# Automated Simulation and Configuration of Attitude and Orbit Control and Thermal Subsystems for Reconfigurable Satellites

T. Meschede*, J. Rießelmann*

*Department of Astronautics, TU Berlin, Germany
e-mail: Thomas.Meschede@tu-berlin.de
e-mail: Jens.Riesselmann@tu-berlin.de

## Abstract

Space debris and cost reduction are two important challenges for todays space industry. One possible concept to handle these topics is to develop serviceable and reconfigurable space systems. The modularization of space systems would be a major step towards that goal, as it makes replacing or upgrading components in on-orbit servicing missions feasible. This paper describes the early development of a modular attitude and orbit control subsystem (AOCS) and thermal control subsystem (TCS) for modular, (self-)reconfigurable satellites at the TU Berlin. First, a basic modular architecture for these two systems is presented. After that, a newly developed simulation architecture and optimization tools are described, which are being used in the development of the modular system. Finally, some first results are presented

## 1 Introduction and motivation

In recent years, space debris has become an important issue, that has a strong influence on the research of space technologies. Especially on-orbit servicing (OOS) is an attractive solution in the fight against a further increase of debris. In addition, it has the potential to lower the cost of future space missions by standardization and extending the lifetime of orbital systems.

On-orbit servicing will likely be carried out by robotic spacecraft and several systems are being developed to demonstrate the feasibility [1, 2]. As a consequence, components and subsystems of spacecraft, which are going to be serviced by a robotic spacecraft, have to be exchangeable in an easy way. For this reason, a high degree of modularity is a key element for on-orbit servicing.

An attractive concept would be the modularization of individual components [3, 4, 5]. Robotic replacement of these components could be realized by exchanging standardized modules. This would enable life extension and the ability to re-purpose entire space missions (figure 1).

In a modularized spacecraft, sensors, actuators and data management systems are distributed over the module assembly and information is being exchanged over a network. Logical and physical locations of all components have to be detected for the system to adapt to changing configurations. There already exist modular spacecraft like the international space station, which undergo reconfiguration changes for example during assembly and when the spacecraft are docking [6, 7]. But they do not automatically change sensor and actuator configurations and also lack automatically generated context-based information as for example their location and orientation.

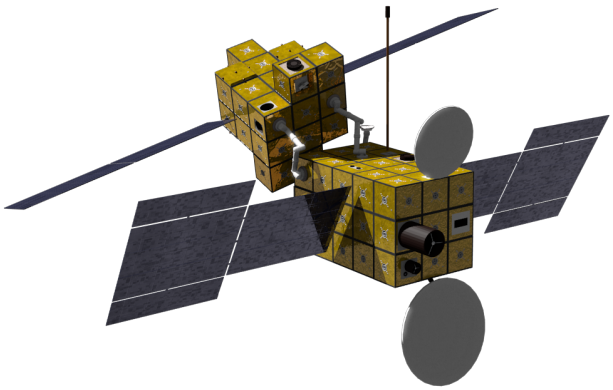

**Figure 1. Example of modular satellites**

For this reason, conventional subsystems have to be adapted to the needs of a serviceable, modular and (self-)reconfigurable spacecraft and new concepts of modular and distributed control methods for these subsystems have to be developed.

## 2 Modular spacecraft

The development of modular spacecraft implies the definition of a detailed catalog of modules, which can be used to create a set of reference satellites. In order for the reference missions to fulfill their requirements the catalog has to be carefully tuned.

In our studies we have identified modularization on component level as the preferred method to build reconfigurable spacecraft. The developed modules have a standardized geometry and can therefore be dynamically removed and attached at different positions to create new configurations [3]. This has consequences for the subsys-

tems of a spacecraft: As most subsystems rely on multiple components which are now spread over different modules, they have to adapt to the distributed and decentralized character of such a system.

