Self-optimizing Machine Management

Timo Kautzmann[1], Micaela Wünsche[2], Marcus Geimer[1], Hartmut Schmeck[2], Sanaz Mostaghim[2]
[1]Institute for Mobile Machines (MOBIMA), Karlsruhe Institute of Technology
[2]Institute of Applied Informatics and Formal Description Methods (AIFB), Karlsruhe Institute of Technology

Abstract
Today's machine management systems in off-highway machines are designed to optimize with respect to a target function without integrating the entire machine or considering environmental interactions. For that reason the interdisciplinary project OCOM – "Organic Computing in Off-highway Machines" started in February 2009 to design an architecture for an off-highway machine in order to close that gap. Optimization of fuel consumption is exemplarily chosen even though many other goals are reachable. This paper will introduce the generic architecture; first results will be presented.

Keywords
Machine Management, Organic Computing, Self-learning, Generic Optimization Architecture, Fuel Consumption

1 INTRODUCTION

The management of an off-highway machine can be described as follows. An operator has control over the hierarchically highest functions, for instance velocity. Since an off-highway machine has many degrees of freedom, an automated machine management is necessary to relieve the human operator and to control low-level functions. These are for instance the rotational speed of the shaft and the gear ratio of the transmission. The automated management optimizes these low-level parameters with respect to the operator's request (e.g. for a change in velocity).
Current management systems are designed to optimize for certain defaults, treating the driver's request as a fixed input into the management system and disregarding other environmental influences like current working cycle. Furthermore, due to a lack of an overall machine observation these management systems are not able to regard the machine as a complex and cross-linked entity and therefore are not able to optimize the system holistically with respect to its multi-parametric dependencies. To underline these statements Subsection 2.1 will give some examples.
As a result, current management systems aren't able to fulfil holistic optimization. Holistic optimization will be understood as follows:

- Holistic optimization is supposed to consider environmental influences like attributes set by the operator or the current working cycle. Considering that the settings of the operator, who actually determines the hierarchically highest objectives, are in many cases sub-optimal, an automated optimization with respect to the input of the driver is necessary. Furthermore, off-highway machines, for instance tractors, perform a tremendous number of different working cycles. For example, the optimal parameter settings for transportation are evidently not optimal for heavy duty plowing. Due to the fact that today, an automated machine management system cannot recognize these different situations, it has to find compromises according to defaults of an implemented target function while setting its working point.
- Holistic optimization is supposed to regard the system as a whole. Since there are many cross-linked systems, an optimization coping with this issue must aggregate all possible influences and know how they may interact with each other. Currently, this is not possible, as each subsystem considers its local parameters only because of the lack of an entire machine observation and controlling.

In this paper a new controlling approach with self-adaptive and learning capabilities will be presented that is able to recognize the environment and perform holistic optimization according to the definition

above. Generic control architectures to handle these matters will be introduced in Subsection 2.2 before the most appropriate architecture for performing holistic optimization for this scenariowill be presented in Section 3. Due to the fact that there is a tremendous number of different influences to an off-highway machine, an a priori adaptation of parameters to optimized values is not possible. Hence, the architecture needs to be equipped with certain learning capabilities to find best settings in current situations.
Since the architecture will be integrated into a test vehicle, an introduction into the communication network between architecture and off-highway machine will be given in Section 4. Current and future work is described in Section 5 and 6. Section 7 closes this paper with a conclusion.
Even though the testing vehicle is a Fendt Vario 412 from AGCO GmbH, the intention is to systematically generalize the architecture to other off-highway machines. The currently considered goal is to optimize fuel consumption; however, additional goals are easily conceivable.

