Cross-Application Perspectives: Tools and Platforms for Control Systems

Ken Butts and Andras Varga

Introduction

Control analysis and design methods are based on rigorous theoretic and systematic foundations. Regardless of application, the concepts of stability, controllability, observability, performance, and robustness are used to analyze the system or device to be controlled and design the control laws that satisfy the given system requirements. Due to the engineering nature of the discipline, control designers have enjoyed a rich history of simultaneous development of theoretical concepts and supporting numerical algorithms. These numerical algorithms are the basis for control-domain-specific computer-aided engineering toolkits that provide ready access to best practices, rigor, and scalability. Although the number of users for this tooling is relatively small, control engineering's fundamental, application-agnostic approach yields a vibrant and sustainable tool market.

Control-oriented software platforms represent integrated software environments whose tool chains cover most aspects of typical industrial control design workflows. One such workflow is the well-known system engineering "V" diagram of Fig. 1, where the requirements are processed through design and implementation with a focus on step-by-step process deliverables and milestones.

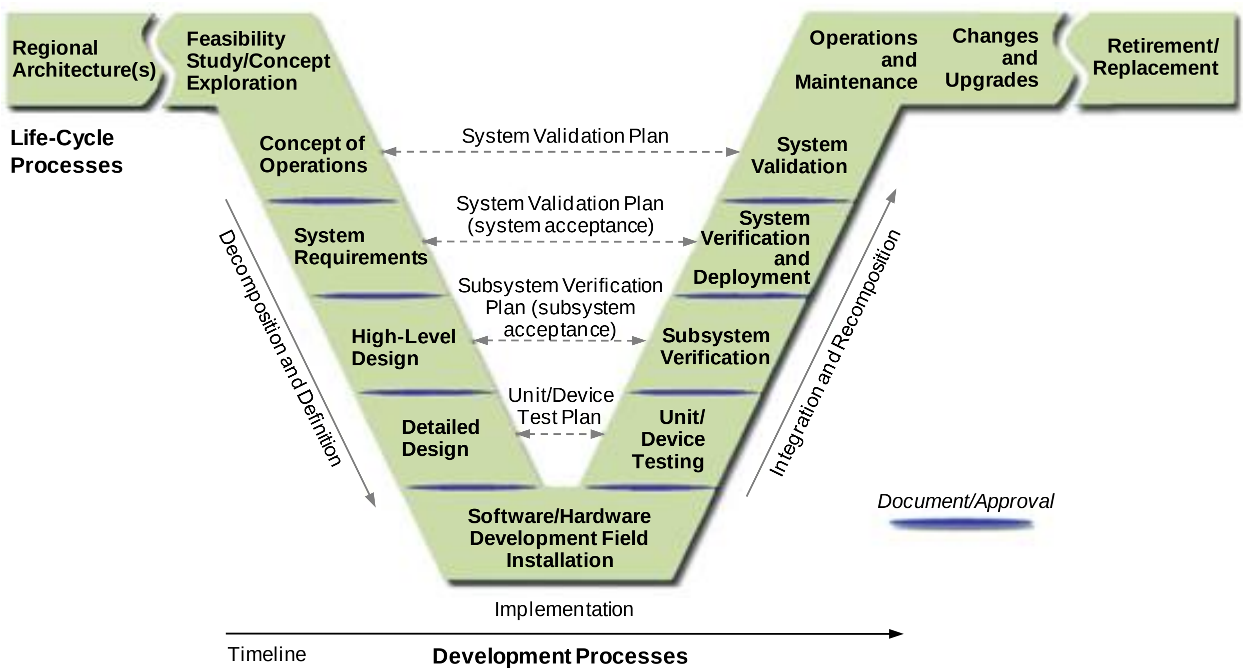
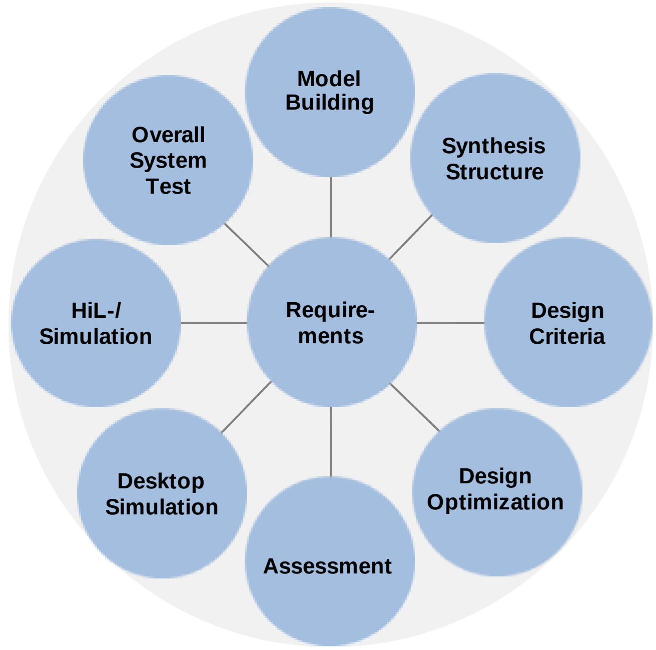