As a result, properties of modular spacecraft and their subsystems are strongly influenced by the properties of the various interfaces used to connect the modules. For example the data interface creates delay times for data packets and the thermal interface allows only a certain amount of heat to be transferred between modules. Additional electrical and mechanical interfaces are also required. Whereas the mechanical, electrical and thermal interface primarily have an influence only on their respective subsystem counterparts (Attitude and Orbit Control System, Electrical Subsystem, Thermal Subsystem), the data interface is used by all of the subsystems and software across the spacecraft. For this reason, the data handling subsystem has a high influence on the architecture of almost all control subsystems of a modular space system.

## 2.1 Data handling subsystem and messaging patterns

As stated before, the data handling subsystem is a critical part in the development of modular systems. Reasons for this are that the amount of information and delay times can increase exponentially with the number of modules used (depending on the messaging pattern). Despite of this, certain real time requirements still have to be fulfilled. The real time capability of the data handling subsystem has for example a direct influence on the accuracy of the AOCS and this has an influence on measurements taken with high accuracy sensors and the quality of image data.

First hints on an architecture can be derived from principles of modular robotics. Most robots make use of modular communication frameworks following a publish-subscribe *messaging pattern* [8, 9]. Others go a step further and try to employ more decentralized messaging patterns [10, 11, 12]. These decentralized methods usually reduce the complexity of the data handling subsystem, but the complexity of the individual sensors and actuators increases as they have to handle more decisions autonomously. From the study of these modular systems three messaging patterns have been identified, which promise some advantages over classical systems, but also introduce some drawbacks:

**publish/subscribe** (figure 2 a, [9])
- ++ frameworks available
- ++ easy software writing
- + extensive use in robotics
- + scalability
- - complex bidirectional communication
- - real time requirements

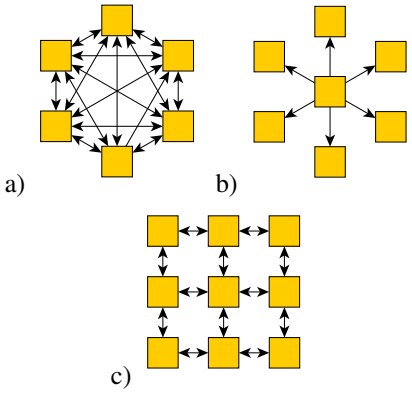


**Figure 2. a) Publish/subscribe network. b) one-way/virtual hormones network. c) self organizing network.**

**one-way** (figure 2 b, [10])
- ++ scalability
- + real time requirements
- + simple communication
- - redundancy
- - not widely adopted yet
- - complex hardware

**self-organizing** (figure 2 c, [12])
- ++ scalability
- + redundancy
- + simple communication
- + simple hardware
- - difficult communication algorithms
- - not widely adopted yet
- - real time requirements

The different requirements of some subsystem tasks make it reasonable to choose different messaging patterns for different tasks. This does not necessarily mean to change the underlying physical layer of the network. The publish/subscribe pattern could for example be the right choice for the attitude control subsystem and the self-organizing pattern could be suitable for the thermal control subsystem.

## 2.2 Modularization and automatic configuration of AOCS

AOCS modularization also has to be done on component level. AOCS-specific components that need to be considered, are for example: reaction wheels, momentum wheels, gyros, thrusters, propellant tanks and all kinds of sensors (star, sun, earth, magnetic field, inertial measurement unit). As many of them as possible should be placed

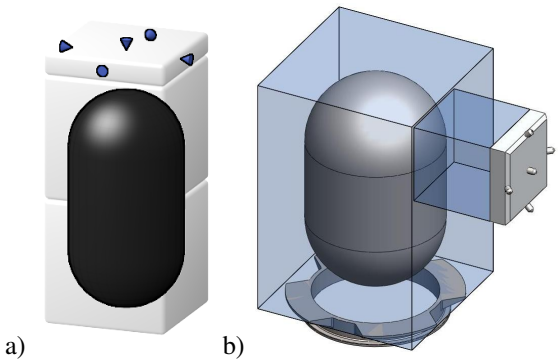


**Figure 3. a) Thruster module with its own propellant tank b) Central tank concept**

in individual modules to make them replaceable in case of a failure or to upgrade or reconfigure a satellite. Some components, like the sun sensor or an inertial measurement unit are very small and can be integrated in multiple modules at once. This has the additional advantage of high redundancy and can also improve sensor data by taking more samples and using sensor fusion methods.

