).prevX]/(mT_s)$;
        $u'_{sc} := K_{sc1} * x_{sc} + K_{sc2} * \dot{x}_{sc} + K_{sc3} * \theta_{sc} + K_{sc4} * \dot{\theta}_{sc}$;
        buffer := $tail$(buffer);

  trajectories
    trajdef periodicControl
      stop when $rt = next\_cycle$
      evolve $d(rt) = 1$;
```

Fig. 6. Safety Controller

2) Safety, Baseline, and Experimental Controllers: The baseline and experimental controllers are of the exact same form as the safety controller in 6, except that they use different gain matrices, K_{bc} for the baseline controller, and K_{ec} for the experimental controller. We take the gains as follows to maximize the stability region for the safety controller, and to improve the performance for the baseline and experimental controllers

$$K_{sc} = \begin{bmatrix} 6.0 \\ 20.0 \\ 60.0 \\ 16.0 \end{bmatrix} \tag{14}$$

$$K_{bc} = \begin{bmatrix} 8.0 \\ 32.0 \\ 120.0 \\ 12.0 \end{bmatrix} \tag{15}$$

$$K_{ec} = \begin{bmatrix} 10.0 \\ 36.0 \\ 140.0 \\ 14.0 \end{bmatrix} \tag{16}$$

3) Switching Controller: The switching controller description is to be written, and it should be noted that the automaton of the switching controller in Figure 9 is not completely accurate at this time.

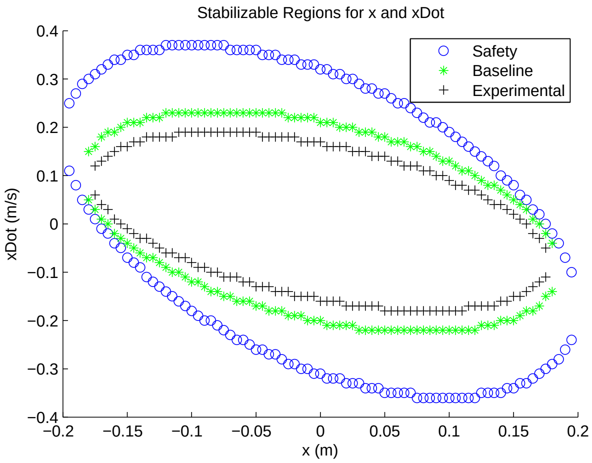

Fig. 7. Stabilizable Region for x and $\dot{x}$

III. Stability Analysis

The stability analysis follows from classical Lyapunov stability (see, e.g. [8] or [9]) and extends to the small-gain stability of systems with disturbances, such as in [3].

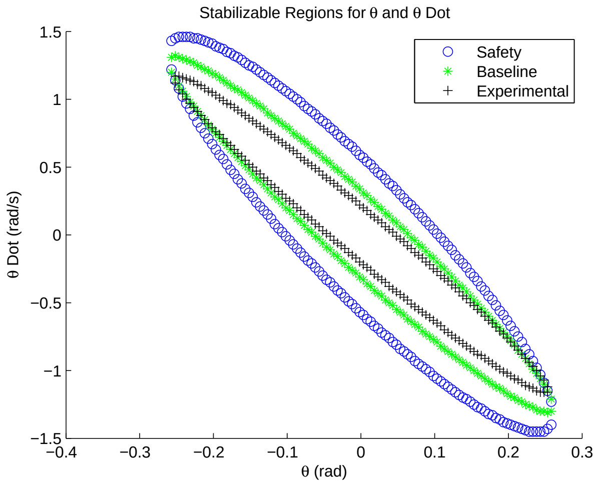

Fig. 8. Stabilizable Region for θ and $\dot{\theta}$

A. Lyapunov Analysis

From classical Lyapunov stability, we know that for linear time-invariant systems, if there exists some positive definite P that solves the Lyapunov equation in Equation 17 for some positive definite Q

$$PA + A^T P + Q = 0 \tag{17}$$

then the system is asymptotically stable and a function V defined by Equation 18 is a Lyapunov function.

$$V = X^T P X \tag{18}$$

For our system, we have the following P matrices

$$P_{sc} = \begin{bmatrix} 8.494 & 32.758 & 3.785 & 11.300 \\ 32.758 & 217.957 & 16.617 & 75.446 \\ 3.785 & 16.617 & 5.319 & 5.785 \\ 11.300 & 75.446 & 5.785 & 26.406 \end{bmatrix} \tag{19}$$

$$P_{bc} = \begin{bmatrix} 10.623 & 48.942 & 6.132 & 16.600 \\ 48.942 & 350.558 & 25.975 & 119.079 \\ 6.132 & 25.975 & 44.397 & 8.836 \\ 16.600 & 119.079 & 8.836 & 42.101 \end{bmatrix} \tag{20}$$

