Formal verification of One Dimensional Time Triggered Velocity PID Controllers

Kenneth Payson
kpayson@andrew.cmu.edu

12/09/14

1: Abstract

This paper provides a formal proof of the safety of a time triggered velocity PID controller that are subject to random force disturbances. This proof is verified using the Keymaera proof assistant. Specifically, this paper shows that given sufficient initial conditions on the gains of the controller, the resulting velocity will remain in some safe range $v_{min} \leq v \leq v_{max}$. Time triggered velocity PID controllers are frequently used in industrial applications to achieve a desired velocity in the face of unknown variables in the system. They can be combined with more complicated controllers. This paper shows how a hybrid controller that contains a velocity PID controller can also be shown to be safe. One such system that relies on a velocity controller is the *European Train Control System*. Proving that a velocity PID controller remains safe in the face of sufficiently small random forces is critical in guaranteeing that the overall behaviour of the system is not only safe in the worst case, but also that it functions correctly under normal operating circumstances.

2: Introduction

PID controllers are feedback based controllers used frequently in industrial applications. PID controllers attempt to minimize the difference between some target state and the current state. Control is applied to the system based on 3 terms, position, which is simply the difference between the desired state and the current state, integral, which is the sum of previous errors, and derivative, which is rate at which the system is changing. PID controllers are highly useful in systems where the mechanics of the system are not fully understood because they are agnostic to the underlying system. However, PID controllers provide no guarantees about the safety of the controller. In order to ensure the safety of PID based controllers some formal verification is necessary for the system.

One such way of verifying the safety of a PID controller is to use proof assistance software such as Keymaera. Keymera is a powerful tool that has been used to prove the safety of real world systems such as the breaking mechanism on trains [1]. Prooving the safety of PID controllers provides additional safety guarantees about complex systems.

3: Existing Research

There has been extensive research on the topic of PID controllers. Research into formal verification of PID controllers has also been conducted.

Platzer et Quesel [1] investigated the safety of PI controllers as they applied to the *European Train Control System* (ETCS). The ETCS system relies on PI controllers to regulate velocity control. In order to prove that the PI controller was safe, the upper and lower bounds, A and -b, for the acceleration of the ETCS system was used. It was shown that given some velocity v at time t that was in the evolution domain of an ideal velocity controller, that same velocity was also reachable at time t given a sufficient PI controller. It should be noted that the integral term s evolved continuously in the hybrid program under study. This differs from the current project where the integral term is taken to be time-triggered.

Arechiga et Loos,Platzer,Krough [2] further the results of [1] to the case of PID controllers for velocity. They examine the case of an *Intelligent Cruise Control* (ICC) system and its formal verification using an automated theorem prover. Similar to [1], the system is constrained to accelerations $a \in [-B, A]$. Furthermore, the integral term z is constrained to be in $[z_{min}, z_{max}]$ Formally, the PID controller is described as

$$h(x_s(t), z_s(t)) = \begin{cases} a_{pid} & -B \leq a_{pid} \leq A \\ A & a_{pid} > A \\ -B & a_{pid} < -B \end{cases}$$

Where $z_s \in [z_{min}, z_{max}]$ and a_{pid} is pure PID velocity controller. Furthermore, this system is divided into to two regions of operation, $SAFE_\epsilon$ consists of regions where any $a \in [-B, A]$ is considered safe, and $\overline{SAFE_\epsilon}$ which is the complementary region. The a_{pid} controller is only exercised in the $SAFE_\epsilon$ region, and formal proof techniques were used to show that the output of the system in this region was always safe.

4: General Approach

This project attempts to formally prove safety properties about velocity PID controllers. Unlike prior research in [1] and [2], this project will attempt to prove safety constraints on velocity PID controllers themselves, as opposed to proving safety constraints on PID controlled systems. The key difference here is that the PID controller is allowed to operate in all regions of the system, instead of relying on a fallback controller to ensure the safety of the system. Specifically, the project will focus on a one dimensional time-triggered control system. A PID controller will be responsible for controlling the velocity of a system that evolves according to Newtonian physics. Formally, the system evolves according to

$$v' = a$$

.

a will be controlled by a canonical PID controller of the form

$$a = K_P(v_{set} - v) + K_I(v_i) - K_D(a)$$

Where v_{set} is the controller setpoint, v is the velocity, a is the acceleration, and v_i is the integral term, or more formally

$$v_i(t) = \int_0^t (v_{set} - v(t))dt$$

Here, the P term is considered to be $(v_{set} - v)$, the I term is taken to be v_i and the D term is considered to be $|a|$. The controller is time triggered, meaning that all the P, I, D components are determined at the beginning of an iteration of the controller and the system is allowed to evolve given these values of P, I and D for some $t \leq T$.

