PESSOA: A tool for embedded controller synthesis.[⋆]

Manuel Mazo Jr.[1], Anna Davitian[2], and Paulo Tabuada[1]

[1] CyPhyLab, University of California, Los Angeles CA 90095, USA
[2] Aerovironment Inc., Monrovia, CA 91016, USA

Abstract. In this paper we present Pessoa, a tool for the synthesis of correct-by-design embedded control software. Pessoa relies on recent results on approximate abstractions of control systems to reduce the synthesis of control software to the synthesis of reactive controllers for finite-state models. We describe the capabilities of Pessoa and illustrate them through an example.

1 Introduction

The synthesis of embedded control software is a challenging task due to the complex interactions between the physical and computational processes involved in embedded applications. The tool introduced in this paper, named Pessoa[3], automatically synthesizes embedded controllers enforcing several temporal logic specifications on physical systems. The controllers synthesized by Pessoa are described by Binary Decision Diagrams (BDD's) [Weg00], which have been shown to be adequate for the automatic generation of hardware [Be07] or software [Be99] implementations. The tool Pessoa illustrates the correct-by-design approach to the synthesis of embedded control software by generating BDDs, describing the control software, from a formal specification.

Most of the tools available for hybrid systems such as Ariadne [Ari], PHAVer [PHA], KeYmaera [KeY], Checkmate [Che], and HybridSAL [Hyba], focus on verification problems. Tools for the synthesis of controllers are more recent and include LTLCon [LTL] for linear control systems and the Hybrid Toolbox [Hybb] for piece-wise affine hybrid systems. What sets Pessoa apart from the existing synthesis tools is the nature of the abstractions (approximate simulations and bisimulations) and the classes of systems admitting such abstractions (linear, nonlinear, and switched [Tab09]). Although Pessoa does not support nonlinear and switched systems natively, they can already be handled as illustrated in the examples in [Pes].

[⋆] This work was partially supported by the NSF awards 0717188, 0820061, and 0834771.

[3] Pessoa Version 1.0 can be freely downloaded from `http://www.cyphylab.ee.ucla.edu/Pessoa/`.

2 Formal models for software and control

Pessoa uses the following notion of system allowing to describe both software and control systems under the same paradigm.

Definition 1. *A* system $S = (X, X_0, U, \longrightarrow)$ *is a tuple consisting of:*

- *a set of* states X*;*
- *a set of* initial states $X_0 \subseteq X$*;*
- *a set of* inputs U*;*
- *a* transition relation $\longrightarrow \subseteq X \times U \times X$*;*

System S is said to be finite when X has finite cardinality and metric when X is equipped with a metric $\mathbf{d} : X \times X \to \mathbb{R}_0^+$.

The "dynamics" of a system is described by the transition relation: existence of a transition $(x, u, x') \in \longrightarrow$ entails that upon the reception of input u at state x, system S evolves to state x'. In [Tab09] it is shown how systems of this form can represent both software and control systems modeling physical processes. While software models naturally lead to finite systems, obtaining models for control systems leading to finite systems requires of some abstraction techniques. Informally, a control system Σ is a differential equation of the form $\dot{\xi} = f(\xi, \upsilon)$, where $\xi(t)$ denotes the state of the system at time t, $\upsilon(t)$ the controlled input, and $\dot{\xi}$ denotes the time derivative of ξ. By adequately discretizing the state space of the differential equation and the input space, with discretization steps η and μ respectively, and by selecting an adequate sampling time τ, it has been shown in [Tab09] and references therein, that useful finite abstractions $S(\Sigma)$ for Σ can be obtained. Such finite abstractions can be related to the control system through a generalization of the notion of *alternating simulation relation* named *alternating approximate simulation relation* [Tab09]. The existence of such relations reduces the synthesis of controllers for Σ to the synthesis of reactive controllers for the finite abstraction $S(\Sigma)$.

3 Pessoa functionalities

Pessoa is a Matlab Toolbox and as such, although the core algorithms have been coded in C, the main functionalities are available through the Matlab command line. All the systems and sets manipulated by Pessoa are represented symbolically using Reduced Ordered Binary Decision Diagrams (ROBDDs) supported by the CUDD library [CUD]. Pessoa Version 1.0 offers three main functionalities:

1. the construction of finite abstractions of linear control systems;
2. the synthesis of reactive controllers for simple temporal logic specifications;
3. refinement of reactive controllers to a ROBDD description of the control software used to simulation of the closed-loop behavior in Simulink.

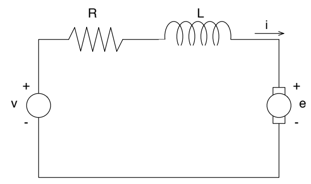
Pessoa currently supports the synthesis of controllers enforcing four kinds of linear temporal logic specifications defined using a target set $Z \subseteq X$ and a constraint set $W \subseteq X$:

1. **Stay:** $\Box\varphi_Z$ where φ_Z is the predicate defining the set Z;
2. **Reach:** $\Diamond\varphi_Z$;
3. **Reach and Stay:** $\Diamond\Box\varphi_Z$;
4. **Reach and Stay (in Z) while Stay (in W):** $\Diamond\Box\varphi_Z \wedge \Box\varphi_W$ where φ_W is the predicate defining the set W.