Thruster modules are a special case, as they require a propellant source of some kind. Two concepts were identified in an earlier project phase [3] to handle this problem. In one concept, each thruster module carries its own propellant tank (figure 3 a). The second concept makes use of a central tank structure, which would have the advantage of a favorable mass distribution. In this case an additional mechanism would be needed to feed the propellant from the tank to a thruster module not directly connected to the central tank system of the satellite (figure 3 b).

When using the above-mentioned publish-subscribe messaging pattern, from the point of view of an AOCS control software, there would almost be no difference to a classical, monolithic space system. Besides the newly introduced delay times, the control software would get additional information that is automatically provided by sensors and actuators to the underlying data network. This includes orientation and position data and information about the structure and mass distribution of the system, which can then be used to automatically reconfigure the AOCS to adapt to new module configurations.

## 2.3 Modularization of thermal subsystem

Regarding the use of a centralized or decentralized control system, the thermal subsystem is subject to the same considerations as the AOCS. But beyond this, even more thermal architectures are thinkable and have to be investigated. Architectures with thermally decoupled modules are possible, but most of them require a thermal interface (compare [13]). This interface has to fulfill overall requirements like removing and replacing the modules in orbit. Additionally, in a modular satellite not only the components and the entire spacecraft have to be controlled, but also the modules themselves.

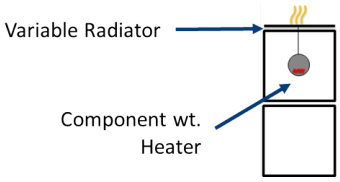


**Figure 4. Isolated architecture: each module is equipped with a heater, a cooler and variable radiators**

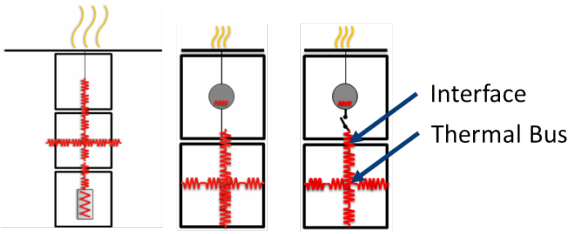


**Figure 5. Three different kinds of architectures with a thermal interface. left: modularized thermal bus; middle: thermal bus; right: discrete thermal bus;**

To master this challenge in all architectures, innovative and classical thermal control elements such as variable radiators, thermal switches and of course heating and cooling elements have to be taken into account. Their implementation can differ in each thermal concept. The control of the modules is comparable to classical satellites with one significant difference: the thermal design has to be done for a wide range of possible orbits according to the application fields of the module catalog. Amongst other things this is a reason why all architectures require for example variable radiators. For the *isolated architecture* (figure 4) there is almost no thermal coupling between the modules. Consequently each module has to be equipped with heaters, coolers and radiators to ensure the standalone thermal control. Using a thermal interface further architectures are possible.

For the *thermal bus* concept, as depicted in figure 5, the module design is similar to the isolated architecture. But beyond this, the interfaces can be used to reduce the thermal gradient of the spacecraft.

The *discrete thermal bus* is similar to the thermal bus, but thermal switches are added at each interface to give them the ability to couple or decouple the elements ther-

mally.

For the *modularized thermal bus* concept there will be no thermal control of individual modules. In this case the TCS modularization has also to be done at component level. The specific modules for heating, cooling and exchanging radiation with the environment have to be placed in the modular satellite at predefined positions comparable to a classical monolithic satellite.