2 RELATED WORK

2.1 Today’s Management Systems

A tractor is both a self-organized and cross-linked system as we will see hereafter. Today’s management systems however optimize only parts of this cross-linked system. Reduction of fuel consumption for example is basically achieved by keeping efficiency at a maximum with reduced shaft speed [1, 2, 3]. Here, AGCO’s “Tractor Management System” (TMS) controls engine and transmission while the driver sets desired tractor speed. Engine speed rises according to imprinted load [4]. Furthermore the system switches off four-wheel drive while driving above a certain velocity to eliminate circulating power in the drive train according to Reiter’s suggestion [5]. Tractor management in a Fendt Vario also deactivates the differential lock at a certain steering angle to avoid bad traction conditions between wheel and soil [6].Other systems also integrate pivoting angle and pressure of the hydraulic pump in order to control engine overloading [7].
One first attempt to combine certain sub-systems of a tractor is reported by Jaufmann [8]. He links the management systems of the transmission, engine, chassis, and lifting device in order to reduce fuel consumption. According to Jaufmann a “distinct” reduction of fuel consumption is achievable depending on the working cycle. Frerichs [9] combines tractor and accessory equipment based on tractor-plow aggregate. Control variables are traction force and slip. Actuating variables are working depth and width, location of plow according to tractor, gear ratio of the transmission and shaft speed. A “distinct” optimization of performance and fuel consumption is reachable, too. Kipp [10] implemented the mentioned management based on digital controlling systems and microcomputers and reached an optimization of 15 % both in performance and fuel consumption.
Schreiber [11] examines the potentials of an overall machine optimization that includes the environment and regards the system as a whole. According to him, an average of 5 to 25 % fuel consumption reduction compared with existing machine management systems and for certain working cycles up to 30 % are realistic.
In the following, a closer look at alternative control architectures and the structure of the hierarchical higher management system will be given. This architecture will be adapted to an operable machine.

2.2 Alternative Control Architectures

Today’s management of an off-highway machine, as seen in Section 2.1, can be characterized more generally. The system performs certain working cycles. The associated tasks can only be managed adequately if all individual subsystems work and especially cooperate. In other words, the main task can only be fulfilled by a combination of subsystems that have to coordinate their actions. Today, subsystems are handled individually. To perform the tractor working cycle “disc-harrowing”, for instance, subsystems like combustion engine, transmission drive, power take-off (PTO) and many more are needed in a specific order, where each subsystem has to deliver the right performance at the right time. Such a system is called self-organized. To control such systems, architectures are required that are both robust and flexible at the same time.

In literature, several approaches have been introduced. The Autonomic Computing Initiative by IBM describes the Monitor-Analyze-Plan-Execute (MAPE) cycle [12], where each autonomic element is equipped with an autonomic manager that monitors the state of the managed element, analyzes the received data, then plans an appropriate interaction with the element and executes it.
Another approach that has been developed in the *Sonderforschungsbereich* 614 at the University of Paderborn is described in [13]. An Operator-Controller-Module monitors a technical system and chooses dynamically at runtime between several predefined controller schemes. These schemes have been optimized beforehand by an offline-learning module of the architecture.
An approach that especially emphasizes a self-learning ability of the system, offline as well as online, in order to adjust it even to previously unknown situations is that of the Observer/Controller-architecture (O/C) of the Organic Computing Initiative [14, 15, 16, 17].
Here, the system under consideration (e.g. the tractor) is called a System under Observation and Control (SuOC). This SuOC is capable of performing its intended function on its own, without interference, but not necessarily in an optimal way. The O/C-architecture is intended to optimize the performance and supervise the system as a whole consisting of independent but cooperating subsystems. At the same time it provides an interface for a system user (or a higher level entity) to provide specific optimization objectives (see Figure 1).
To do this, an Observer records and analyzes at all times the status of the SuOC, and reports an aggregated description of the current status to a Controller. This Controller decides whether the system status requires an action, and if so, takes it to influence system performance.
High-level optimization objectives can be given to the Controller. Depending on the observed situation, the controller can also influence settings in the Observer by switching between different models of observation.

In this paper, we will describe how this Observer/Controller-Architecture can be used to reduce fuel consumption in an off-highway machine, specifically a tractor.

