Real-time software for versatility, scalability and reconfigurability in complex embedded feedback control systems

Prof. Jan Wikander[1], Prof. Henrik I Christensen[2], Dr Martin Törngren[1]
[1]Department of Machine Design
[2]Center for Autonomous Systems (CAS) and
Department of Numerical Analysis and Computing Science
Royal Institute of Technology, 100 44 Stockholm Sweden

1. Summary

The proposed project is a collaboration between Centre for Autonomous Systems (CAS), ARTES and Arcticus Systems AB. The goal of the project is to provide a software architecture suitable for scalable, maintainable and reconfigurable mechatronic systems controlled by an embedded distributed computer system. The project is focused on multi-degree-of-freedom mechanical systems of a complex configuration, found in e.g. legged locomotion systems. The specific characteristics of the software architecture are its ability to support; hierarchies of control levels; extensive mode shifting; predictable real-time execution, synchronization and communication; reconfiguration including scaling of the mechanical system and finally partitioning and allocation of software functions for close to optimal resource utilisation. The project involves, on an equal share basis between ARTES and CAS, two PhD students over 2+2 years, with a corresponding total budget 2390 + 2607 = 4997 KSEK.

2. Problem statement

The proposed project is aimed at finding solutions in terms of software architectures which support the design, maintenance and scaling of real-time computer controlled mechanical systems and machines, especially in the context of autonomous systems. Such architectures should support:

- building tailored systems from components,
- maintaining such systems over extended periods of time e.g. by replacing components or by changing/adding functionality through software,
- scaling the system in terms of number of components, e.g. number of mechanical degrees of freedom,
- distributed systems through partitioning and allocation of software for close to optimal resource utilisation,
- hierarchies of control levels corresponding typically also to hierarchies of sample rates,
- extensive mode shifting inherent in the control design, i.e. when the dynamic models and thereby the corresponding control algorithms change extensively during normal operation,
- reconfiguration of the mechanical system; this could in its ultimate form facilitate autonomous reconfiguration of the machine to adapt to new tasks.

2.1. Research problem in the context of autonomous systems

The proposed project is an integrated part of one out of two major research themes within CAS, namely autonomous systems for difficult terrain. This theme is initially aimed at construction of locomotion systems that can be used for efficient mobility and navigation in a difficult terrain. The major hypothesis is; *four legged locomotion is outstanding for smooth, reasonably fast, versatile, cautious and efficient locomotion, on a scale similar to humans, in difficult terrain*. Although robots with legged locomotion provide excellent mobility, they constitute extremely complex systems in terms mechanics, degrees of freedom, dynamic models, actuation, sensing, control algorithms, computer systems, communication and systems integration. This problem of complexity and the fact that most of the functionality is implemented by software, is the major motivation for the proposed project. Major goals for the whole difficult terrain research activity in relation to the project proposed here are in short:

- Robust control methods/algorithms for static and dynamic walking in difficult terrain.
- Control architectures for extremely complex mechanical systems with respect to the multi-objective situations appearing in difficult terrain (balance, disturbance rejection, turning, moving forward etc.).
- Software/hardware which support modularity, scalability, reconfigurability and predictable timing behaviour.
- Internal state estimation algorithms considering a large number of sensors, very complex dynamic models and the characteristics of the large and heterogeneous computer control system.

A unique characteristic of the overall research work is the mechatronic perspective involving engineers, students and researchers with a background in different disciplines; mechanical engineering, electrical engineering, computer science and automatic control. A general goal is that the technical research achievements should be generic in the sense of having applicability in a number of engineering contexts other than autonomous systems.

3. Main ideas and approach

In mechanical systems and machines the implementation of new and improved functionality is gradually moving from mechanics to software. Due to the power of software the new designs become versatile but at the same time also very complex. The typical attempt to handle the complexity is by modularisation which in this type of applications means modularisation in mechanics, electronics and software. This leads to distributed and decentralized systems potentially allowing systems to be built from components. This project is about developing a software architecture that supports development and maintenance of such systems. In the following the main ideas and approach are outlined.

