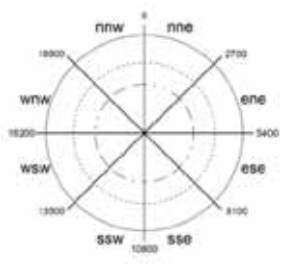
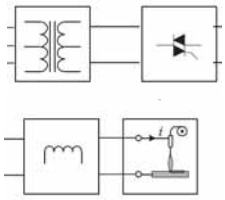
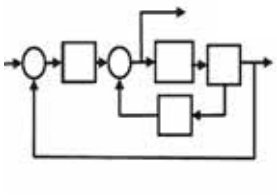
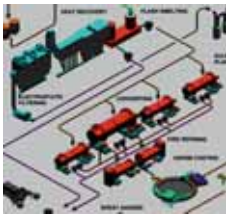
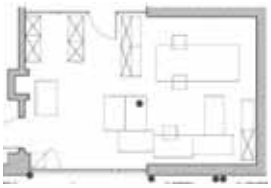
SNE SIMULATION NOTES EUROPE


SNE Special Issue

Modelling and Simulation in Modern Control Engineering

Volume 26 No.4 December 2016 doi: 10.11128/sne.26.4.1035

Journal on Developments and Trends in Modelling and Simulation

Membership Journal for Simulation Societies and Groups in EUROSIM

ARGESIM

SNE Editorial Board

SNE Aims and Scope

Simulation Notes Europe publishes peer reviewed contributions on developments and trends in modelling and simulation in various areas and in application and theory, with main topics being simulation aspects and interdisciplinarity.
Individual submissions of scientific papers are welcome, as well as post-conference publications of contributions from conferences of **EUROSIM** societies.

SNE welcomes also special issues, either dedicated to special areas and / or new developments, or on occasion of vents as conferences and workshops with special emphasis.

Furthermore **SNE** documents the **ARGESIM Benchmarks** on *Modelling Approaches and Simulation Implementations* with publication of definitions, solutions and discussions (*Benchmark Notes*). Special *Educational Notes* present the use of modelling and simulation in and for education and for e-learning. **SNE** is the official membership journal of **EUROSIM**, the Federation of European Simulation Societies. A News Section in **SNE** provides information for **EUROSIM** Simulation Societies and Simulation Groups.

SNE is published in a printed version (Print ISSN 2305-9974) and in an online version (Online ISSN 2306-0271). With **Online SNE** the publisher **ARGESIM** follows the **Open Access** strategy, allowing download of published contributions for free – identified by a DOI (Digital Object Identifier) assigned to the publisher **ARGESIM** (DOI prefix 10.11128).

Print SNE, high-resolution **Online SNE**, full **SNE Archive**, and source codes of the *Benchmark Notes* are available for members of **EUROSIM** societies.

Author's Info. Authors are invited to submit contributions which have not been published and have not being considered for publication elsewhere to the **SNE** Editorial Office. Furthermore, SNE invites organizers of EUROSIM conferences to submit post-conference publication for the authors of their conferences.
SNE distinguishes different types of contributions (*Notes*):

- *Overview Note* – State-of-the-Art report in a specific area, up to 14 pages, only upon invitation
- *Technical Note* – scientific publication on specific topic in modelling and simulation, 6 – 10 pages
- *Education Note* – modelling and simulation in / for education and e-learning; 6 - 8 pages
- *Short Note* – recent development on specific topic, max. 6 p.
- *Software Note* – specific implementation with scientific analysis, 4 – 6 4 pages
- *Benchmark Note* – Solution to an ARGESIM Benchmark; commented solution 4 pages, comparative solutions 4-8 pages