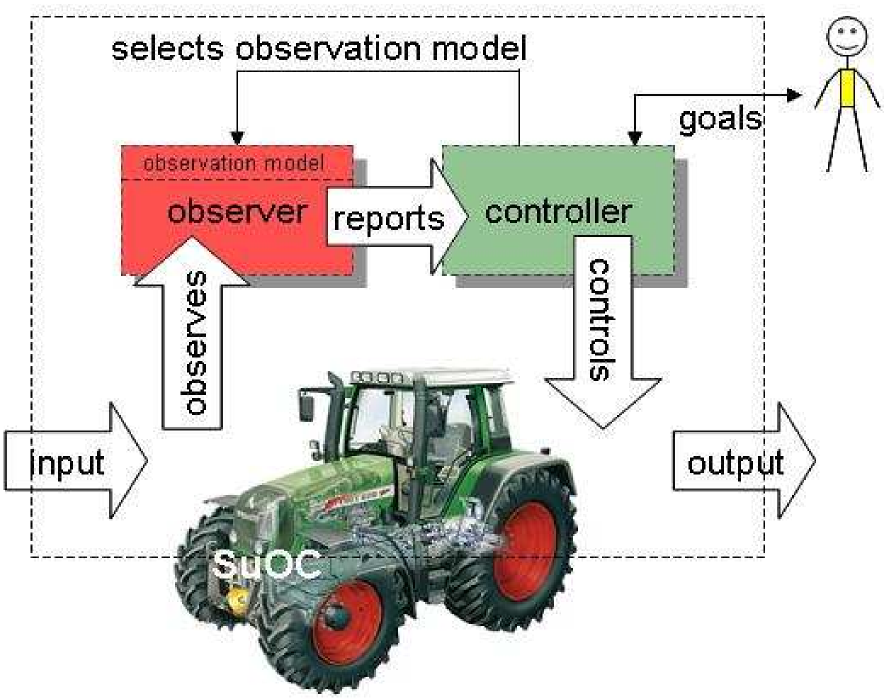

Figure 1: Observer/Controller Architecture

3 ARCHITECTURE

3.1 System under Observation and Control

The system we want to observe and control is a Fendt Vario 412. A tractor basically provides three kinds of power: traction power determined by velocity and traction force, rotational power of the PTO, determined by rotational speed and torque as well as hydraulic power by flow rate and pressure. From a more systematically point of view, a tractor can be divided into four parts: the combustion engine as main power source, the drive train containing transmission and wheels, PTO and the hydraulic pump, which may be linked according to Figure 2. The engine controller basically adjusts shaft speed according to imprinted shaft torque (except for PTO power). Since required traction, rotational and hydraulic power at the tractor's output are input into the model of Figure 2, torque as a result gets adjusted and aggregated at the shaft and thereby determines fuel consumption "be" of the engine. To sum up, a tractor is a deeply cross-linked entity; alterations in one part may lead to a completely new system state.

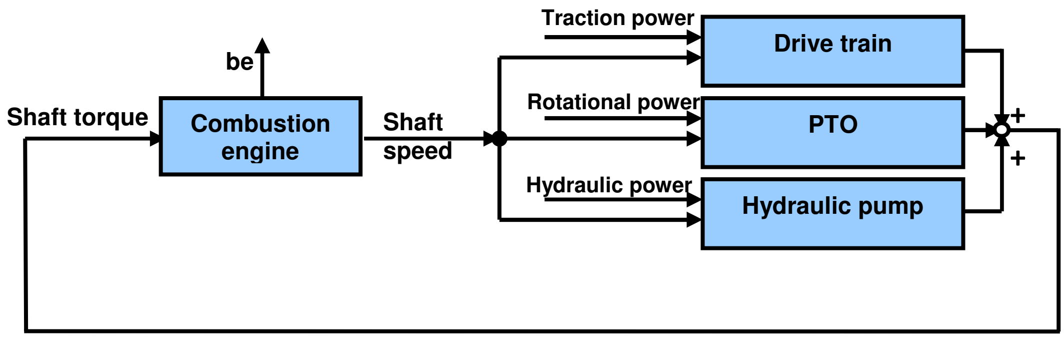

Figure 2: Tractor model

The combustion engine will be regarded as a rotational speed source with adjustable speed. The transmission of the tractor is a ML90, an infinitely variable hydrostatic-mechanical transmission which provides maximum degrees of freedom. Sensor signals to the observer ("raw data") are shown in Figure 3 as well as actuator signals from the controller ("action").