Control problem. There is a multitude of more or less integrated control related requirements which must be concurrently adhered to. On system locomotion level these requirements typically may be formulated in terms of functions like propelling, braking, turning, reversal, posture, balance, locomotion smoothness, tracking, force distribution, terrain disturbance rejection, emergency handling e.g. when stumbling, etc. We may regard these functions as locomotion behaviours which need to be integrated by means of a control architecture. Another important characteristic of the control problem is the extensive mode shifting which occurs as the dynamic model of the vehicle change during operation. The event that a foot hits the ground leads to such model alteration and mode shifting.

Distributed system. The complexity of a legged robot is mainly due to the large number of degrees of freedom. The four legged system considered here has 12 actuated joints. The amount of transducers (sensors and actuators) requires that special emphasis is put on signal transmission with respect to signal quality, timing and physical transmission. In the chosen hardware architecture, local control or signal processing units are located close to the transducers to limit the amount analog wiring in favour of serial communication over a field bus, in this case a CAN bus. The computer system architecture, when further developed, is expected to be heterogeneous both in communication and execution.

Modelling. The complexity and broadness of the control problem in combination with the distributed and heterogeneous computer system requires means for modelling. Further, any software architecture has to have a basis in a modelling framework capable of capturing the characteristics of the control design and of the computer implementation. Such a modelling framework, which supports modelling of both hardware and software in distributed real-time computer implementations of control systems, exists (the AIDA models, [1] Redell (1998). The models describe mainly the control design and the computer hardware and software, and are intended to be used together with existing computer aided control engineering tools. Important features of the AIDA models include specification of timing and triggering in a way that especially supports multi-rate feedback control systems. The models have been used to model parts of the four legged vehicle as a case study [2] (Redell 1998 b) to evaluate the functionality and usefulness of the models, the outcome of the case study is very satisfactory. The AIDA models will be used and further developed as a basis for describing and evaluating different candidate software architectures.

Structuring. Structuring is usually referred to as the activity of obtaining a suitable organization of a system. More precisely this refers to how a system is decomposed into components (functions/objects) and how these components are related. Structuring is thus a design activity where the approach taken has a large influence on several essential systems properties such as timing and dependability. For a motion control application to be implemented in a distributed computer system, structuring thus represents a fundamental problem characterized by several, sometimes conflicting constraints, see e.g [3] Törngren and Wikander (1996). Clearly, application or system structuring is strongly related to software architecture. The architecture of a software system is defined by the computational components and interactions (data, precedence, time) among those components. The definition of the components is directly linked to control design and structuring of the overall application. System requirements are sometimes subdivided into functional and non-functional ones. The functional aspects correspond to determining a suitable structure from an implementation independent view. As opposed to this, an implementation related structure is often described using entities such as operating system tasks processes and inter-process communication. Since both functional and non-functional characteristics are captured by the modelling framework outlined above, the main idea here is to base the work on the software architecture on thorough structural considerations supported by the AIDA models.

Research approach. The research approach is summarized in the below points.

- Establishment of a requirements specification for the architecture. This task is performed strongly linked to analysis of the control requirements (as formulated at the corresponding point in time) as well as non-functional platform requirements. **Deliverable**: Specification.
- A state of the art (SOTA) study on software architectures for real-time feedback control systems. This study includes definition of concepts and notation (e.g. what is an architecture?, which are the problems?, etc.). **Deliverable**: Report.
- Evaluation (without any implementation or at least only minor implementation work) of existing relevant software architecture concepts with respect to the specification. This task includes the establishment of an evaluation procedure. Here the AIDA modelling framework is expected to very useful. **Deliverable**: Evaluation report.
- Development of a first version of a software architecture closely linked to the control software implementation on the DTD. This task is strongly liked to structuring and mapping of the control application. **Deliverables**: Architecture specification (ver 1) and structuring guidelines
- Experimental verification. The results of the research will form a basis for the software implementation on the Difficult Terrain Demonstrator. This demonstrator, and future versions of it, is expected to used as a research platform over an extended period of time, at least a decade. Grow and change aspects will be especially important as the functionality of the DTD will be continuously improved (by adding, replacing and reconfiguring real-time