Further info and templates (doc, tex) at **SNE's** website.
www.sne-journal.org

Editorial

Dear Readers - *This fourth SNE issue of the year 2016, SNE 26(4), the special issue 'Modelling and Simulation in Modern Control Engineering' was suggested and compiled by SLOSIM, the Slovenian Simulation Society – underlining the strategy of SNE to publish contributions on recent trends and developments in modelling and simulation. For the front cover we have chosen figures from the eight contributions, sketching model approaches and other characteristic features, which indeed show the broad variety of modelling and simulation in modern control engineering. This issue also rounds up the emphasis of SNE on thematic – oriented issues, with special issues either of topic-oriented subjects – like this issue and SNE 26(3)with emphasis on System Dynamics – or of issues emphasizing on general developments in modelling and simulation – like SNE 26(2) – the 'EUROSIM 2016 Congress' special issue. And last but not least, this issue reflects the status of SNE as membership journal of EUROSIM, the Federation of European Simulation Societies, and the activities of the member societies as SLOSIM, which are invited to edit special issues.*

I would like to thank all authors for their contributions, and especially the guest editor Vito Logar from SLOSIM for compiling this special issue. And last but not least thanks to the Editorial Office for layout, typesetting, preparations for printing, and web programming for electronic publication of this SNE issue.

Felix Breitenecker, SNE Editor-in-Chief, eic@sne-journal.org; felix.breitenecker@tuwien.ac.at

Contents SNE 26(4)
Special Issue 'Modelling and Simulation in Modern Control Engineering'

SNE doi: 10.11128/sne.26.4.1035

SNE Contact & Info

SNE Print ISSN 2305-9974, SNE Online ISSN 2306-0271

→ *www.sne-journal.org*
✉ *office@sne-journal.org, eic@sne-journal.org*

✉ SNE Editorial Office, Andreas Körner
ARGESIM / Math. Modelling & Simulation Group,
Vienna Univ. of Technology /101,
Wiedner Haupstrasse 8-10, 1040 Vienna , Austria

SNE SIMULATION NOTES EUROPE

WEB: → *www.sne-journal.org*, DOI prefix 10.11128/sne

Scope: Technical Notes, Short Notes and Overview Notes on developments and trends in modelling and simulation in various areas and in application and theory; benchmarks and benchmark documentations of ARGESIM Benchmarks on modelling approaches and simulation implementations; modelling and simulation in and for education, simulation-based e-learning; society information and membership information for EUROSIM members (Federation of European Simulation Societies and Groups).

Editor-in-Chief: Felix Breitenecker, Vienna Univ. of Technology, Math. Modelling and Simulation Group
✉ *Felix.Breitenecker@tuwien.ac.at,* ✉ *eic@sne-journal.org*

Layout / Administration: A. Körner, A. Mathe, J. Tanzler, C. Wytrzens, et al.; ✉ *office@sne-journal.org*

Print SNE: Grafisches Zentrum, Vienna Univ. of Technology, Wiedner Hauptstrasse 8-10, 1040, Vienna, Austria

Online SNE: ARGESIM /ASIM, address below

Publisher: ARGESIM ARBEITSGEMEINSCHAFT SIMULATION NEWS
c/o Math. Modelling and Simulation Group,
Vienna Univ. of Technology / 101, Wiedner Hauptstrasse 8-10,
1040 Vienna, Austria;
www.argesim.org, ✉ *info@argesim.org*
on behalf of ASIM *www.asim-gi.org* and EUROSIM
→ *www.eurosim.info*

Editorial SNE Special Issue 'Modelling and Simulation in Modern Control Engineering'

The progress in computer technology and IT has made *Modelling and Simulation* approaches an essential tool in modern industry in several aspects, especially when it comes to control design. The field has been given special attention, as the pursuit of modern systems are both state-of-the-art design and adequate process control, where modelling and simulation approaches are indispensable when effective, optimal and lean operation is required.

The issue starts with the contribution of Š. Korečko et al. - *Jadex/JBdiEmo Emotional Agents in Games with Purpose: a Feasibility Demonstration*. The authors present a 3D game engine jMonkeyEngine, combined with Jadex agent system and JBdiEmo emotional extension and their use in virtual testing grounds for development of software controllers of various devices, embedded to them.

The paper by D. Dovžan et al. - *Evolving Fuzzy Model (eFuMo) method for on-line fuzzy model learning with application to monitoring system* presents an eFuMo method – a modelling approach based on model design and adaptation according to measured data. The authors show that evolving fuzzy model can be used to predict sensor signals in case of their failure. Similarly, eFuMo approach can also be used for model-based control, when real-time measurements are not accessible.