## 3 Simulation architecture for modular satellites

As mentioned above, the development of a modular satellite can be seen as the task of optimizing a module catalog until it fulfills the requirements of certain reference missions. Because of the complexity of this task and as it is not feasible to iteratively build and test multiple modular satellite configurations, a sophisticated simulation is required, which can automate this iterative process.

Detailed models of satellite dynamics, the electrical subsystem and thermal simulation are needed to test the performance of such modular systems. Also, the modularity of the system requires the model to be highly adaptive to configuration changes. As a consequence, a new simulation architecture is being developed, which is specifically adopted to the requirements of modular spacecraft and has to fulfill the following requirements:

- Adapt to changing configurations of a satellite.
- Sufficient accuracy and detail to test satellite requirements.
- Multi-domain simulation to simulate back-coupling effects between different subsystems.
- Ability to include hardware into the simulation (Hardware-In-the-Loop).
- Ability to include communication middleware and control software into the simulation.

The last two requirements mean, that the model has to be capable of real-time simulation.

The resulting architecture is pictured in figure 6. For each domain (mechanical, thermal), separate models are used, which can later be connected to create a true multi-domain simulation. At the current development stage the AOCS and thermal model are not connected.

The module catalog consists of three sub-catalogs, which hold formalized descriptions and parameters of components, modules and reference satellites. The catalog is saved as an XML database to make data exchange between different applications easier. In addition to the catalog a python-interface has been developed to modify and analyze the data.

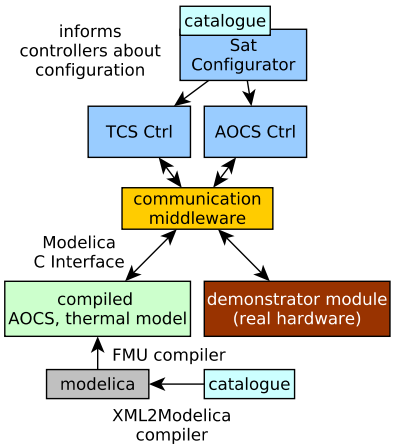


**Figure 6. Simulation architecture**

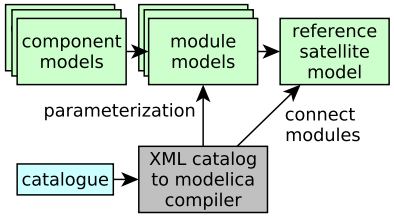


**Figure 7. Building process of hierarchical model from the XML-catalog**

The model of the satellite is implemented using the programming language *modelica* [14]. The component orientated nature of modelica makes it possible to create hierarchical, detailed models of the modular reference satellites simply by reconnecting the modules. (figure 7).

The modules of the AOCS model include several actuators (reaction wheels, gyros, magnetic coils, thrusters) and sensors (star sensor, inertial measurement unit, magnetic field sensor). More sensors are built into every module and thus distributed over the entire system. Each module also has rotation sensors in the interfaces to detect the relative orientation of a module to its neighbors. A control software can then calculate the overall configuration of the satellite using the provided rotation information.

The AOCS model is implemented in modelica as follows: each module consists of a certain number of interfaces, which are represented in the simulation as translational and rotational transformations (black “bars” in figure 8). Orange squares represent bus connectors, which are used to connect the signal bus to neighboring modules and sensors/actuators like an inertial measurement unit (IMU). Gray bars represent transformation connectors, which are used to connect the structure of two mod-