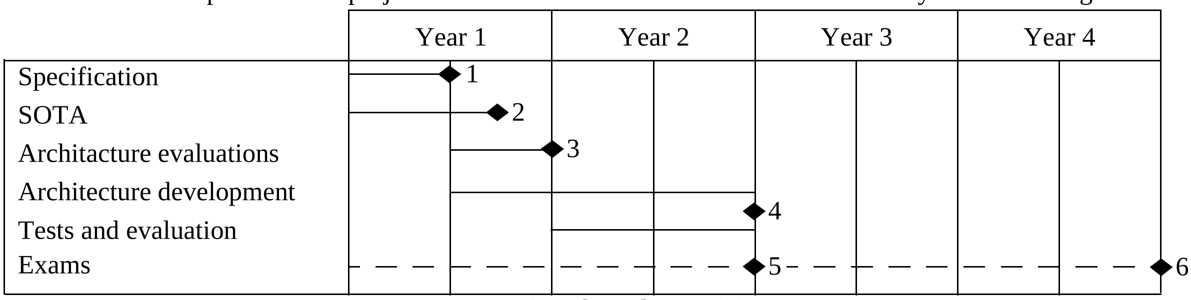
software components) through contributions from several generations of doctoral students and researchers. The DTD will thus serve as a good mean for thorough verification and further development of the software architecture.

4. Expected results and impact (in order of priority)

- A software architecture suitable for the considered type of applications and governed by modularity, scalability, dependability, testability, resource utilisation, reconfigurability, mode shifting and control hierarchies. The architecture will be implemented and evaluated on completely self-contained four legged robot *Sleipner 3*.
- Two well educated systems oriented licentiates and later PhDs ready for professional contributions in industry.
- A further developed and evaluated modelling framework (AIDA models provide the basis) suitable for modelling a complex control design and its implementation on a heterogeneous distributed computer system.
- A methodology and corresponding guidelines for the structuring of the considered type of control applications (machine embedded complex control systems). The structuring guidelines will be governed by the same set of x-abilities as discussed in the first point.
- The modelling framework, the structuring guidelines and software architecture are expected to be directly applicable in industry after the four year project.

5. Project plan

The project plan is divided into tasks and (major) milestones for the first two years. The platform is expected to undergo significant changes over the next two years and the relevant candidate will also be given an opportunity to influence the final emphasis of the project. This fits well with the ARTES model of 2+2 years of funding.



Milestones: 1) Specification
2) SOTA report
3) Evaluation report
4) Technical report(s)
5) 2 licentiate theses
6) 2 doctoral theses

6. Preliminary budget

Costs	Year 1	Year 2	Year 3	Year 4	Grand total
Student salary	590	610	660	680	2540
Experimental	110	110	110	110	440
Others	67	67	67	67	268
Dept. OH	142	145	155	158	600
Premises	130	133	142	145	550
KTH OH	142	145	155	158	600
Total	1180	1210	1288	1318	4997

Funding	Year 1	Year 2	Year 3	Year 4	Grand total
CAS reserved	590	605	644	659	2498
ARTES	590	605	644	659	2498
Total	1180	1210	1288	1318	4996

7. Related research

From a mechatronics/robotics perspective the most relevant work is the work on modular manipulator systems going on at CMU. The Reconfigurable Modular Manipulator System (RMMS) constitutes a rapidly deployable manipulator system being modular in both hardware [4] (Paredis et al. 1996) (mechanical and electrical) and software [5] (Stewart et al. (1995)). The control software for the RMMS uses the Chimera real-time operating system which supports reconfigurable and reusable software components. Another very interesting system architecture, providing upgradeability even on the fly, is the SIMPLEX architecture, [6] Sha et al. (1996). This architecture with its focus also on dependability, has a unique connection also to control engineering providing a framework for designing analytically redundant control software, generalized also to distributed systems [7] Sha (1997). In terms of open control systems for machine control, allowing machine builders/users to build/evolve systems from components supplied from different vendors, the most recent developments are summarized in [8] (Koren and Pritschow (1998)). A major activity in open system architectures is the OSACA project (EU - Esprit) in which the Mechatronics Lab at KTH has been a partner. The basis of OSACA is the software platform consisting of the reference architecture, the communication system and the configuration system. The platform supports interoperability, portability, scaleability and interchangeability. Another general framework for the design of distributed real-time systems is described in [9], Eriksson (1997). An important result from this work is an object model for hierarchical decomposition of application functionality and the semantics for communication and synchronisation. The mapping from the model to a distributed resource structure is supported by a set of prototypical tools.