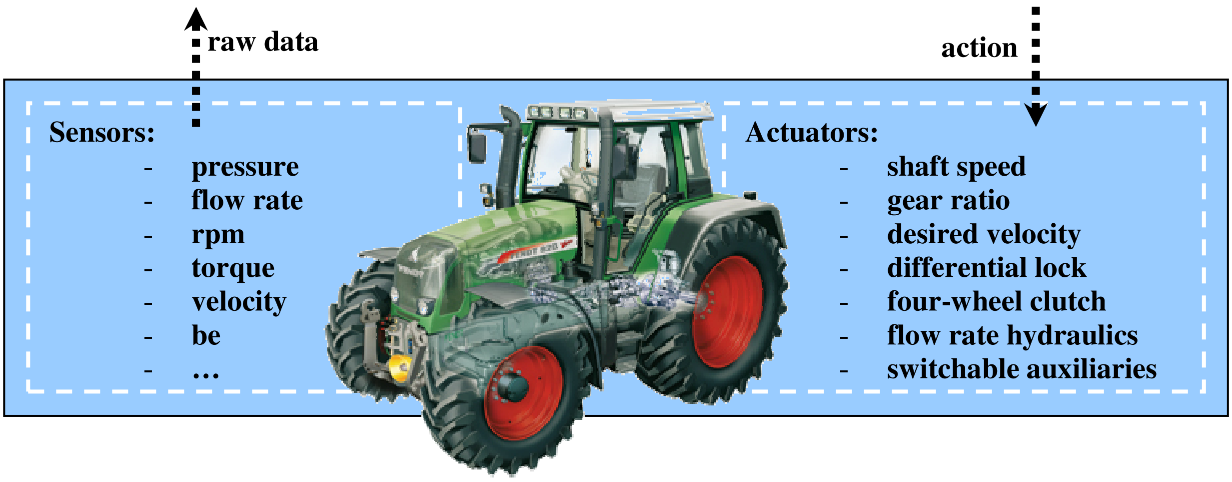

Figure 3: System under Observation and Control

3.2 Observer

The Observer part of the described O/C-architecture continually monitors the System under Observation and Control (SuOC). An internal schematic is shown in Figure 4. Sensor data is sampled by the *monitor* module, both concerning overall system status and individual data from specific components of the machine. Which sensor data is to be read, and at which sampling frequency, is specified in the observation model set by the Controller part of the O/C-architecture. All sampled data is then stored in the *log* module, for possible later use.

In the *pre-processor* module, the monitored data is cleared from noise and outliers by low-pass filtering, before it is evaluated in the *data analyzer* module.

In data analysis, statistical values are derived from specified time windows over the incoming data stream, like arithmetic mean or minimum and maximum value. Also, linear regression and clustering of data points are performed, in order to identify inherent patterns. The aim is to identify the current working cycle of the machine, in order to enable the Controller to adjust all components of the machine appropriately.

To further help the Controller in this task, the *predictor* module of the Observer also receives the system state from the *data analyzer*, and on the basis of this data and the experience it accumulates over time, the following system state is predicted.

All the information gathered within the modules of the Observer is then collected by the *aggregator* module, and passed on to the Controller part of the O/C-architecture.