The paper by P. Boškoski et al. - *Model-based prediction of the remaining useful life of the machines* deals with simulation-based life-span prediction of shot blasting machines. The authors show that simulated estimation of the remaining life of the machine is satisfactory and can be used for maintanence planning of the system.

The paper *AMEBA-evolutionary computation method: Comparison and toolbox development,* by M. Corn and M. Atanasijević-Kunc presents the AMEBA method and the corresponding toolbox. The method is based on evolutionary algorithms and can be used for different purposes, such as system identification or control design. Comparative results between AMEBA method and other relevant methods are shown to demonstrate its potential and accuracy when identifying a nonlinear multivariable system.

The next three papers deal with more problem-oriented challenges, arising from practical applications. M. Golob's - *Modelling and Simulation of GMA Welding Process and Welding Power Sources* presents a practical problem arising from power source control in welding units. The author shows that proper electrical model allows controller development, ensuring steady and pulsed direct current welding. A problem oriented study from a different field is in focus of the paper *Inverse Simulation Methods Applied to Investigations of Actuator Nonlinearities in Ship Steering* by D. J. Murray-Smith. The paper shows that inverse simulation methods can be used to predict the ship's rudder saturation and rate limiting effect in terms of the maneuverability of the vessel. It is also shown that a two-stage inverse-simulation method allows direct assessment of the difference between desired and achievable maneuvers. Another study was performed by T. Björkqvist et al. - *Conversion of Iterative Balance Models to Directly Calculating Explicit Models for Real-time Process Optimization and Scheduling*. The authors use a method for direct evaluation of the model output, instead of using an iterative calculation and show its implementation on modelling of the copper production line. The method is used for process optimization and scheduling and is significantly faster than classical modelling methods.

The last paper *Modelling of indoor lighting conditions in buildings for control design purposes*, by V. Logar presents a fuzzy modelling approach to describe indoor lighting conditions in buildings. The model can be, due to its simplicity, used for broader environments, such as control design or model-based control.

The scientific value of this contributions will be used also in the preparation of the curricula and syllabi for the doctoral study in the frame of European ERASMUS+ Project 573751-EPP-1-2016-1-DE-EPPKA2-CBHE-JP entitled 'InMotion - Innovative teaching and learning strategies in open modelling and simulation environment for student-centered engineering education' in which the partner University of Ljubljana, Faculty of electrical engineering also covers the area of modern control systems in computer modelling and simulation engineering.

The editor would like to thank all authors, who have contributed to this special issue and to the SNE Editorial Office for the support in compiling this special issue.

Vito Logar, University of Ljubljana, Faculty of Electrical Engineering; vito.logar@fe.uni-lj.si

Jadex/JBdiEmo Emotional Agents in Games with Purpose: a Feasibility Demonstration

Štefan Korečko*, Branislav Sobota, Peter Zemianek

Department of Computers and Informatics, Faculty of Electrical Engineering and Informatics, Technical University of Košice, Letná 9, Košice, Slovakia; *stefan.korecko@tuke.sk

Simulation Notes Europe SNE 26(4), 2016, 195 - 204
DOI: 10.11128/sne.26.tn10351
Received: November 10, 2016
Accepted: December 5, 2016 (Special Issue Review)

Abstract. The jMonkeyEngine 3D game engine, combined with Jadex agent system and JBdiEmo emotional extension may offer a suitable toolset for effective creation of feature-rich virtual environments, provided that an appropriate interface, allowing to use the full potential of all included components, exists. Then, such environments may profit from the jMonkeyEngine ability to model and simulate the physical world and capability of Jadex and JBdiEmo to express both rational and emotional aspects of characters inhabiting it. One of the meaningful ways of utilization of such environments is to use them as virtual testing grounds for software controllers of various devices, embedded to them. To involve real humans in the testing, they may have a form of a game, where the testing occurs during an interaction between the devices and players. In this paper we present both the interface and the embedding on an emergency simulation game called JFireEmSim2. The primary goal of the player in the game is to rescue a family from a house under fire and the controller embedded into it is of a simple autonomous cleaning robot. The paper describes the architecture of the game, focusing on the interface, implementation of characters as Jadex and JBdiEmo agents and embedding of the controller. It also discusses suitability of the components for the given task.