8. Relation to the ARTES profile

In terms of aspects/topics of the ARTES profile defined in the current CFP this project is strongly related to early analysis of architectural choices, grow and change aspects, resource handling, distributed systems and implementation of control systems.

9. Industrial relevance

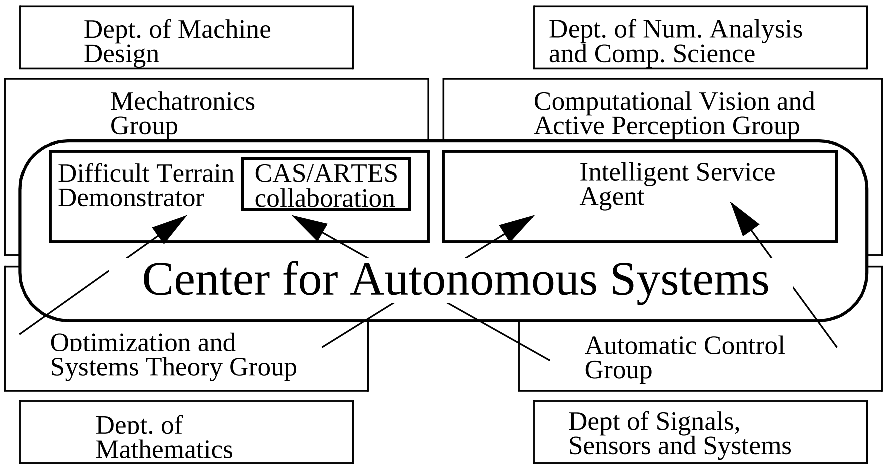
As previously discussed, the implementation of new and improved functionality inmachines is gradually moving from mechanics to software.This is a very clear trend in most branches of industry; automotive, industrial automation, mobile robots, machine tools, aircraft etc. Some key characteristics of applications found in these branches are; safety criticality, feedback control loops, physical integration/embedding of controls in the machine, a mix of time and event driven functionality and heterogeneous resources for both computation and communication. There is an urgent need in industry for theory, methodology and tools which support the design of this type of systems. A scientific foundation for how to design real-time software architectures based on structuring and modelling of application *and* implementation is an essential part of this industrial need.

10. Relation to other SSF programs

The proposed project is a direct cooperation on an equal share basis between the ARTES program and the SSF funded Center for Autonomous Systems (http://www.cas.kth.se). CAS research is organised around two major demonstrators; the Intelligent Service Agent and the Difficult Terrain Demonstrator (DTD). The latter, which is the focus of this project application, has been outlined above. Currently there are 4 full time and 2 half time doctoral students doing research directly applicable to the DTD. Funding has been reserved for two more full time students of which one constitutes the CAS share of this application. In total, CAS engages about 20 research students and additional senior staff. The ARTES student and the corresponding CAS student of this project will sign up as ARTES doctoral students and thereby take part also in the various networking activities of ARTES.

11. Context

The below figure gives an overview of CAS in terms of research groups and demonstrators.



As CAS is a cooperation among several departments the DTD doctoral students represent different departments; Machine Design, Mathematics and S^3 (Signals, Sensors and Systems). Thus, the students of this application will be part of a focused research group of critical size and of an important interdisciplinary mix. Due to the substantial experimental part of the DTD a technician (MSc) has been hired to be in charge of the experimental robot(s). In addition a PhD is being recruited to be full time project leader of the DTD.

Complementary activities and funding. The funding of the research group centered around the DTD as described above amounts to approximately 2.6 MSEK per year (doctoral students) excluding the students considered in this application. I addition to this the experimental budget amounts to 350 KSEK for 1998. The project is long term. Funding for technician and project leader is not included in the above.