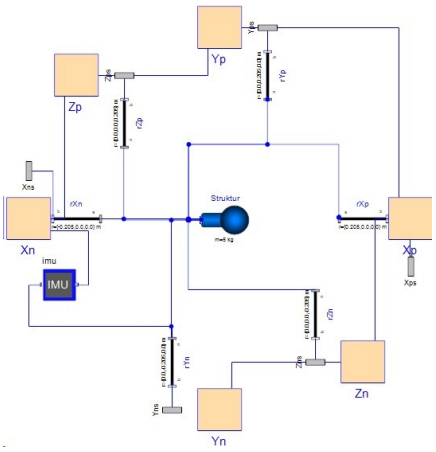


**Figure 8. Internal structure of a single module without specific components and data (orange) and mechanical (gray) connectors.**

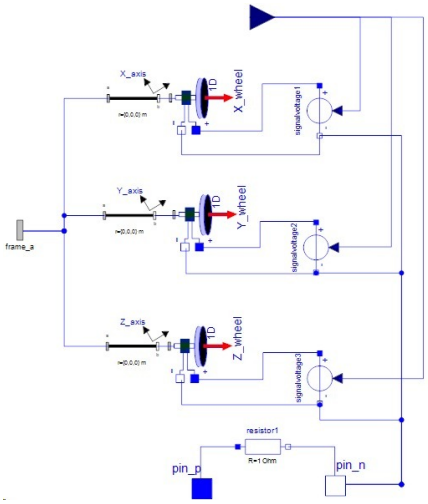


**Figure 9. Model of a 3-axis reaction wheel**

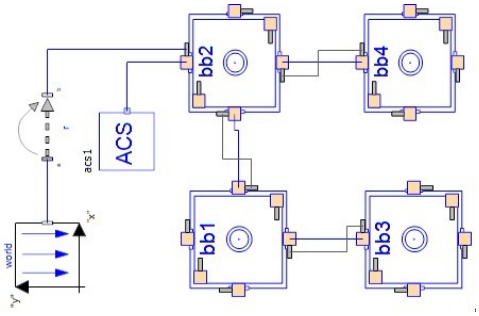


**Figure 10. Top-level model view with four connected modules and publish/subscribe ACS control module**

ules and represent mechanical interfaces. The thermal model is implemented in a similar way using nodes.

Each module also has parameters for its mass, rotation, center of mass and inertia tensor. Within the module individual components are included from a separately developed modelica package of space components, whose properties can also be parametrized (for example a 3-axis reaction wheel with electrical and mechanical interfaces in figure 9).

The data-handling subsystem is implemented as a publish/subscribe network and is simulated by using modelicas expendable connectors. Simulation of delay times, white noise (implemented according to [15] using the Box–Muller transform) and discrete sampling takes place in the individual components. Each module has an individual bus address and all sensor and actuator information of a specific module can be accessed through this address.

Finally, individual modules can be connected together to form an assembly representing a satellite. Usually, depending on the used compiler, simulation state variables are automatically chosen for the final simulation. Because the modules are highly interconnected, many algebraic loops are introduced in the model and the modelica compiler has to be forced to choose the right variables as state by using the *FreeMotion* object from the standard modelica library (the arrow object in figure 10). In the case of flexible connections (simulated using an analogous model consisting of springs), each module has its own set of state variables.

Control programs can be written as an individual modelica block and then connected to signal bus interface of an arbitrary module. It will automatically be detected from the rest of the network. A small, hand-built satellite with an ACS control software is shown as an example in figure 10.

As the satellites can include dozens of modules and its corresponding connections, parametrization by hand would be a relatively tedious task and thus can be performed automatically. For this task a compiler is used that translates the data of the XML catalog into a parametrized modelica model similar to figure 10. In addition, the use of an automatic compiler makes it possible to use the simulation for optimization tasks (figure 11).