The controller will also be subject to random disturbances of the form F where $F \in [-F_{max}, F_{max}]$.

With the random disturbances, the system now evolves according to

$$a^+ = a + F$$

$$v' = a^{+}$$

With random disturbances, the P term remains as $v_set - v$. Likewise, the I term remains as v_i. However, the D term is now taken to be $a^{+} = a + F$. Thus, the overall action of our controller can be described as

$$a = Kp(v_set - v) + K_I(v_i) - K_D(a^{+})$$

.

This system is considered to be safe if it can be shown for some $v_{min} < v_{max}$ such that $\forall t \geq 0$, $v_{min} \leq v(t) \leq v_{max}$. It should be noted that this is a sufficient condition for non-divergence of the controller, but does not ensure convergence of the controller.

This safety condition provides strong guarantees for the general safety of mechanical systems that contain velocity controllers. Primarily, this result suffices to show that if some mechanical system is considered unsafe when operating outside a given velocity range, then the system does not require intervention by a safety controller. Furthermore, it provides guarantees on the velocity during the controller operation that can in turn be used to provide guarantees about the position during control.

4: *dL* Formalization

This section formalizes the controller described in section (3). It uses hybrid program syntax and *dL* notation that is described in depth in [1].

Let K_p, K_i, K_d be positive real constant coefficients of the PID controller.

Let I be the integral term of the PID controller.

Let I_{max} be the maximum integral term of the PID controller.

Let $v_{min} < v_{max}$ be the safety constraints on the velocity of the system.

Let $F_{min} \leq F_{max}$ be the minimum and maximum values for the force disturbance F on the system.

Let $0 \leq T$ be a time trigger of the control system.

Let v_{set} be the set point of the controller.

Let v be the current velocity of the controller.

Let a be the current acceleration of the controller.

Let L be some logical formula of the above variables that guarantees the safety of the system, described in section (6).

The system evolution can be described as :

$$E \equiv F := *; ?(F \leq F_{max} \& F \geq -F_{max}); t := 0; v' = a + F, t' = 1 \& t \leq T;$$

The system control can be described as :

$$C \equiv I := I + T(v_{set} - v);$$

$$(I := I; ?(I >= -Imax \& I <= Imax)) ++ ((I := -Imax; ?(I < -Imax)) ++ (I := Imax; ?(I > Imax)));$$

$$a := K_p(v_{set} - v) + K_i(I)$$

$$a := a - K_d(a + F)$$

The dL formula that is equivalent to the safety of this system is then

$$(L \wedge t = 0 \wedge I = 0) \rightarrow \{(C; E;)^*\}[v_{min} \leq v \leq v_{max}]$$

5: Intermediate Results

(a)

In this step, the safety of a P controller was proved with no disturbance and explicitly set gains. Furthermore, the set point of the controller, v_{set} was taken to be 0. Additionally, we had safety constraints of the form $-v_{max} <= v <= v_{max}$. Specifically, the system evolved according to

$$v' = a$$

. The controllers action was taken to be

$$a = Kp * (-v)$$

.

The key initial condition required to prove this controller safe was $Kp = \frac{1}{T}$. We can see that at each iteration of the controller, $|\Delta v| <= |Kp * T * v|$. By setting $Kp = \frac{1}{T}$, we have ensured that at any iteration, $|\Delta v| <= |v|$. In essence, we have ensured that the controller never "overshoots" the set point in any iteration. A formal proof of this using the Keymaera Proof Assistant is attached (`project_ss1.key.proof`).

