

Automated Design of Self-Adaptive Software with Control-Theoretical Formal Guarantees*

Antonio Filieri^a, Henry Hoffmann^b, and Martina Maggio^c

^aUniversity of Stuttgart, Stuttgart, Germany

^bUniversity of Chicago, Chicago, IL, USA

^cLund University, Lund, Sweden

Abstract: Self-adaptation enables software to execute successfully in dynamic, unpredictable, and uncertain environments. However, most of the current approaches lack formal guarantees on the effectiveness and dependability of the adaptation mechanisms, limiting their applicability in practice.

Control theory established a broad set of mathematically grounded techniques for the control of dynamic systems for several engineering fields. While control shares self-evident similarities with software adaptation, modeling software behavior as a system of differential or difference equations is not straightforward, nor is mastering the mathematical background needed for synthesizing a suitable controller.

In this paper we focus on the automatic modeling and controller synthesis for systems with a single knob affecting the satisfaction of a quantitative requirements. Effectiveness and performance of the controller are guaranteed by construction. The approach is fully automated and implemented in several programming languages, empowering non-experts with the ability of applying control principles to a wide range of software adaptation problems.

The dynamic natures of modern systems require software to withstand sudden and unpredictable changes in its execution environment; e.g., application workload fluctuations and system component failure. The use of *autonomic* or *self-adaptive* techniques has been proposed to help engineers design software able to modify their own behavior to maintain goals in response to unpredicted changes. While adaptation of an application's *functional* aspects (i.e., semantic correctness) often requires human intervention, its *non-functional* aspects (such as reliability, performance, energy consumption, and cost) represent an important and challenging opportunity for applying self-adaptive techniques.

Non-functional aspects can be often mapped into specific *quantitative properties* that can be measured and used to trigger adaptations guaranteeing requirements are met even in the face of unforeseen environmental fluctuations [FGLM11]. Such measurement-driven adaptation has been studied for decades in the context of *control theory*. Control systems have achieved widespread usage in many engineering domains which interact with the physical world. In such systems, the controller measures quantitative feedback from a *sensor* (e.g., a speedometer), and determines how to tune an *actuator* (e.g., a fuel intake) to effect the controlled *plant* (e.g., an engine). One major advantage of using control theory is that such techniques emit analytical guarantees of the system's dynamic behavior.

*This paper reports a summary of [FHM14]. Please refer to the original paper for a complete exposition.

While researchers have applied some notions from control theory to software systems, there are many challenges that must be overcome to advance the application of control theory to software systems and many of these challenges arise from the difficulty of modeling the controlled systems [DHP⁺05, ZUW⁺09, FGLM11, FGLM12]. Specifically, software applications have complex, often non-linear, interactions with the hardware and system software that support their execution. This difficulty in defining concise and precise models of software behavior usually leads to the design of controllers focused on particular operating regions or conditions, or ad hoc solutions which address specific computing problems using control theory, but do not generalize [TAL08, LLA⁺06, HME10].

We address this need by presenting a methodology which automatically builds suitable system models and then uses those models to synthesize a controller suitable for the self-adaptive management of non-functional application requirements. Given a software system and a non-functional requirement (e.g., performance, accuracy, energy), our methodology first uses a training phase to generate a linear model of the system and then synthesizes a configurable controller. The controller overcomes potential non-linearities using a Kalman filter to adapt the linear model dynamically. In addition, for drastic changes in system behavior, the controller incorporates a change-point detection strategy to trigger an online model rebuilding phase. This methodology allows non-expert users to apply control theoretic techniques, without requiring a priori knowledge about system's behavior.

We evaluated our methodology in two ways. First, we performed a formal assessment of the guarantees it provides. Second, we performed an empirical assessment of the methodology on three different software applications: video compression, energy efficient resource provisioning, and dynamic binding and delegation. Details on the approach and its evaluation can be found in [FHM14].

References

- [DHP⁺05] Yixin Diao, J.L. Hellerstein, S. Parekh, R. Griffith, G. Kaiser, and D. Phung. Self-managing systems: a control theory foundation. ECBS, pages 441–448, Apr 2005.
- [FGLM11] Antonio Filieri, Carlo Ghezzi, Alberto Leva, and Martina Maggio. Self-adaptive software meets control theory: A preliminary approach supporting reliability requirements. ASE, pages 283–292. IEEE, 2011.
- [FGLM12] Antonio Filieri, Carlo Ghezzi, Alberto Leva, and Martina Maggio. Reliability-driven dynamic binding via feedback control. SEAMS, pages 43–52, Jun 2012.
- [FHM14] Antonio Filieri, Henry Hoffmann, and Martina Maggio. Automated Design of Self-adaptive Software with Control-theoretical Formal Guarantees. ICSE, pages 299–310. ACM, 2014.
- [HME10] Joseph L. Hellerstein, Vance Morrison, and Eric Eilebrecht. Applying Control Theory in the Real World: Experience with Building a Controller for the .NET Thread Pool. *SIGMETRICS Perform. Eval. Rev.*, 37(3):38–42, January 2010.
- [LLA⁺06] Chenyang Lu, Y. Lu, T.F. Abdelzaher, J.A. Stankovic, and Sang Hyuk Son. Feedback Control Architecture and Design Methodology for Service Delay Guarantees in Web Servers. *IEEE Trans. Parallel Distrib. Syst.*, 17(9):1014–1027, Sept 2006.
- [TAL08] Mara Tanelli, Danilo Ardagna, and Marco Lovera. LPV model identification for power management of Web service systems. CCA, pages 1171–1176, Sept 2008.
- [ZUW⁺09] Xiaoyun Zhu, Mustafa Uysal, Zhikui Wang, Sharad Singhal, Arif Merchant, Pradeep Padala, and Kang Shin. What Does Control Theory Bring to Systems Research? *SIGOPS Oper. Syst. Rev.*, 43(1):62–69, January 2009.