Introduction

Jadex [5], [6] is a software framework, where applications are composed of active, service providing components. The components can be implemented in several forms, with cognitive BDI agents being historically the first and probably the most sophisticated ones.

BDI stands for belief–desire–intention, a model of human practical reasoning, introduced in [4]. A BDI agent has beliefs expressing what it knows about itself and its environment, desires that represent states it would like to achieve and intentions that provide means to achieve the states.

The simplicity of BDI is the source of both its popularity and criticism. The critics point out that BDI focuses on rational reasoning and ignores other aspects, such as emotions. To deal with this issue in the context of the Jadex framework we designed and implemented JBdiEmo [11], [12] emotional engine, which uses a modified version of the OCC model of human emotions. The OCC model [16] considers emotions to be results of cognitive processes and divides them into three classes: emotions that are reactions to events, reactions to agents and reactions to objects. The version used in JBdiEmo originates from [17] and has a form of an inheritance-based hierarchy of emotions.

A promising utilization of Jadex and JBdiEmo is in computer games, where they can simulate both rational and emotional behavioral aspects of non-playable characters (NPCs), modeled as agents. Such utilization is supported by Jadex via a visualization interface for the jMonkeyEngine (jME) 3D game engine. In [13] we used the interface to develop an emergency simulation game, where the player's goal is to rescue people from a flat under fire. The game proved that JBdiEmo can be used with the interface without any modifications, but also revealed that the interface provides only limited access to jME.

In [13] we intended to use the platform consisting of Jadex, JBdiEmo and jME for ordinary computer games and serious games for education and training. However, from the control system point of view, it is also interesting to examine the possibilities of its utilization for the so-called games with a purpose (GwP) [2], i.e. for games where players are helping to solve serious problems.

GwP try to hide their true purpose behind an interesting gameplay and players can be completely unaware of it. In our case, the GwP should provide a virtual environment for evaluation and testing of software controllers or their executable prototypes. In the game, the controller can be represented by an entity that resembles the device to be controlled by it in the real world. And the purpose of the player will be to evaluate the controller by interacting with the corresponding entity during the gameplay.

There is one prominent issue to deal with in order to use the Jadex/JBdiEmo/jME platform in such way: A new interface between Jadex and jME that will overcome the limitations of the visualization interface, provided by Jadex, should be developed. This is necessary to be able to use the full potential of jME for virtual environment creation. The new interface has been designed and experimentally implemented in a new emergency simulation game, named JFireEmSim2. JFireEmSim2, which is presented in this paper, shares the basic goal with the original one [13], but features more sophisticated gameplay and graphically-rich environment. To demonstrate the feasibility of the GwP idea outlined above, it also includes a formally verified control program supervising an autonomous robot.

The rest of the paper starts with a short overview of the Jadex/JBdiEmo/jME platform and limitations of the original visualization interface (section 1). Section 2 presents the new game, focusing on its overall architecture, new Jadex/jME interface and implementation of NPCs as Jadex/JBdiEmo agents. Section 3 elaborates the GwP idea by presenting a cost-effective version of corresponding development process and embedding of the control program into an already existing game, in this case the JFireEmSim2. The paper concludes with a summary of achieved results and plans for future research and development.

1 The Platform

The software platform, both games are built on, consists of three components, implemented in Java: the Jadex agent system, the emotional engine JBdiEmo and the game engine jMonkeyEngine (jME). While Jadex and jME are standalone components, JBdiEmo can be used only with Jadex.

1.1 Jadex

In Jadex agents are defined by beliefs, goals and plans. Goals stand for desires and plans for intentions. The plans are executed with respect to the current goals of the agent, messages the agent receives or events occurring in the system. On the other hand, an execution of a plan may result in new goals, messages or events.

Agents are specified as classes and actual agents are instances of these classes. Each class is described by an agent definition file (ADF), written in XML. An ADF defines all BDI elements (i.e. beliefs, goals and plans). It contains names, parameters and properties of the elements and links to Java classes that implement beliefs and plans. At runtime, an agent consists of a model, created from his ADF, and a set of instantiated objects, representing his plans and beliefs.