For lack of space it is not possible to provide further details on how the abstraction and synthesis algorithms are implemented. Such details can be found in [Pes].

4 Example

The example that we consider consists in regulating the velocity of a DC motor. The electric circuit driving the DC motor and the two linear differential equations defining the dynamics Σ of this system are shown in Figure 1. The variable x_1 describes the angular velocity of the motor, the variable x_2 describes the current i through the inductor, and the variable u represents the source voltage v that is treated as an input. The model parameters take the following values in international system units: $R = 500 \times 10^{-3}$ (resistance), $L = 1500 \times 10^{-6}$ (inductance), $J = 250 \times 10^{-6}$ (moment of inertia), $B = 100 \times 10^{-6}$ (viscous friction coefficient) and $k = 50 \times 10^{-3}$ (torque constant).



$$\dot{x}_1 = -\frac{B}{J}x_1 + \frac{k}{J}x_2 \qquad (1)$$
$$\dot{x}_2 = -\frac{k}{L}x_1 - \frac{R}{L}x_2 + \frac{1}{L}u. \qquad (2)$$

Fig. 1. DC motor circuit and equations describing its dynamics.

The control objective is to regulate the velocity around 20 rad/s. In practical implementations the DC motor is connected to a constant voltage source through an H-bridge. By opening and closing the switches in the H-bridge we can only choose three different values for the voltage: -10V, 0V, and 10V. In order to synthesize a controller under these input constraints we define the input space to be $\mathsf{U} = [-10, 10]$ and set the input quantization to $\mu = 10$. The time quantization is set to $\tau = 0.0001$ and the space quantization to $\eta = 0.05$. The resulting abstraction is computed in 17 minutes on a MacBook Pro with a 2.26 GHz Intel Core 2 Duo processor and 2GB of memory.

The target set is selected to be $Z = [19.5, 20.5] \times [-0.7, 0.7]$ so as to obtain a current ripple no larger than 0.7 Amperes around 0 while reaching the desired angular velocity. Moreover, by introducing the constraint set $W = [-1, 30] \times [-3, 3]$ the peak current is limited to 3 Amperes. We synthesize a controller enforcing the "reach and stay while stay" specification in 108 seconds. The closed-loop simulation results in Figure 2 show that the target set is reached while the current ripple and peak values limitations are respected. For a more detailed version of this example and other illustrative examples of Pessoa capabilities we refer the reader to [Pes].

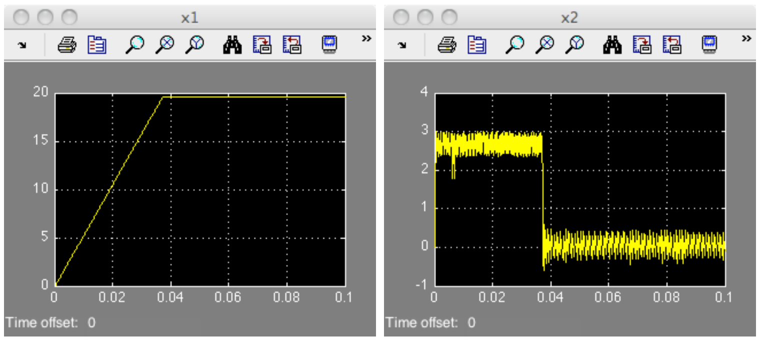


Fig. 2. Evolution of velocity (left) and current (right).

5 Conclusions and future work

Pessoa can be used to compute finite abstractions of continuous control systems, synthesize reactive controllers, and refine the synthesized controllers to Simulink blocks. Pessoa is currently being improved by extending its scope to natively support the abstraction of non-linear and switched continuous dynamics, allow full LTL specifications, multi-resolution quantization of the input and state spaces of the control system, support quantitative control objectives, and provide automatic code generation.

References

[Ari] Ariadne. `http://trac.parades.rm.cnr.it/ariadne/`.

[Be99] F. Balarin *et al.* Synthesis of software programs for embedded control applications. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 18(6):834–849, June 1999.

[Be07] Roderick Bloem *et al.* Specify, compile, run: Hardware from psl. *Electron. Notes Theor. Comput. Sci.*, 190(4):3–16, 2007.

[Che] Checkmate. `http://www.ece.cmu.edu/~webk/checkmate/`.

[CUD] Cudd. `http://vlsi.colorado.edu/~fabio/CUDD/`.

[Hyba] Hybridsal. `http://sal.csl.sri.com/hybridsal/`.

[Hybb] Hybrid Toolbox. `http://www.dii.unisi.it/hybrid/toolbox`.

[KeY] Keymaera. `http://symbolaris.com/info/KeYmaera.html`.

[LTL] LTLCon. `http://iasi.bu.edu/~software/LTL-control.htm`.

[Pes] Pessoa internal report. `http://sites.google.com/a/cyphylab.ee.ucla.edu/pessoa/publications/Pessoa.pdf`.

[PHA] Phaver. `http://www-artist.imag.fr/~frehse/phaver_web/index.html`.

[Tab09] Paulo Tabuada. *Verification and Control of Hybrid Systems*. Springer, 2009.

[Weg00] I. Wegener. Branching programs and binary decision diagrams - theory and applications. In *SIAM Monographs on Discrete Mathematics and Applications*, 2000.