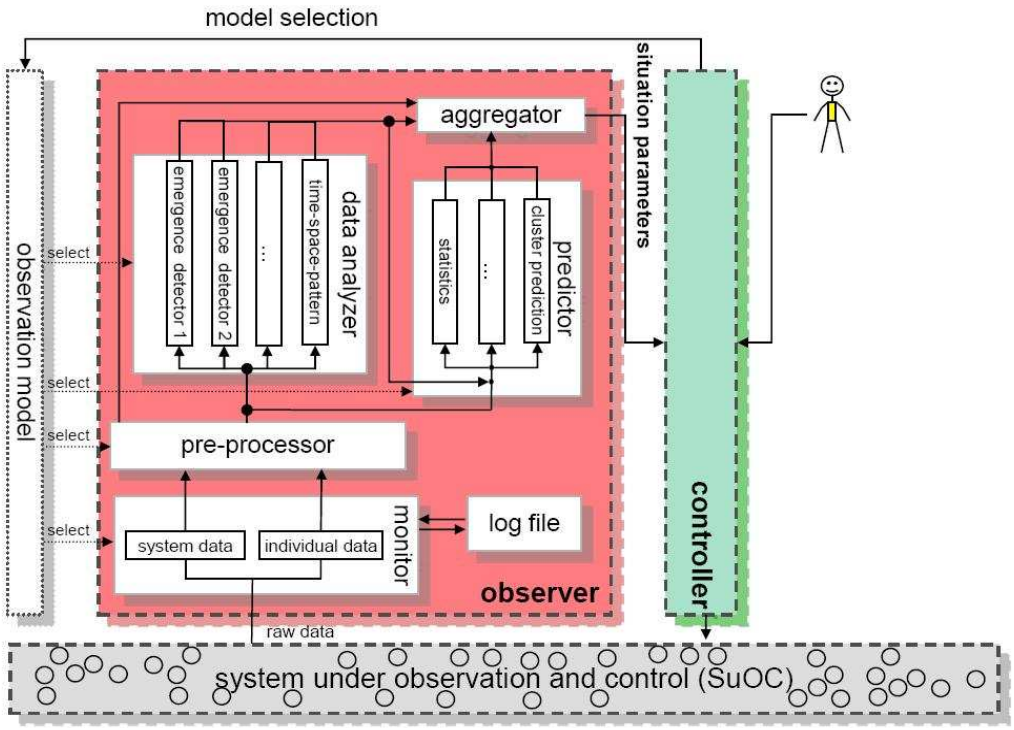

Figure 4: Observer [16]

3.3 Controller

The Controller part of the O/C-architecture receives all relevant information about the system state from the Observer, and on this basis decides in which way to influence the machine. Figure 3 shows the different possibilities for the Controller to act. Internal schematics of the Controller are shown in Figure 5.

For selection of a specific action, the Controller has a *mapping* module that assigns to a system state C_i an appropriate action A_i. Over time, the Controller will adjust this mapping, thus learning to apply the best action in every situation, even if the situation was previously unknown. This learning process takes place at two levels, *online* and *offline*, as is explained in the following.

If the currently reported system state is already part of the mapping, the according action is taken and then stored in the *action history*. After some time steps t, the system state is again recorded in the *situation parameters history*, and considered the outcome of action A_i in situation C_i. Depending on whether the outcome was positive or negative, the corresponding situation-action mapping is evaluated and assigned a fitness value. In this way, the system learns *online* which mappings are best suited. Eventually, mappings with a low fitness will be replaced.
However, if the currently reported system state is not part of the mapping, no immediate action is taken by the Controller to influence the machine. Instead, a *simulation model* of the machine, that is part of the Controller, is initialized with the system state C_i. An *adaption* module generates new rules, tests them *offline* in simulation and evaluates the simulated outcome. The best new rule is then incorporated into the mapping. The same method of *offline learning* of new rules is also used when replacing mappings with a low fitness.
The basis for the evaluation of situation-action mappings always consists in objectives that are provided externally. In this case, it is a reduction of fuel consumption, but the optimization process can be influenced in any direction by imposing an alternative goal.