Beliefs are stored in a form of facts. These are accessed by goals and plans to acquire stored data values. A fact can be an arbitrary Java object.

Goals represent agent's specific motivations such as to reach a new state or to perform some activities. Jadex implements a full lifecycle for goals [5]: A goal can be created when its creation condition is met, or during an execution of some active plan. A newly created goal is adopted and enters the main part of its life cycle. On the basis of its context condition, an adopted goal can be in the state ‘option’ or ‘suspended’. A goal in the ‘option’ state eventually becomes active and executes corresponding plans. The active goal can be then suspended if the context condition is broken during the execution of the plans. If the plans of the goal achieve desired results, the goal is finished in the state of success. If they fail to achieve them, it is finished in the state of failure. A goal may also be dropped at any time if its results are no longer desired. Creation, context and other conditions are specified in ADF.

Plans provide means to achieve active goals. For every currently active goal, plans are executed until the goal is reached, suspended, failed or dropped. They are instantiated at runtime when corresponding events (e.g. a goal creation) are triggered. They are also capable of creating new goals.

Configuration of a whole Jadex application is specified in an application XML file, which defines how many agents of which type will populate the application. If 2D or 3D visualization is used, its settings are a part of this file, too.

1.2 JBdiEmo

JBdiEmo engine extends Jadex agents by emotions associated with their rational plans, beliefs and goals. The set of all emotions an emotional agent has, together with their intensity values, form his emotional state. The engine supports the whole modified OCC model and how the emotions are mapped to BDI elements is shown in Figure 1. JBdiEmo is implemented in such a way that agent's actions can influence his emotional state and the emotional state can, in turn, influence agent's further actions.

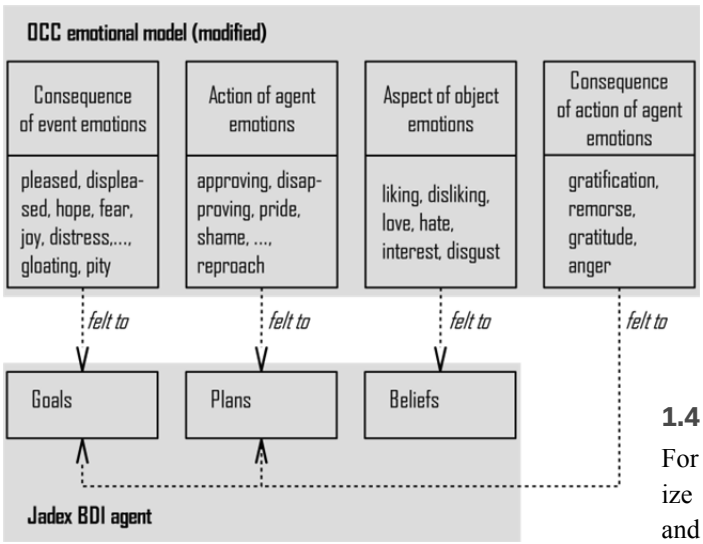


Figure 1: OCC to BDI mapping as implemented in JBdiEmo.

The engine consists of seven components:

1. JBdiEmo core, responsible for representation of agent's emotional state, checking of eliciting conditions of each emotion, emotion intensity value calculation and messaging between emotional agents. Agents access it via a belief.
2. Emotional agent initialization plan, implementing an initial agent model mapping. It initiates processes that repeatedly monitor agent's events and the whole Jadex platform for a presence of other emotional agents.
3. Inter-agent emotional messaging plan, providing message delivery between emotional agents.
4. Language extension allowing to distinguish ordinary beliefs, goals and plans from the ones with associated emotions (i.e. from the emotional ones).
5. GUI for visualization of the actual emotional state of emotional agents. It also shows history of the events that influence emotional plans, beliefs or goals (emotional events).
6. Logger recording emotion intensity values to XML files for future processing.
7. Helpers, which provide mathematical calculations and other auxiliary functionality.