## 4 First Results

### 4.1 AOCS model

Some first small simulations have been conducted, using the newly developed AOCS model. The simulation of sensor noise and delay times for specific messaging patterns are already possible. Also, arbitrary module configurations can be assembled. A variety of generic controllers

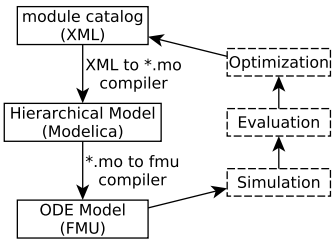


**Figure 11. Optimization loop for simulation model**

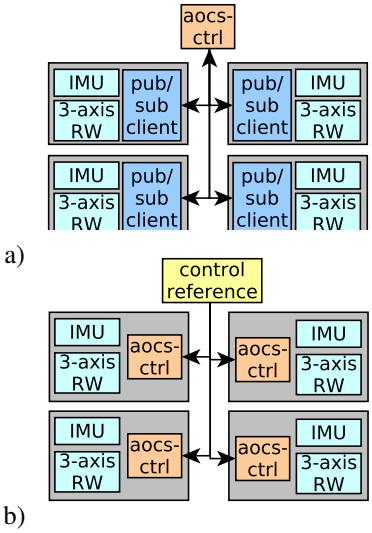


**Figure 12. Simulation models of different messaging layouts. a) publish subscribe architecture, b) one-way communication implementation with distributed controllers**

seem to work out-of-the-box when feeding them with automatically prepared information about the actuators and sensors of the satellite. Using these controllers, a step response to a change in the reference attitude was simulated in two different messaging layouts for the data handling subsystem (figure 12). The publish-subscribe architecture uses the mean values from sensors (IMU) with delay and gives back commands to reaction wheels with additional delay times. The second architecture resembles a one-way communication in the sense that there is no centralized feedback controller. Instead aocs-controllers are distributed over every module and fed with the reference value. The results of the simulation in figure 13 (a) and b) with pub/sub, c) one-way) show that negative effects of modularized spacecraft can be dealt with by combining different communication architectures. Because the control algorithms have not been optimized yet, it is not yet

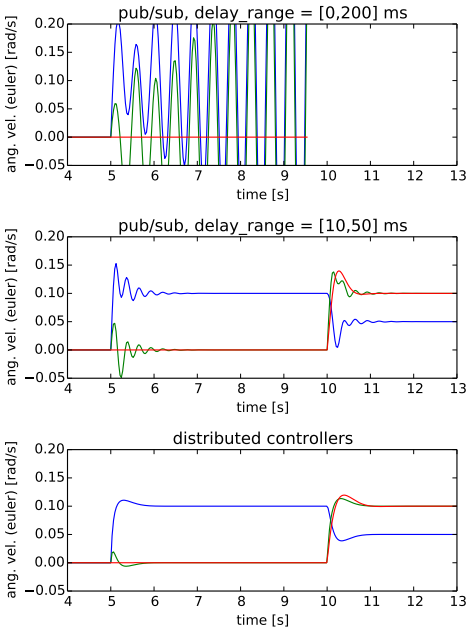


**Figure 13. Step response for a generic state controller (not yet optimized) in different messaging pattern layouts.**

possible to test the performance in relation to monolithic systems at the current development stage.

When using sampled noise, the simulation has been very slow, due to the generation of thousands of events. Because of this reason the noise generator has been rewritten to automatically adapt to the step size of the solver. This feature makes real time simulation possible (figure 14) when a lot of control tasks are being run at the same time.

## 4.2 Thermal model

First simulations have shown that only architectures with a thermal interface are feasible. In the isolated architecture the power demand increases considerably (see [13]). Beyond this, the modularized thermal bus is not the best choice, because simulations show that it results in a higher thermal gradient and higher power consumption for the TCS compared to the thermal bus architecture.

One goal in the optimization of the TCS is to reduce the power consumption of the TCS components. An important part lies on the quality of the heat conductivity of the thermal interface. To figure this out, simulations were done with the thermal conductivity inside and between the blocks as control variables. The temperatures of the electronic boxes inside the modules were logged. The influence of the thermal conductivity on the thermal gradient between the electronic boxes is shown in figure 15.