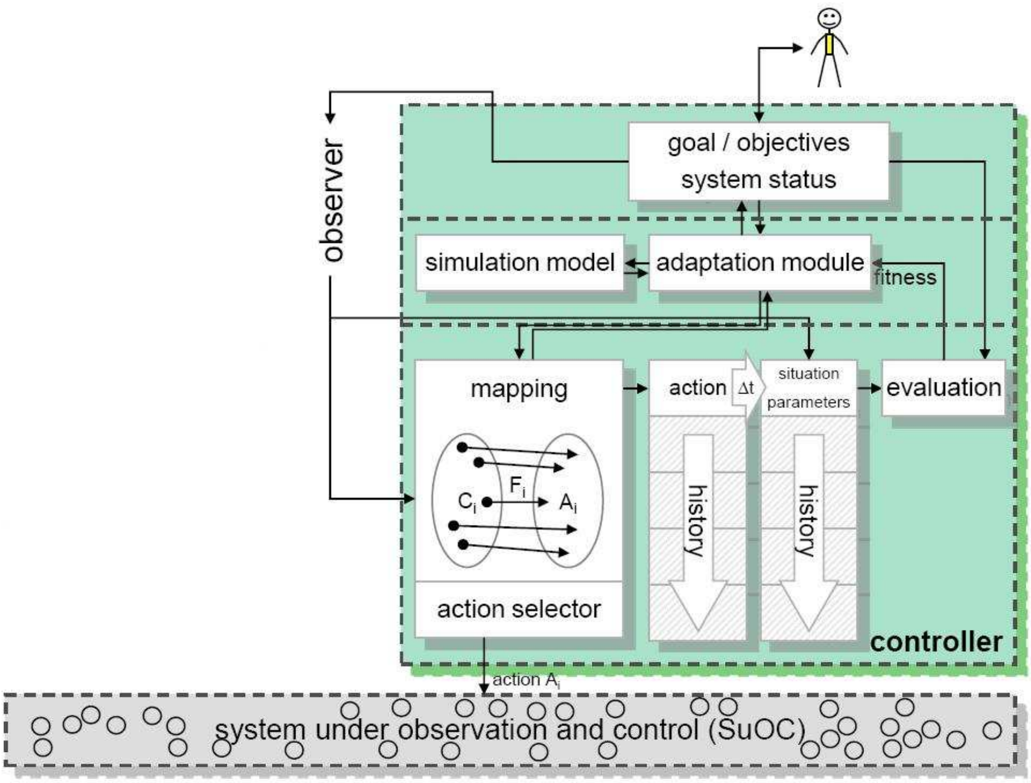

Figure 5: Controller [16]

4 COMMUNICATION

Standard for modern tractors is communication via the CAN-Bus. Fendt Vario 412 communicates basically via 2 different buses, Transmission- and Comfort-Bus. The nodes of the Comfort-Bus connect devices for tractor-driver interaction like operating terminal and joystick. The Transmission-Bus links actuators like hydraulic valves, combustion engine and the electromotor to control gear ratio of the transmission. Both buses are connected with a Tractor Electronic Controlling Unit (TECU). One of its main tasks is to perform safety and plausibility checks. An external interaction like signals from the adapted Observer/Controller architecture should pass through these checks before transferred to tractor actuators. For this reason the generic architecture is supposed to be connected with the Comfort-Bus as shown in Figure 6. In this way TECU retains low-level control. Safety critical proposals of the architecture are supposed to be recommended to the driver via display.

To receive as well as to transmit external signals, an external ECU a so-called CAN-Gateway as an interface between tractor and prototyping hardware is necessary. Since CAN-information is encoded with an identifier that represents the expertise of tractor development, the gateway must be designed in assistance with AGCO. At this point priorities of the signals to be transmitted will have to be determined to guarantee real-time capabilities for the most important information.
The so called "AutoBox" from dSPACE is used as prototyping hardware that includes the Observer/Controller (O/C) architecture. The AutoBox will have the right to transmit as well as to receive signals and is able to request certain signals from the TECU. An additional node for a laptop will be set to visualize data transfer of the OCOM-Bus (see Fig. 6).
Sensor signals that are not receivable via OCOM-Bus must be recorded manually and sent to the AutoBox in a parallel way.