1.3 jMonkeyEngine

jMonkeyEngine (jME) is an open source 3D game engine, built on the OpenGL graphic library. It also provides an integrated development environment, called jMonkeyEngine SDK, which is based on the NetBeans Platform. Thanks to the features like material and terrain editors the comfort of game development in jME is comparable to commercial engines, such as UDK or Unity. In should also be noted that jME uses Bullet to simulate physical phenomena, which, according to [7], is one of the more accurate physics engines available in contemporary games.

1.4 Jadex 3D visualization interface

For Jadex, jME is only one of several options to visualize agent behavior. The other ones are textual output and 2D graphics. The application XML file is responsible for the visualization configuration. Here, 3D models and animations are associated with agents and models, textures and positions of other objects are defined. A direct access to jME is also provided, but only to a subset of its features.

The master-slave relationship between Jadex and jME prevents developers to use important jME features, such as more sophisticated visual effects, collision detection and physics engine. As [13] shows, this puts several constraints on the visual appearance of the game and forces developers to implement mechanisms like collision detection on the Jadex side. In consequence, the time spared by using Jadex to implement NPCs can be lost because of the additional implementation work.

2 JFireEmSim2 Game

The constrains the original visualization interface puts on developers, may render the whole idea of using Jadex and JBdiEmo for NPCs implementation inefficient. Fortunately, they can be overcome by designing a new interface that put Jadex and jME in more equal position.

The *JFireEmSim2* game (Figure 2), which implements such interface, is situated in a village, where the player has two goals. First, he has to rescue a family of four from a house under fire; and second, he has to save a depressed person on a nearby cemetery before he commits suicide. To successfully save the family, all its members have to be taken out of the house. A person may refuse to follow the player because of the fear of getting burned or due to the position in the family.

The actual appearance of the game can be seen in Figure 2, where the situation after saving the first member of the family is captured. There are health bars (red) for all the family members (father John, mother Marie, son Joe and daughter Jane) in the upper left corner. Player's health and extinguisher status are shown in the lower left corner. The player's character is the fireman on the right side, seen from behind. The house under fire is the wooden one on the left. There are two persons in front of the house; a neighbor in white T-shirt, observing the event, and the saved person (Joe, in blue shirt). In the background we can see another fire site and a fireman. Their role is described is Section 3.

Structurally the game can be divided into three components.

1. Game core with the entry point of the game and classes defining the basic gameplay, appearance of the game and user input processing. They have white background and names typed in normal font in the class diagram in Figure 3 and are explained in more detail in Section 2.1.
2. Jadex/jME interface allowing non-restricting communication between both frameworks. It consists of two classes, Communicator and AgentControl (white background and names in bold in Figure 3), described in Section 2.
3. Agents representing NPCs, which implement their behavior. The classes and ADF files belonging here have light gray background in Figure 3, and are treated in Section 2.3.

There is also a fourth part in Figure 3, consisting of 5 classes with dark gray background. These belong to the control program embedded into the game and are described in more detail in Section 3.

2.1 Game core

The core of the game is designed as a typical jME application with its main class, `App`, implementing the jME base class `SimpleApplication`.

The class `App` contains the entry point of the game (method `main`) and methods required by `SimpleApplication`, such as `simpleInitApp` to initialize and `simpleUpdate` to update the game in each game loop cycle. Its properties, among others, provide access to the physics engine (`bulletAppState`), handle various objects in the environment, such as individual fire sites (properties `fire`, `firePositions` and `fireNodes`) and implement 2D user interface elements (e.g. `hudControl` to show health of the player and NPCs). `App` also includes an object called `start`, which initializes Jadex in a separate thread.

The player of the game is represented by the `Player` class, which defines its graphical appearance, including animations, properties (e.g. health, extinguisher charge level) and keyboard and mouse input processing.

NPCs are defined on three levels, first two of them belonging to the core:

1. `Character`, a class which provides basic functionality for all NPCs in the game, such as movement, collision resolution and animations. The class also defines an abstract method `act`, which should define NPC behavior and all its descendants must implement it.
2. Inherited classes, holding aspects specific to corresponding character category. These are SavingPerson for the family members, OtherPerson for a neighbor observing the situation and QuestPerson for the person about to commit a suicide on the cemetery. These classes are connected to Jadex agents (level 3) via corresponding AgentControl objects of the interface and their instances, one for each NPC, are properties of the App class.
3. Jadex agents, specifying their rational and emotional behavior.