Other related projects carried at the Mechatronics Group in the area of real-time systems are:

- DICOSMOS - Distributed control of safety critical mechanical systems. The current research focus is 1) Timing problems in real- time control systems, 2) Model based structuring and fault tolerance, 3) Event triggered sampling, 4) Investigation and evaluation of cost-effective communication services, 5) Fail-silent computer nodes and Fail-silent communication systems. The project, which is funded by NUTEK, is a cooperation between Computer Engineering at CTH, Automatic Control at LTH and Mechatronics at KTH.The funding at KTH/Mechatronics amounts to 480 KSEK for 1998.
- DOORS - Distributed object oriented real-time systems. The focus is on real-time control software and its verification and validation. The overall goals are to provide a methodology for handling requirements in all phases of development; for analysis of software safety including the impact of different software architectures; and finally for handling of different classes of timing requirements and the derivation of those from an application level. The project, which is funded by NUTEK, is a cooperation between Mechatronics at KTH, Industrial Control Systems at KTH and Computer Engineering at MDH. The funding at KTH/Mechatronics amounts to 427 KSEK for 1998.
- AIDA - Automatic control in distributed applications. This is the project in which the AIDA models, as discussed above, have been developed. In addition to the models the project aims at tool prototypes for design and early implementation analysis of distributed feedback control systems. The project, which is funded by NUTEK and Branschgruppen Datorstyrd Mekanik, is done in cooperation with ABB Robotics, Scania, SAAB Military Aircraft and Atlas Copco Controls.
- OSACA - Open system architectures for controls within automation. OSACA is a joint European initiative which aims to improve the competitiveness of the manufacturers of machine tools and control systems. The main goal is to specify a manufacturer independent system architecture for open control systems. The major result is a specification and supporting source code for a reference architecture, a configurations system and a communication system. An OSACA handbook is available and additional information is readily available on http://www.osaca.org. The funding at KTH/Mechatronics amounts to 300 KSEK for the first half of 1998, the project ends 980630.

Research cooperation. That fact that the proposed project has a strong component of research cooperation should be clear from the above.

Industrial cooperation. The project is carried out in cooperation with Arcticus Systems AB. This cooperation is based on the fact that the real-time operating system Rubus and accompanying tools in the Rubus Product Line, have been selected for the robot system. This means that the very first and early version of the computer system includes a Rubus real-time operating system in each processor node. See the supporting letter. In addition to this, CAS in general has cooperation with other companies like ABB Industrial Systems, Celsius Tech, Permobil, Electrolux, Weda Poolcleaners and more. Also the Mechatronics Group has an established cooperation with Branschgruppen Datorstyrd Mekanik.

12. References

[1] Redell, O., (1998). Modelling of Distributed Real-Time Control Systems, An approach for Design and Early Analysis. Licentiate Thesis, KTH, Machine Design, TRITA-MMK 1998:9, ISSN 400-1179, ISRN KTH/MMK--98/9--SE.

[2] Redell, O., (1998). Modelling and Implementation Analysis of the Distributed Real-Time Control System for a Four Legged Vehicle. Technical Report, KTH, Machine Design, TRITA-MMK 1998:8, ISSN 400-1179, ISRN KTH/MMK--98/8--SE.

[3] Törngren, M. and Wikander, J., (1996). A Decentralization Methodology for Real-Time Control Applications. Journal of Control Engineering Practice, Vol. 4, No 2, pp. 219-228, 1996. Elsiveirer Science.

[4] Paredis, C., Brown, B., Khosla, P., (1996). A Rapidly Deployable Manipulator System. Proc. of the 1996 IEEE International Conference on Robotics and Automation, Minneapolis, Minnesota, April 1996.

[5] Stewart, D., Khosla, P. (1995) Rapid Development of Robotic Applications Using Component Based Real-Time Software. Proc. of the 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems, Pittsburgh, PA, August 5-9 1995.

[6] Sha, L., Rajkumar, R., Gagliardi, M. (1996). Evolving Dependable Real Time Systems. The Proc. of IEEE Aerospace Conference 1996.

[7] Sha, L. (1997). Shifting the Computation Paradigm of Real-Time Control Systems. In SNART’97 Preprints, Lund University of Technology, Sweden, August 21-22, 1997.

[8] Koren, Y., and Pritschow, G. (Ed.) (1998). Open Architecture Control Systems - Summary of Global Activity. ITIA Series, Vol. 2. Published by ITIA - Institute for Industrial Technologies and Automation

[9] Eriksson, C. (1997). A Framework for the Design of Distributed Real-Time Systems. Doctoral Thesis, KTH, Machine Design, TRITA-MMK 1997:2, ISSN 400-1179, ISRN KTH/MMK--97/2--SE.

13. Appendicies

1) Short CV of Jan Wikander
2) Short CV of Henrik Christensen
3) Supporting letter from Arcticus Systems