```
automaton SwitchingController($u_{min}, u_{max}, T_{switching}, T_s$ : Real)
  type
    Mode = Enumeration [saf, bas, exp]

  signature
    input sample($\theta', x'$ : Real)
    input safetyOutput($u'_{sc}$ : Real)
    input baselineOutput($u'_{bc}$ : Real)
    input expOutput($u'_{ec}$ : Real)
    output switchingOutput($u'$ : Real)

  variables
    internal $\theta_{sw}$ : Real := 0; $\dot{\theta}_{sw}$ : Real := 0;
      $x_{sw}$ : Real := 0; $\dot{x}_{sw}$ : Real := 0;
      $u_{sw}$ : Real := 0; $rt$ : $Real$ := 0;
      $next\_cycle$ : AugmentedReal := $T_{switching}$;
      let $time\_left := next\_cycle - rt$;
      $u_{sc}$ : Real; $u_{bc}$ : Real; $u_{ec}$ : Real;
      $u_c$ : Real := 0; $ready_c$ : Bool := $false$;
      $mode$ : $Mode$ := $saf$; $rt$ : $Real$ := 0;

  transitions
    input sample($\theta', x'$)
      eff $\theta_{sc} := \theta'$; $x_c := x'$;
      $\dot{\theta}_{sc} := [\theta(t) - \theta(t - mT_s)]/(mT_s)$;
      $\dot{x}_{sc} := [x(t) - x(t - mT_s)]/(mT_s)$;

    input safetyOutput($u'_{sc}$)
      eff $u_{sc} = u'_{sc}$;
      set ready?

    input baselineOutput($u'_{bc}$)
      eff $u_{bc} = u'_{bc}$;
      set ready?

    input expOutput($u'_{ec}$)
      eff $u_{ec} = u'_{ec}$;
      set ready?

    output switchingOutput($u'$)
      pre $now_s = next\_cycle$
      eff $next\_cycle := next\_cycle + T_{safety}$;
      unset all ready signals