All classes implementing entities that can be seen in the game are also connected to corresponding 3D models, textures and animations.

2.2 Jadex/jME interface

The new interface, implemented by classes `Communicator` and `AgentControl`, is designed as universal; it doesn't even require the part connected to Jadex to be implemented in jME. Its purpose is to keep the state of objects representing NPCs on the jME side synchronized with beliefs of corresponding Jadex agents.

Figure 2: JFireEmSim2 screenshot.

The class `Communicator` is implemented using singleton design pattern to ensure that only one instance of it is available in the game. Its property `agents` holds a list of `AgentControl` instances, one for each Jadex agent. The `Communicator` itself just allows to add and remove agents, so the whole synchronization is in the hands of the `AgentControl` objects, which write values to agents' beliefs, with their put methods and read belief values with `getBoolean`, `getInt` and `getFloat` methods.

While on the jME side the interface is accessed via the property `agent` of the classes inherited from `Character`, on the Jadex side it is done through a belief called `shared` and plan `UpdatePlan`. These are defined for each agent. The value of `shared` is an instance of the corresponding `AgentControl` object, obtained via the `Communicator`. The `UpdatePlan` automatically updates beliefs of the agent every time the values stored in the corresponding `AgentControl` object are changed.

2.3 NPCs behavior

The behavior of NPCs in the game is defined almost exclusively by Jadex agents and each NPC has its own ADF. In Figure 3 ADFs are depicted in the form similar to classes, i.e. divided into three blocks. The first one contains stereotype `<<ADF>>` to distinguish them from classes and name of the ADF. The second one lists agent's beliefs and the third one goals. Plans are shown as separate classes, connected to corresponding ADFs. The diagram doesn't show goals and beliefs common to all agents, i.e. the ones belonging to jBdiEmo and the belief `shared`. The goals and beliefs of `Marrie`, `Joe` and `Jane` are similar to those of `John`: The `family_saved` belief is specific to `John`. `Joe` and `Jane` don't have the `child_saved` belief, but have additional belief called `reproached`. `Joe` also has the belief `jane_saved` and `Jane` has `joe_saved`.

To illustrate how the agents define behavior of the NPCs, let us have a look on the family to be saved. When the game begins, an active goal of all family members is wander and `WanderPlan` is carried out. This means that the persons randomly wander around the house. If a person gets too close to a fire, the `run_from_fire` goal becomes active and `RunFromFirePlan` tries to get the person to safe distance. However, when the intensity of the `fear` emotion, felt to the goal `run_from_fire`, becomes too high, the `cry` goal becomes active. This initiates the `CryPlan`, which makes the person stop for few seconds and perform appropriate animation. Another aspect that can abort `RunFromFirePlan` execution is the value of `disapproving` emotion felt to the `wander` goal being higher than `0.7`. Then the `GiveUpPlan` is executed, which means that the person stays in place and only the player's presence can change it by making the `stay_calm` goal active and `StayCalmPlan` executed. When the player commands a person to follow him, the `follow_player` goal and `FollowPlan` are activated, provided the corresponding conditions are met: the mother refuses to follow until the children are saved, and the father is the last one to leave. Otherwise, the `reject_follow` goal and `RejectPlan` are activated.

3 Controller in Game

Now, let us assume that we are developing a safety-critical part of a control program for an autonomous cleaning robot, which works as follows: The robot stores a list of locations to clean in its memory. After it's turned on, it goes to the first location from the list and cleans it. Then it proceeds to the next one. After cleaning all locations, it goes to a parking position and switches to a standby mode.