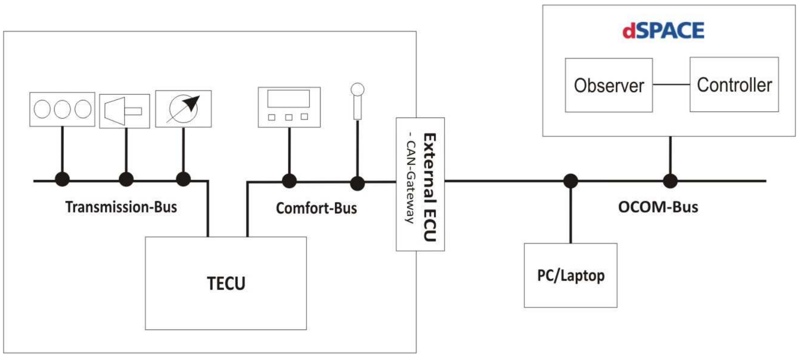

Figure 6: Communication architecture

5 CURRENT WORK: SIMULATION MODEL

After an intense analysis of the tractor, the modelling of the machine in MATLAB/Simulink started recently focusing on the power flow through the individual parts of the machine. Figure 2 in Section 4.1 introduced that model approach. To describe stationary power characteristics, a modelling backwards from load to combustion engine is appropriate because of an easier computation of the power flow.
Having a closer look into the drive train, demanded velocity and traction force is input into an empiric tire-soil model according to Schreiber [11] by using different propulsion-slip and friction-slip curves to compute torque and rotational speed of each insert shaft according to the state of differential lock and four-wheel clutch. After adequate aggregation torque and speed are transferred into the transmission. Efficiency curves especially from the hydrostatic devices in the transmission provided from AGCO are used to build a transmission model to calculate engine speed and torque for each input speed, torque and gear ratio. According to the power demands of the PTO and hydraulic pump, gear ratio is set by a controller.
The stationary engine's fuel consumption map of Fendt Vario 412 provides overall consumption and thereby overall efficiency for each stationary spot can be computed.

6 FUTURE WORK

Future tasks will concentrate on the verification and validation of the simulation model described in Section 5. To learn how the generic architecture works and what sensor signals are necessary to control the tractor properly, an AMESim-Model used as "Software In the Loop" (SIL) - model will be developed. Results of that step will serve to develop the CAN-Gateway in assistance with AGCO. Furthermore sensor signals not to be available via CAN-Gateway must be provided by means of own instrumentation.

Moreover, the Observer and Controller parts of the architecture are adjusted to an off-highway machine. Especially working cycle recognition and system state prediction in the Observer are developed. In the Controller, the system model will be integrated, and appropriate safety mechanisms for the generation of new situation-action mappings established.

7 CONCLUSION

In this paper we outlined an architecture for a self-learning machine management system of an off-highway machine that is based on the generic O/C architecture. The management will be adapted to an existing machine and will be able to perform holistic optimization as described in Section 1. Since additionally equipped with self-learning capabilities, it will be able to evolve permanently. Evidently interesting is the fact that the architecture is supposed to come across all strategies mentioned in Section 2.1 automatically as well as finding new ones resulting of the consideration of the system as a whole in combination with the knowledge of the current working process and driver abilities.
After introducing the architecture, the objectives were outlined by listing problems with state of the art optimizations. Especially the different driving performances and the cross-linkage of the tractor system legitimate the introduction of this new architecture. Then a closer look at the different parts of the generic architecture were given, the tractor as System under Observation and Control which provides overall system information, the Observer which receives and analyzes them and the Controller who generates actions according to the observed situation..Due to the early phase of this project, the presentation of results of using this architecture will be the topic of future publications

ACKNOWLEDGEMENTS

We gratefully acknowledge the financial support by the German Research Foundation (DFG). Furthermore we want to thank AGCO Fendt for their support and helpful hints.