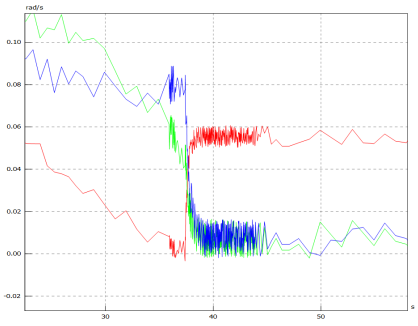

**Figure 14. The frequency of the sensor noise corresponds to the current step width of the simulation. In this example, noise is being calculated much more accurate during a step response.**

A temperature difference of roughly 40K can be achieved with state of the art thermal solutions. That needs to be reduced, to minimize the power requirement of cooling and heating for the entire satellite. Even with a good thermal conduction between the blocks, the gradient there is roughly 20K. This results in requirements for the configuration of the modular system e.g. to place the blocks with higher power dissipation in estimated colder positions. Nevertheless the thermal interface seems to be one of the bottleneck for a modular satellite system.

## 5 Further verification and demonstration plans

The simulation in its current developing stage can only emulate the behavior of the mentioned communication patterns. This makes it very difficult to check the results for their validity. For a better quality estimation of the potential of these modular control systems the integration of the above-mentioned middleware frameworks directly in our simulation is being developed.

The modularity also provides the opportunity to make the simulation itself modular. This can be done by treating each module as an individual simulation thread. The communication in the background is then handled by the original communication middleware. This would enable new possibilities of integrating hardware into the system by replacing individual simulated modules by hardware that is being developed (figure 16). The flexibility and scalability introduced by this approach is also beneficial for interactive simulation.

For all these reasons a demonstrator is planned (figure 17), which integrates with our current simulation architecture and has the following goals:

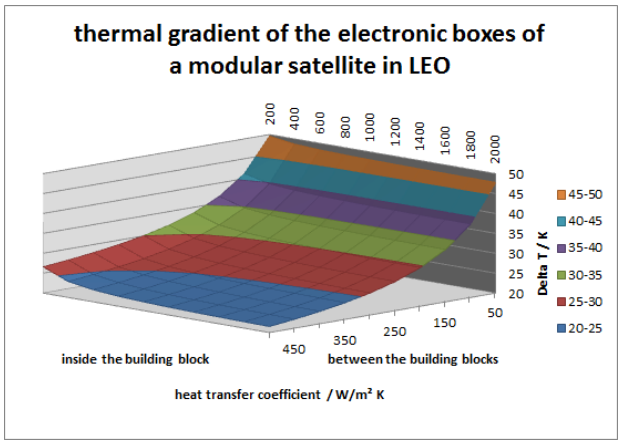


**Figure 15. Thermal gradient of the analogous model of the electronic boxes of a 2x3x2 modular satellite in LEO (sun-pointing)**

**Figure 16. Demonstration modules.**

- Verification of modular flight software for a modular system.
- Test of the newly developed Interface Control Unit (with different optical communication concepts).
- Test of the data concept with respect to typical control tasks of a modular satellite bus.
- Test of redundancy concepts.
- Visualization and testing interaction paradigms with the simulated system.

## 6 Conclusion

An early-stage concept for modular spacecraft, which can adapt their AOCS and thermal subsystems to new reconfigurations has been described. A new simulation architecture has been characterized, which is capable of simulating modular spacecraft in high detail and help with the design of an optimized catalog of modules for building such space systems. Although the simulation is still in an early development stage it is already an essential tool in the design of a modular satellite architecture. The results also provide a starting point to identify key technologies

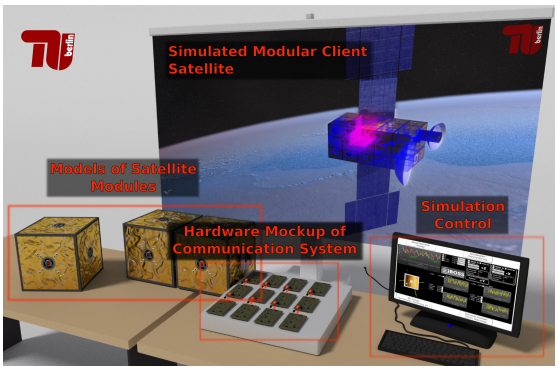


**Figure 17. Hardware setup for the demonstration.**

as, for example, the required network hardware and thermal interfaces for the TCS.