  trajectories
    trajdef periodicControl
      stop when $rt = next\_cycle$
      evolve $d(rt) = 1$;
```

Fig. 9. Switching Controller

$$P_{ec} = \begin{bmatrix} 10.183 & 45.835 & 5.660 & 15.507 \\ 45.835 & 332.254 & 22.748 & 112.560 \\ 5.660 & 22.748 & 36.583 & 7.718 \\ 15.507 & 112.560 & 7.718 & 39.2890 \end{bmatrix} \tag{21}$$

and then the corresponding Lyapunov functions $V_{sc} = X^T P_{sc} X$, $V_{bc} = X^T P_{bc} X$, and $V_{ec} = X^T P_{ec} X$. Figures 10, 11, and 12 show the Lyapunov functions along the system trajectory from initial conditions of $X_0 = \begin{bmatrix} 0.300 & -0.200 & 0.061 & -0.050 \end{bmatrix}^T$.

B. Small-Gain Theorems

Small-gain theorems allow us to bound the error introduced by the delay to talk about a system

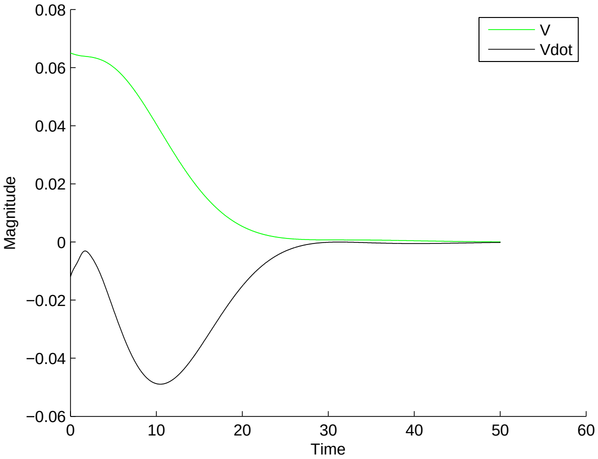

Fig. 10. Lyapunov Function for Safety Controller

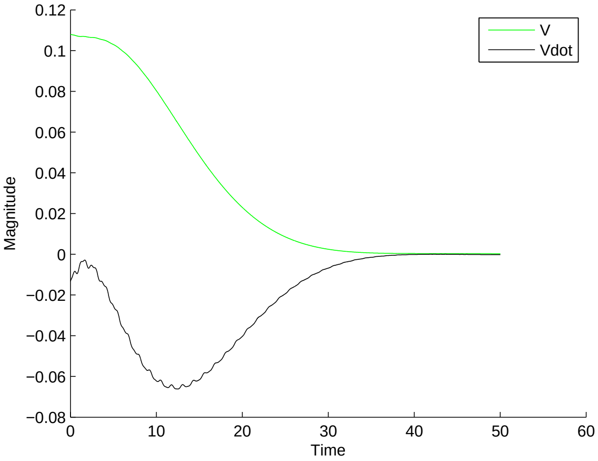

Fig. 11. Lyapunov Function for Baseline Controller

without delay, and instead with a disturbance injection. Writing the delayed control as

$$u(t) = KX(t-\tau) = KX(t) + \theta(t) \tag{22}$$

where $\theta(t) = KX(t-\tau) - KX(t)$ is the error from delay.

Now writing the system model again with this delay error as a disturbance input we have

$$\dot{X}(t) = (A + BK)X(t) + B\theta(t) \tag{23}$$

The Lyapunov function time-derivative changes in the form of

$$\dot{V} = -X^T QX + X^T PB\theta \tag{24}$$

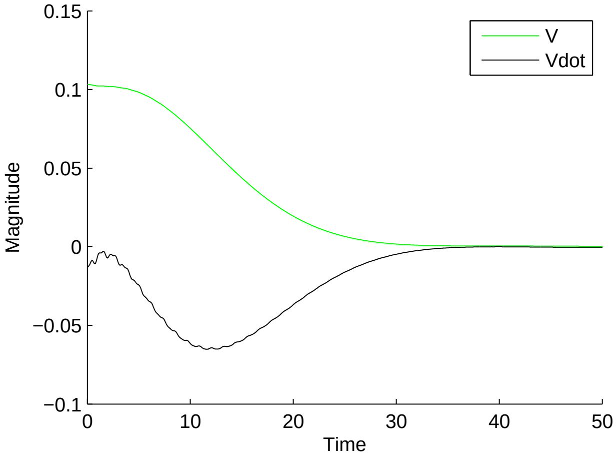

Fig. 12. Lyapunov Function for Experimental Controller

where P and Q are the same functions defined above in Equation 17.

We then apply the conservative bound

$$\dot{V} \le -X^T QX + |X| \cdot |\theta| \cdot ||PB|| \tag{25}$$

which can be computed as

$$\dot{V} \le -\lambda_{min}(Q)|X|^2 + |X| \cdot |\theta| \cdot ||PB|| \tag{26}$$

where $\lambda_{min}(Q)$ is the smallest eigenvalue of Q.

$$\dot{V} = -|x|(\lambda_{min}(Q)|x| - ||PB|| \cdot |\theta|) \tag{27}$$

$$|x| > \frac{||PB||}{\lambda_{min}(Q)} \cdot |\theta| \tag{28}$$

Define $\rho(r) = cr$, and take

$$c = \frac{||PB||}{\lambda_{min}(Q)} \tag{29}$$

Writing θ now as

$$\theta = -\int_{t-\tau}^{t} K(AX(s) + BKX(s-\tau))\,ds \tag{30}$$

we can now bound θ as

$$|\theta| \le \tau(||KA|| + ||KBK||) \cdot ||X||_{[t-2\tau,t]} \tag{31}$$

Defining a constant d as this bound

$$d = (||KA|| + ||KBK||) \tag{32}$$

The small-gain theorem states then, assume there exists some τ such that

$$\tau cd < 1 \tag{33}$$

then we can define the maximum tolerable delay as

$$\tau < \frac{1}{cd} \tag{34}$$

Following this definition for τ, we compute τ_σ for each of the controllers, yielding $\tau_{sc} = 9$, $\tau_{bc} = 12.3$, and $\tau_{ec} = 18.6$ milliseconds of delay is tolerable for each of the respective controllers to still ensure stability. Note that all of these tolerable delays are less than one control cycle period of $T_s = 20$ milliseconds. Also note that this result has not included state projection of the type defined in Equation 13, and used a rather conservative bound in Equation 26, which looking at the work of [10] could perhaps improve.

IV. Results

Our analysis of the inverted pendulum system using the small-gain theorem indicates that the model-based state projection used in the controller is necessary for stability. The maximum delay the 'pure' system (without state projection) can tolerate is less than the sampling period. Of interesting note is that the tolerable delay is directly related to the magnitude of the gains of the linear state feedback. The experimental controller, with the highest gain, can tolerate the most delay, while the safety controller, with the smallest gain can tolerate the least.