Figure 1. Systems engineering V diagram [1].

From: *The Impact of Control Technology*, T. Samad and A.M. Annaswamy (eds.), 2011. Available at www.ieeecss.org.

Alternatively, a control engineering task view is shown in Fig. 2. From this perspective, one can see how an effective software platform supports requirements-centered control engineering iteration, where results are validated against the requirements and the requirements are occasionally updated to satisfy design tradeoffs. Typical commercial platforms provide tool chains with rich functionality for standard tasks such as control-oriented plant model building, control system design, or simulation-based assessment while simultaneously providing support to related aspects such as rapid prototyping, software/ hardware-in-the-loop verification, or code generation. The widespread use of these platforms in industry and academia confirms the needs for a broad spectrum of tools supporting all aspects of control systems development. Several success stories provide evidence that without such platforms, many advanced control applications, especially in the aerospace, automotive, and process industry areas, would not be possible.



Source: DLR

Figure 2. Model-based control system design.

Control systems are becoming ubiquitous, more critical, more heterogeneous, and more distributed. These systems must be developed reliably, predictably, and productively. Consider the statement by the U.S. President's Council of Advisors on Science and Technology: "It is difficult to overstate the contribution of networking and information technology (NIT) to America's security, economy, and quality of life. This contribution is the consequence of rapid advances in an array of technologies, some now ubiquitous, such as Internet search engines and wireless devices. Other technologies, such as simulation software and embedded systems, are essential to the effective performance of sectors that include national security, energy, health care, manufacturing, and transportation. The cumulative effect of these technologies on life in the United States and around the world has been profound and beneficial" [2]. Control engineering tools and platforms are used to aid modeling and analysis, address large-scale systems, reduce time-to-market, and mitigate the limits of ad hoc and manual design processes.

Control engineering tools and platforms are used to aid modeling and analysis, reduce time-to-market, and overcome the limits of ad hoc and manual design processes.

Specific Industrial Needs

Although the fundamentals of control engineering are common across application domains, some variations in approach can be identified. For example, model-building tools for the aerospace and automotive industries must primarily support physically oriented modeling, whereas for the process industries, due to the higher complexity of plant dynamics, model-building tools must use a data-driven plant modeling approach. The widely used block-diagram-based (also called *causal*) model-building approach has well-known intrinsic limitations, and therefore, in some domains (such as aerospace and automotive), object-oriented (also sometimes called *acausal*) modeling techniques are gaining greater acceptance, with obvious benefits of more physical insight and better reusability of model components.

Controller tuning philosophies also vary markedly across the aerospace, automotive, and process control industries. The control designer is solely responsible for parameter tuning in the aerospace industry, whereas in the automotive industry, downstream product specialists typically set the tuning parameters to meet vehicle performance targets. Auto-tuning algorithms are often used in the process industries to allow plant control engineers to perform control law tuning on site.

The control designer is solely responsible for parameter tuning in the aerospace industry, whereas in the automotive industry, downstream product specialists typically set the tuning parameters to meet vehicle performance targets. Auto-tuning algorithms are often used in the process industries to allow plant control engineers to perform control law tuning on site.

Assessment needs also vary significantly among activity domains. Although simulation-based assessment (also in conjunction with Monte Carlo techniques) is widely used across many industries, the assessment of safety-critical control applications (such as flight control) requires rigorous verification of the robustness of control laws in the presence of uncertainties. Since the verification effort can represent more than 50% of the total development cost of the control system, new optimization-based approaches (such as worst-case search) in conjunction with parallel computation techniques can contribute greatly to cost reduction.

Automatic code generation is widely used for rapid prototyping, especially in the automotive and aerospace industries. In particular, the requirements for code generation in the aerospace industry are more stringent because certifiable code imposes the use of certified code-generation tools as well, which are usually not part of common control design platforms. The gap in the tool chain may even necessitate manual recoding of control laws. No single platform can presently address the entire development workflow in aerospace.