(b)

In this step, the safety of a PI controller was proved with no disturbance and a range of valid gains. Furthermore, the set point of the controller, v_{set} was taken to be 0. Additionally, we had safety constraints of the form $-v_{max} <= v <= v_{max}$. The system evolved according to

$$v' = a$$

. The controllers action was taken to be

$$a = Kp * (-v) + Ki * I$$

. The integral term I, was initially set to be 0. Upon each iteration of the controller, The I term is updated according to

$$I = I - v * T$$

. Furthermore, a new term I_{max} is introduced to bound the absolute value of I. Thus, at each iteration, a new value of I is computed according to

$$I = Max(Min(I - vT, I_{max}), -I_{max})$$

. The introduction of an I_{max} term is not as artificial as it may first appear. Bounds on the integral term are often introduced in industrial applications. In previous examinations of safety properties of PID controllers, bounds on the integral term have been used as well [2]. Furthermore, the canonical representation of a controller in Keymaera allows the system to evolve for any t such that $0 <= t <= T$, where T is the time trigger on the system. Clearly this representation would prove problematic for proving boundedness on the I term.

In order to prove the safety of this controller, the initial condition

$$v_{max} * Kp \geq I_{max} * Ki$$

was introduced. This condition ensures that at v_{max}, the action of the P term will always be greater that the action of the I term. The action of the P term will always force the control output towards the set point, and so at v_{max}, the controller must accelerate towards the set point. In conjunction with $Kp * T <= 1$, this initial condition is in fact sufficient to show that the controller will never exceed the set point throughout the entire execution of the controller. A formal proof of this using the Keymaera Proof Assistant is attached (`project_ss2.key.proof`).

(c)

In this step, the safety of a PI controller with random force disturbances was proven. The set point of the controller, v_{set} was taken to be 0. Additionally, we had safety constraints of the form $-v_{max} <= v <= v_{max}$. The system evolved according to

$$v' = a + F$$

where $F \in [-F_{max}, F_{max}]$ is a bounded random force. It should be noted that the random force was constant throughout a full execution of the dynamics. However, because the dynamics are allowed to evolve for any amount of time t such that $0 \leq t \leq T$, cases in which the random force changes rapidly are accounted for in this proof. The action of the controller remained unchanged from part (b) with

$$I = Max(Min(I - vT, I_{max}), -I_{max})$$

and

$$a = Kp * (-v) + Ki * I$$

In order to prove the safety of this controller, the initial condition

$$v_{max} * Kp \geq I_{max} * Ki$$

from part (b) was strengthened:

$$v_{max} * Kp \geq I_{max} * Ki + F_{max}$$

. This condition ensures that at v_{max}, the action of the P term will always be greater that the action of the I term as well as the F term. The action of the P term will always force the control output towards the set point, and so at v_{max}, the controller must accelerate towards the set point, regardless of the values of the I term and the random force F. A formal proof of this using the Keymaera Proof Assistant is attached (`project_ss3.key.proof`).

(d)

In this step, the safety of a PID controller with random force disturbances was proven. The set point of the controller, v_{set} was taken to be 0. Additionally, we had safety constraints of the form $-v_{max} <= v <= v_{max}$. The system evolved according to

$$v' = a + F$$

where $F \in [-F_{max}, F_{max}]$ is a bounded random force. The action of the controller was

$$I = Max(Min(I - vT, I_{max}), -I_{max})$$

and

$$a = Kp * (-v) + Ki * I; a = a - Kd(a + F)$$

The action of the controller is a slight adaptation of the originally proposed controller

$$a = Kp * (-v) + Ki * I - Kd * (a + F)$$

This deviation from the canonical form of a PID controller was made such that for each iteration of the control loop, the D term opposes the controllers output. The challenge with verifying the canonical controller is that under a time triggered system, on iterations where the sign of the acceleration changes, the action of the D term acts with the acceleration on the next iteration.

In a canonical PID controller, the intention of the D term is to oppose the change in the error of the controller. This leads to more rapid settle times of the controller (the time it takes for the controller to stabilize on the set point). Thus, it should be considered a satisfactory substitution to implement the PID control in two steps if it ensures that the derivative term will always oppose the current change in error (velocity).