V. Future Work

There is some remaining work to complete for this case study to provide interesting results. First, we will use small-gain theorems to find the maximum tolerable delay with state projection, and in addition, compare the subsequent stability regions by solving a linear-matrix inequality (LMI) problem that computes this region. An offhand guess is that the new maximum tolerable delay will be roughly one sampling period plus the previously computed tolerable delay. After that, we will look at the small-gain result in the system with minor nonlinearities that can be linearized, such as the $sin(\theta)$ term which for small θ can be linearized to θ. The last step is an investigation of the average-dwell time (ADT) of the switching between the safety, baseline, and experimental controllers to see if ADT relates to the small-gain delay and stability regions.

As far as long-term future work, for the verification of the model to most closely match the real system, analyzing both the measurement and actuation errors from quantization and delay (from analog-to-digital conversion and digital-to-analog conversion, respectively) would provide a complete answer of the disturbances introduced by these non-ideal blocks. There is also another avenue potentially to follow in the area of software verification through the co-stability concept.

VI. Conclusion

This case study of the inverted pendulum with a controller following the Simplex architecture displays that recent results in small-gain theorems can be useful in practical systems. Since the maximum tolerable delay for stability was less than the control period length, it was necessary to project the state forward to the next control cycle to guarantee stability. As described in the future work section, there is more extension to do on this result for it to be truly interesting, which is what is to be completed before the final deadline.

References

[1] D. Seto, B. Krogh, L. Sha, and A. Chutinan, "The simplex architecture for safe on-line control system upgrades," in *Proc. American Control Conference*, Philadelphia, PA, Jun. 1998, pp. 3504–3508.

[2] D. Seto and L. Sha, "A case study on analytical analysis of the inverted pendulum real-time control system," Carnegie Mellon Univ., Pittsburgh, PA, CMU/SEI Tech. Rep. 99-TR-023, Nov. 1999.

[3] D. Liberzon, "Quantization, time delays, and nonlinear stabilization," *IEEE Trans. Autom. Control*, vol. 51, no. 7, pp. 1190–1195, Jul. 2006.

[4] R. Sanfelice and A. Teel, "On hybrid controllers that induce input-to-state stability with respect to measurement noise," in *Proc. 44th IEEE Conf. on Decision and Control*, Seville, Spain, Dec. 2005, pp. 4891–4896.

[5] N. Lynch, R. Segala, and F. Vaandrager, "Hybrid i/o automata," *Inf. Comput.*, vol. 185, no. 1, pp. 105–157, 2003.

[6] S. Mitra, "A verification framework for hybrid systems," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA 02139, Sep. 2007. [Online]. Available: http://users.crhc.uiuc.edu/mitras/research/thesis.pdf

[7] M. Bodson, J. Lehoczky, R. Rajkumar, L. Sha, and J. Stephan, *Analytic redundancy for software fault-tolerance in hard real-time systems*. Kluwer Academic Publishers, 1994.

[8] C. Chen, *Linear System Theory and Design*, 3rd ed. New York, NY: Oxford University Press, 1999.

[9] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Upper Saddle River, NJ: Prentice Hall, 2002.

[10] E. Fridman, M. Dambrine, and N. Yeganefar, "Input to state stability of systems with time-delay: A matrix inequalities approach," *Automatica*, vol. 44, no. 9, pp. 2364–2369, 2008.