Furthermore, functional safety standards (see [3] for a draft version) establishing development requirements for automotive safety-critical systems will result in demands for qualified processes and tooling. It is imperative that the automotive control design community properly manage the adoption of such standards for systems, hardware, and software development.

To make the general workflow situation slightly more concrete, we list a representative tool suite for automotive control system development:

- Requirements management: IBM Telelogic DOORS®.
- Plant modeling:
  - Acausal: Modelica Association Modelica®;
  - Causal: MathWorks Simulink®;
  - Empirical: MathWorks Model-Based Calibration Toolbox™;
  - Parameter identification: MathWorks System Identification Toolbox™.

- Control design:
  - Analysis and synthesis: MathWorks Control System Toolbox™;
  - Algorithm specification: MathWorks Simulink®;
  - Automatic code generation: dSPACE TargetLink.
- Verification and validation:
  - Model-in-the-Loop, Rapid Controller Prototyping, Software-in-the-Loop, Processor-in-the-Loop, Hardware-in-the-Loop: dSPACE;
  - Automatic Test Generation: Reactive Systems Reactis®.
- Calibration:
  - Design of experiments: MathWorks Model-Based Calibration Toolbox™;
  - Test automation: A&D Technology ORION;
  - Data acquisition, visualization, and analysis: ETAS INCA.

Opportunities for Research and Education

Given the dynamic nature of the development of control theory, gaps naturally exist between theory, design methodologies, and supporting tools. To reduce these gaps, many opportunities exist for improving tooling, including the following:

- Support for cyberphysical systems and networked embedded control: The rapid advancements in control, computing, communication, and the physical sciences enable system designs that are beyond our ability to analyze and verify. Theoretical system research in the cyberphysical systems and networked embedded control domains is expected to lead to systematic design methodologies and thence to the development of appropriate tools.

- Architectural analysis and design: Control systems must provide cross-cutting qualities such as function, safety, reliability, security, and energy efficiency. Architecture-based annotation and abstraction techniques allow system designers to model, assess, and confirm these qualities for large-scale, component-based systems [4]. The deployment of new instructional modules would accelerate the adoption of this emerging system engineering discipline, and research on new system verification methods that use these annotations and abstractions will greatly enhance system designers' capabilities and capacities.

- Productivity increase: A significant increase in the efficiency of control system development can be expected in different industries by developing and using adequate tools. Some examples are: control-oriented physical plant modeling (all domains); efficient certification/qualification tools (aerospace/automotive); certifiable/qualifiable autocode from control design tools (aerospace/automotive); verification and validation tools allowing automatic test generation at the system level or for requirements coverage (all domains); system identification tools for use in the field (process industries). With the advent of cheap multicore/cluster computing architectures, the use of parallel computational techniques will be an important facilitator of productivity increases.

- Numerical algorithms: System-theoretic algorithms will continue to form the core of computer-aided control systems design (CACSD). Numerical algorithms and accompanying software tools developed for new control domains often represent enabling technologies for the applicability of advanced control techniques. From an educational perspective, more emphasis needs to be placed on numerical algorithms in control engineering curricula.

Additional Workshop Participants

The authors would like to thank the following workshop participants for their contributions to this section: Maryam Khanbaghi (Corning), Alexander Knoll (EADS), Massimo Maroni (Alenia Aermacchi), Johann Bals (DLR), Dragan Obradovic (Siemens), Thomas Bak (Aalborg University), Luigi Glielmo (Università del Sannio in Benevento), and Anuradha Annaswamy (MIT) .

References

[1] Federal Highway Administration, U.S. Department of Transportation. "Systems Engineering for Intelligent Transportation Systems" [Online], Publication No. FHWA-HOP-07-069 EDL-14340, January 2007, Available at http://ops.fhwa.dot.gov/publications/seitsguide/section3.htm.

[2] President's Council of Advisors on Science and Technology. "Leadership Under Challenge: Information Technology R&D in a Competitive World: An Assessment of the Federal Networking and Information Technology R&D Program" [Online], August 2007. Available at http://www.nitrd.gov/pcast/reports/PCAST-NIT-FINAL.pdf.

[3] International Standards Organization. ISO/DIS 26262-(1 to 10), ISO Standards, ICS, 43 Road vehicles engineering, 43.040.10 [Online], 2009, Under development, Draft available at http://www.iso.org/iso/home.htm.

[4] Society of Automotive Engineers. SAE Architecture Analysis & Design Language (AADL), SAE Standard AS-5506 [Online], January 20, 2009, Available at www.aadl.info.