In order to prove this controller, the initial condition from part (c) was modified such that the I_{max} and P terms were scaled by $1 - Kd$. In the modified controller, the D term always acts against the current direction of motion. However, F_{max} term was scaled by $1 + Kd$ because the evolution has a total contribution of random forces that can be described as $F_1 - Kd * F_0$, where F_1 is the random force generated during this iteration, and F_0 is the random force generated by the previous iteration. Because they are independent, the total random force contribution in this iteration can be as high as $(1 + Kd)F_{max}$.

This leads to the new initial condition

$$(1 - Kd) * V_{max} * Kp \geq (1 + Kd) * F_{max} + (1 - Kd) * I_{max} * Ki$$

A formal proof of this using the Keymaera Proof Assistant is attached (`project_ss4.key.proof`).

(e) In this step, the safety of a PID controller with random force disturbances was proven. The set point of the controller, v_{set} was constrained such that $v_{min} \leq v_{set} \leq v_{max}$. The safety constraint was of the form $v_{min} \leq v \leq v_{max}$. A more detailed description of the key initial constraints is described in the following section.

A formal proof using the Keymaera Proof Assistant is attached (`project_ss5.key.proof`).

6: Proof Sketch

In section (4), a dL formalization of the PID controller as well as the safety condition was presented. This section presents an overview of the initial conditions used to prove the controller's safety, as well as an informal overview of some of the key steps used in proving this controller to be safe. A formal proof using the Keymaera Proof Assistant can be found in the accompanying file (`project_ss5.key.proof`).

The full set of initial conditions used were

1. $T > 0 \wedge I_{max} \geq 0 \wedge F_{max} \geq 0$

2. $K_p \geq 0 \wedge K_i \geq 0 \wedge K_d \geq 0$
3. $I = 0 \wedge F = 0$
4. $v_{min} \leq v \leq v_{max}$
5. $v_{min} \leq v_{set} \leq v_{max}$
6. $K_p * T \leq 1$
7. $K_d \leq 1$
8. $(1 - K_d)(v_{max} - v_{set}) * K_p \geq (1 + K_d) * F_{max} + (1 - K_d) * I_{max} * K_i$
9. $(1 - K_d)(v_{set} - v_{min}) * K_p \geq (1 + K_d) * F_{max} + (1 - K_d) * I_{max} * K_i$

The initial conditions in (1) are intuitive properties of the system. The initial conditions in (2) simplify the notion of a PID controller to accept constants that are strictly positive. The initial conditions in (3) set the integral term and random force term to be 0 initially.

The initial conditions (4) and (5) ensure that both the target set point and initial velocity are within the safety bounds of the system.

The initial condition (6) ensures that in any iteration of the controller, the change in velocity resulting from the P term is at most the offset between the velocity and the set point. This is used to show that the P term will never overshoot the set point in a given iteration.

The initial condition (7) ensures that the D term does not cause the system to become unstable by ensuring that it acts as a multiplicative constant (less than 1) on the P and I terms.

Initial conditions (8) and (9) are used to show that at the safety boundaries of the controller.

On each iteration of the control loop, the value of I is updated according to

$$I = Max(Min(I + (v_{set} - v) * T, I_{max}), -I_{max})$$

and a is updated according to

$$a = Kp * (-v) + Ki * I; a = a - Kd(a + F)$$

The system evolves according to

$$v' = a + F$$

, where $-F_{max} \leq F \leq F_{max}$

In order to show the safety of the system, we must show inductively that after each iteration of the control loop, we have both

1. $v \leq v_{max}$
2. $v \geq v_{min}$

Clearly this holds initially. Additionally, at any point during the controller's evolution, we have

$$-I_{max} \leq I \leq I_{max}$$

.