REFERENCES

[1] BRUNOTTE, D.: *Traktormanagement auf dem Systemfahrzeug XERION im Feldversuch*. Landtechnik 2001, pp. 35-40, 2001.

[2] SEEGER, J.: *Untersuchungen von Antriebsstrangstrategien eines TMS*. Landtechnik 1999, pp. 93-98, 1999.

[3] HAAS, W.: *Der Tempomatregler als Beispiel für ein modernes Antriebsstrangmanagement*. Landtechnik 2002, pp. 31-36, 2002.

[4] http://www.fendt.com/de/400-vario-getriebe.asp, last accessed on December 15, 2009

[5] REITER, H.: *Verluste und Wirkungsgrade in Traktorgetrieben. VDI*, Düsseldorf, 1990

[6] N.N.: Workshop manual of a Fendt Vario 400. (unpublished)

[7] FORCHE, J.: *Management hydraulischer Antriebe in mobilen Arbeitsmaschinen*. Landtechnik 2003, pp. 239-244, 2003.

[8] JAUFMANN, A.: *Untersuchung eines Traktormanagements*. Landtechnik 1997, pp. 17-20, 1997.

[9] FRERICHS, L.: *Grundlagen für das elektronische Traktor-Pflug-Management*. Forschungsbericht der Agrartechnik der MEG Nr. 204, Hohenheim, 1991.

[10] KIPP, C.: *Optimierung des Leistungsumsatzes von Traktoren durch Einsatz elektronischer Hilfsmittel*. Dissertation Technische Universität Berlin, Berlin, 1987.

[11] SCHREIBER, M.: Kraftstoffverbrauch beim Einsatz von Ackerschleppern im besonderen Hinblick auf CO2-Emissionen. Shaker, Aachen, 2006.

[12] KEPHART, J. O. and CHESS, D. M.: *The vision of autonomic computing*. IEEE Computer, 1, pp. 41–50, 2003.

[13] OBERSCHELP, O., HESTERMEYER, T., KLEINJOHANN, B. and KLEINJOHANN, L.: *Design of self-optimizing agent-based controllers*. In CfP Workshop 2002 – Agent-Based Simulation 3, 2002.

[14] SCHMECK, H.: *Organic Computing -- A New Vision for Distributed Embedded Systems*. In Proceedings of the Eighth IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC 2005), pages 201--203, Los Alamitos, CA, USA, 2005.

[15] BRANKE, J., MNIF, M., MÜLLER-SCHLOER, C., PROTHMANN, H., RICHTER, U., ROCHNER, F. and SCHMECK, H.: *Organic Computing – Addressing complexity by controlled selforganization*. In Proceedings of the 2nd International Symposium on Leveraging Applications of Formal Methods, Verification and Validation (ISoLA 2006), 2006.

[16] RICHTER, U., MNIF, M., BRANKE, J., MÜLLER-SCHLOER, C. and SCHMECK, H.: *Towards a generic observer/controller architecture for Organic Computing*. In C. Hochberger and R. Liskowsky, editors, INFORMATIK 2006 – Informatik für Menschen!, volume P-93 of GI-Edition – Lecture Notes in Informatics (LNI), pp. 112–119. Köllen Verlag, 2006.

[17] MNIF, M., RICHTER, U., BRANKE, J., SCHMECK, H. and MÜLLER-SCHLOER, C.: *Measurement and control of self-organised behaviour in robot swarms*. In P. Lukowicz, L. Thiele, and G. Tröster, editors, Proceedings of the 20th International Conference on Architecture of Computing Systems (ARCS 2007), volume 4415 of LNCS, pp. 209–223. Springer, 2007.

Contact:
Dipl.-Ing. Timo Kautzmann
Institute for Mobile Machines, Karlsruhe Institute of Technology (KIT)
Fritz-Erler-Straße 1-3 76131 Karlsruhe, Germany
phone: +49 (0721) 608-8652
fax: +49 (0721) 608-8609
Email: Kautzmann@kit.edu