On a Control-Theoretic Approach for Proving Termination

Oliver Theel

Darmstadt University of Technology
Department of Computer Science
D-64283 Darmstadt, Germany

Phone: [49] (6151) 16-5306
Fax: [49] (6151) 16-5410
Email: `theel@informatik.tu-darmstadt.de`

Keywords: Termination, Self-Stabilization, Convergence, Control Theory, Ljapunov Equation, Ljapunov Function

Self-stabilization is a powerful property of a distributed algorithm: a self-stabilizing distributed algorithm is guaranteed to reach a pre-defined set of so-called *legitimate states* in finite time, regardless it its initial system state. Once in a legitimate state, a self-stabilizing distributed algorithm does not voluntarily leave the set of legitimate states. Only in case of faults, leading to a perturbation of the system state, the algorithm may switch to a so-called *illegitimate state*.[1] Being in an illegitimate state, the algorithm re-enters a legitimate state as soon as new faults cease to appear and errors in the system state as well as failed components have been corrected or repaired. Thus, self-stabilizing distributed algorithms tolerate – although in a non-masking manner – any number and distribution of *transient* faults [3]. Furthermore, self-stabilizing distributed algorithms do not need to be correctly initialized, since an arbitrary, improper initialization corresponds to an illegitimate state. From there, as described above, the algorithm will autonomously converge to a legitimate state. Due to these advantages, self-stabilizing distributed algorithms are subject to excessive research focusing on design as well as verification techniques.

The verification of a self-stabilizing distributed algorithm is generally done in two steps. In one step, *convergence* into the set of legitimate state in finite time must be shown. In another step, the *closure* of the set of legitimate states must be proven. Whereas proving closure is easy, proving convergence is much more challenging. In fact, proving convergence is an instance of proving termination, since for any self-stabilizing algorithm $\mathcal{A}$ an alternative algorithm $\mathcal{B}$ can be given that repeatedly executes $\mathcal{A}$'s program actions in a loop while a certain state predicate P is false. This state predicate evaluates to true if and only if the current state of the system is identical to a legitimate state. Consequently, through $\mathcal{B}$'s termination proof follows $\mathcal{A}$'s convergence proof and vice versa. Thus, all methods for proving termination and convergence that exist in self-stabilization and termination literature can be applied. However, the existing methods all boil down to a well-foundedness argument: One has to devise a system state function mapping the system state into a domain for which a total order among the entities can be given and where a smallest domain element with respect to the total ordering exists. Building on this function, the proof must show that while the system performs subsequent state transitions, the function value decreases with respect to the total ordering. This guarantees that after a finite number of state transitions the function value is minimal. If the corresponding system state is a legitimate state then convergence has been proven. Although it is obvious how to perform a convergence proof using such a termination function (which is called *convergence function* in self-stabilization literature), it is be no means easy to devise such a function since

[1] A system state is either a legitimate state or an illegitimate state.

it somehow captures "the very essence of convergence" of the algorithm. Techniques to systematically identify those functions are therefore highly looked for.

Self-stabilizing algorithms exhibit certain analogies with stable feedback systems used in electrical and mechanical engineering. In contrast to self-stabilizing algorithms which present a relatively young area of research in computer science, feedback systems research dates back for more than a century. Not surprisingly, feedback system theory offers a variety of methods for reasoning about system stability. The basic idea of our research is to investigate to what extend feedback system theory can be used for reasoning about the self-stabilization property of algorithms.

As a first result, we could identify a system model for so-called *discrete-time non-linear dynamic variable structure feedback systems* [5] that allows the modeling of self-stabilizing algorithms in terms of feedback system terminology. This system model can be seen as connector of two research areas: it bridges the gap between self-stabilizing algorithms and feedback systems. As a consequence, based on the model, a powerful criterion of control theory, known as *Ljapunov's "Second Method"* [1, 2] can directly be adopted for reasoning about system stability and, thus, for the reasoning about the self-stabilization property of a modeled algorithm. If a so-called *Ljapunov equation* for a given system can be solved then a convergence function (called *Ljapunov function*) is guaranteed to exists and is identified in the course of the calculation. Unfortunately, the Ljapunov equation cannot always straightforwardly be solved,[2] but it does simplify convergence function identification for many self-stabilizing algorithms. For instance, for self-stabilizing algorithms that can be modeled as *linear* time-discrete dynamic dynamic feedback systems (with and without variable structure), a standard starting point always solves the Ljapunov equation leading to an identification of a suitable Ljapunov function. But even in the non-linear case, Ljapunov theory offers many good starting points for solving the Ljapunov equation or for direct identification of a Ljapunov function.

Interestingly, it turned out that the transfer function approach (presented at WST'99, see [4] for details) seamlessly fits into our new approach: through the system model and Ljapunov theory we are now able to identify a termination function solely based on a given transfer function.

In the talk, we would like to present the control-theoretic approach for identifying termination functions as sketched in this abstract. We plan to illustrate the usefulness of the approach through a sample algorithm whose convergence property is to be proven. We will show how to model the algorithm as an instance of the system model and how to systematically identify a suitable Ljapunov function. Finally, we would like to elaborate on the potentials and limitations of the presented technique.

References

[1] R. E. Kalman and J. E. Bertram. Control System Analysis and Design Via the "Second Method" of Lyapunov - Part II: Discrete-Time Systems. *Transactions of the ASME, Journal of Basic Engineering*, 82:394–400, June 1960.

[2] M. A. Ljapunov. Problème général de la stabilité du movement. *Ann. Fac. Sci. Toulouse*, 9:203–474, 1907.

[3] M. Schneider. Self-stabilization. *ACM Computing Surveys*, 25(1):45–67, 1993.

[4] O. Theel and F. C. Gärtner. On Proving Convergence through Transfer Functions (Extended Abstract). In Jürgen Giesl, editor, *Proc. of the 4th International Workshop on Termination (WST'99), Dagstuhl, Germany*, pages 46–47, May 1999.

[5] V I. Utkin. Variable Structure Systems with Sliding Modes. *IEEE Transactions on Automatic Control*, 22:212–222, 1977.

[2] E.g., for instable systems the equation can, of course, never be solved, since the existence of a Ljapunov function implies system stability.