After the system evolves for some $0 \leq t \leq T$, we can see that the velocity can be expressed as

$$v = v_0 + (a + F)t = v_0 + (1 - K_d)(K_p(v_{set} - v) + K_i I - K_d(F^-)) * t + F * t$$

Note that $v_{min} \leq v_0 \leq v_{max}$ is the velocity before the current execution of the controller, and F^- is the random force from the last execution of the controller. Then clearly we have

$$((1 + K_d) * -F_{max} + (1 - Kd) * -I_{max} * Ki)t + v_0 \leq$$

$$v_0 + (1 - K_d)(K_p(v_{set} - v) + K_i I - K_d(F^-)) * t + Ft \leq$$

$$((1 + K_d)F_{max} + (1 - Kd)I_{max} * Ki)t + v_0$$

Invoking initial conditions (8) and (9) gives

$$(1 - K_d)(v_{min} - v_{set}) * Kp * t + v_0 \leq ((1 + K_d) * -F_{max} + (1 - Kd) * -I_{max} * Ki)t + v_0 \leq$$

$$v \leq$$

$$((1 + K_d)F_{max} + (1 - Kd)I_{max} * Ki)t + v_0 \leq (1 - K_d)(v_{max} - v_{set}) * Kp * t + v_0$$

Now invoking (7) and (6) gives the desired property

$$v_{min} \leq v \leq v_{max}$$

.

Because this property holds inductively, it must hold for any number of executions of the control loop.

7: Applications

PID controllers are frequently used in situations where the exact dynamics of the system are unknown. The system examined in this paper is specifically referred to as a velocity controller. The input to the controller is some target velocity, v_{set} and the output is some acceleration to reach the velocity, a. Consider the case of a cruise control system for a car. A PID velocity controller can be used to achieve a desired velocity. The controller output, a, could be used to compute how much fuel to deliver to the engines in order to produce the desired acceleration in the car. While it may seem like a simpler controller could be sufficient to produce the desired velocity, a PID controller proves more robust in the event of non-deterministic forces. For example, a deterministic controller would need to account for numerous external forces such as variable friction due to weather conditions. However, a PID controller does not require a detailed model to successfully achieve its target velocity.

Additionally, PID controllers are often used in conjunction with other controllers. Consider the case of some robotic arm that is given a planned trajectory to achieve some arbitrary goal. This trajectory consists of a set of target points, target velocities, $v_1, v_2, v_3...$ and target forces to be applied to the robotic arm $F_1, F_2, F_3...$ at each successive iteration of the control loop. While many of the dynamics of the system act deterministically, for example gravity and friction, some forces may be left un-accounted for such as air friction. Frequently, the controller will consist of two primary terms, a feed-forward term used to counteract the known dynamics of the system, as well as a PID controller to account for the unknown dynamics of the system. In this case, the output of the controller in the ith iteration could be expressed as $a_i = F_i + PID(v_i)$, where $PID(v_i)$ represents a velocity PID controller with set point v_i.

In both of the above applications, a formal proof of the safety of a PID controller as it has been presented in this paper is critical. While many cyber-physical systems rely on emergency controllers wrapped around the fine grained controllers to absolutely guarantee safety properties, It is important to know that for some small force disturbances, $F \in [-F_{max}, F_{max}]$, the PID controller will behave safely and not require the intervention of emergency controllers. Furthermore, formalizing these proofs in a proof verification system eliminates the possibility of incorrect proofs. It is clear how the safety guarantees provided in this paper would apply to the safety of a cruise control system that must remain between some velocity $v_{min} \leq v \leq v_{max}$ under normal operation.

However, even more complicated systems such as the robotic arm with a feedforward term also benefit from a formal proof of a PID velocity controller. If the feedforward term properly accounts for all known constant forces on the system, eg gravity, friction, then any additional force will cause the system to evolve according to the canonical

$$v' = a + F$$

. It is now clear that the verified velocity controller proven in this paper can ensure that robotic arms velocity will differ at most by some fixed amount if it can be shown that the feedforward component differs from the actual force on the arm at all times by at most F_{max}. Thus, even more complicated system can benefit from this formal proof a velocity PID controller.

8: Sources

[1] European Train Control System: A case study in formal verification. In Karin Breitman and Ana Cavalcanti, editors, 11th International Conference on Formal Engineering Methods, ICFEM, Rio de Janeiro, Brasil, Proceedings, volume 5885 of LNCS, pages 246-265. Springer, 2009

[2] Nikos Archiga, Sarah M. Loos, Andr Platzer and Bruce H. Krogh. Using theorem provers to guarantee closed-loop system properties. In Dawn Tilbury, editor, American Control Conference, ACC, Montral, Canada, June 27-29. pages 3573-3580. 2012. IEEE