As a next step we aim to develop the simulation to a point where it can make detailed conclusions about the performance of modular spacecraft under different circumstances

## 7 Acknowledgment

This work has been co-funded by the German Aerospace Centre, DLR under national registration no. 50RA1200.

## References

[1] D. Reintsema, J. Thaeter, A. Rathke, W. Naumann, P. Rank, and J. Sommer, "DEOS–the German robotics approach to secure and de-orbit malfunctioned satellites from low earth orbits," *Proceedings of the i-SAIRAS*, pp. 244–251, 2010.

[2] B. Benedict, "Rationale for Need of In-Orbit Servicing Capabilities for GEO Spacecraft," *arc.aiaa.org*, pp. 1–10, 2013.

[3] J. Weise, C. Avsar, M. Buhl, L. Dornburg, T. Meschede, and K. Brieß, "A Novel Design Approach Based on Building Blocks for Servicable Satellites enabling On-Orbit-Servicing," in *62nd International Astronautical Congress*, (Cape Town, South Africa), 2011.

[4] S. Nakasuka, H. Sahara, and Y. Sugawara, "A novel satellite concept "Panel Extension Satellite (PETSAT)" consisting of plug-in, modular, functional panels," 2007.

[5] C. Reynerson, "Spacecraft modular architecture design for on-orbit servicing," *Aerospace Conference Proceedings, 2000 ...*, vol. 4, pp. 227–238, 2000.

[6] R. H. R. Bishop, S. S. J. Paynter, and J. J. W. Sunkel, "Adaptive control of space station with control moment gyros," *Control Systems, IEEE*, no. October, pp. 23–28, 1992.

[7] T. Rupp, T. Boge, R. Kiehling, and F. Sellmaier, "Flight dynamics challenges of the german on-orbit servicing mission DEOS," *... on Space Flight Dynamics*, vol. 49, no. 8153, 2009.

[8] M. Quigley and K. Conley, "ROS: an open-source Robot Operating System," *... on open source ...*, no. Figure 1, 2009.

[9] J. Koning and C. Heemskerk, "Evaluating ITER remote handling middleware concepts," *Fusion Engineering and ...*, Mar. 2013.

[10] W. Shen, B. Salemi, and P. Will, "Hormone-inspired adaptive communication and distributed control for conro self-reconfigurable robots," *Robotics and Automation, IEEE ...*, vol. 18, no. 5, pp. 700–712, 2002.

[11] R. C. Moioli, P. a. Vargas, and P. Husbands, "A multiple hormone approach to the homeostatic control of conflicting behaviours in an autonomous mobile robot," *2009 IEEE Congress on Evolutionary Computation*, pp. 47–54, May 2009.

[12] Z. Butler, K. Kotay, D. Rus, and K. Tomita, "Cellular automata for decentralized control of self-reconfigurable robots," in *Proc. IEEE ICRA Workshop on Modular Robots*, pp. 1–7, 2001.

[13] J. Riesselmann, "Thermal Architectures and Interface Ideas for Modular Servicable Satellites," in *64th International Astronautical Congress*, (Beijing, China), pp. 1–7, 2013.

[14] Modelica Association, "Modelica - A Unified Object-Oriented Language for Physical Systems Modeling Language Specification Version 3.3," tech. rep., 2012.

[15] A. Klöckner, F. van der Linden, and D. Zimmer, "Noise Generation for Continuous System Simulation," *ep.liu.se*, 2014.