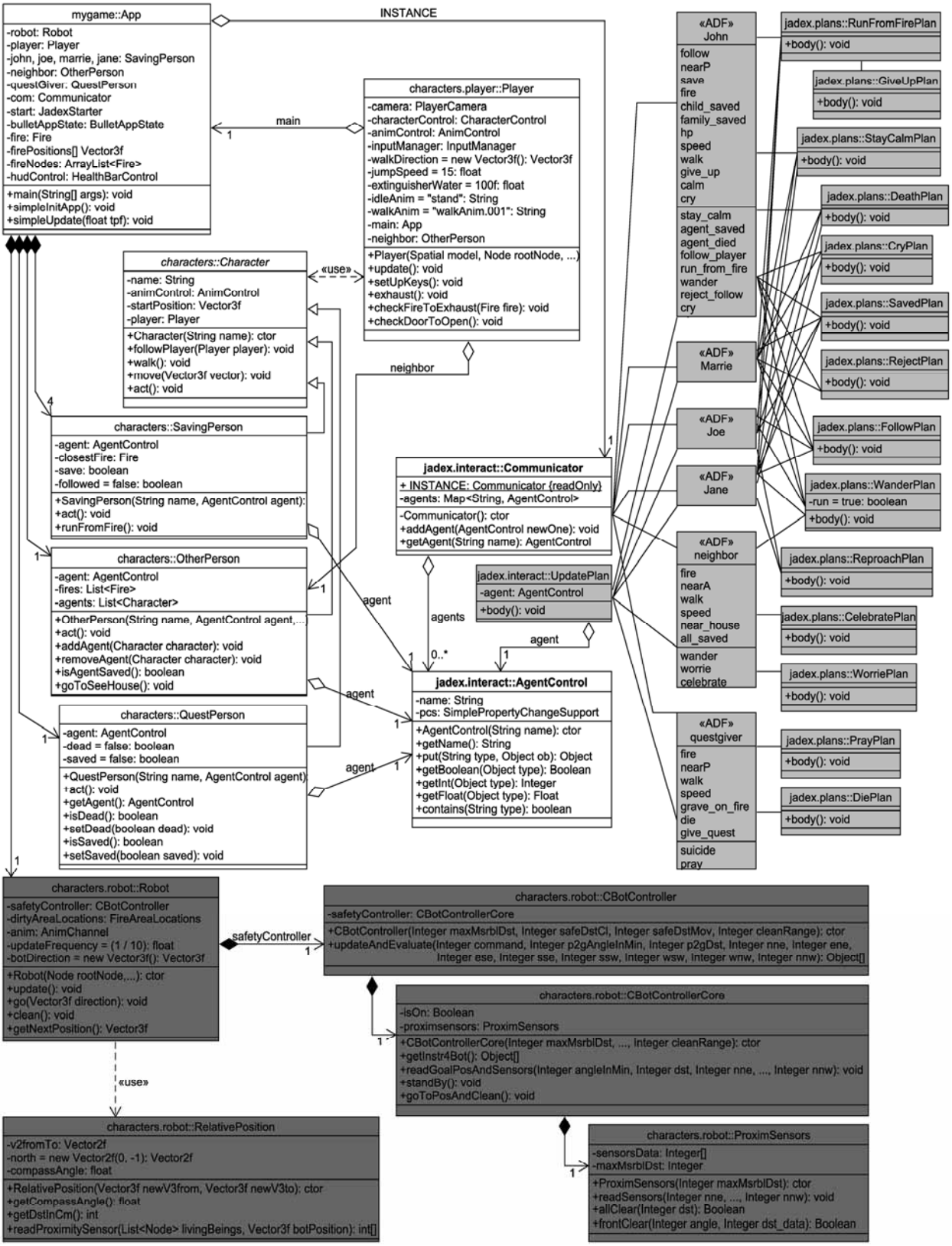


Figure 3: UML class diagram showing essential part of the JFireEmSim2 game structure.

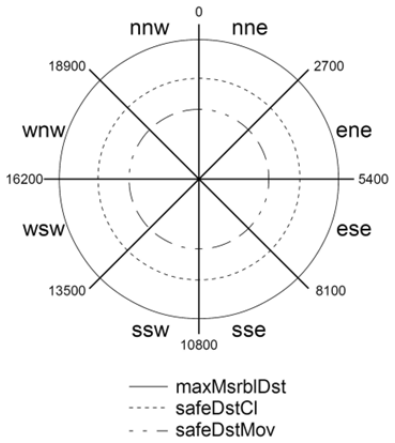


Figure 4: Cleaning bot sensors arrangement.

The robot should be able to perform its job in public areas, so from the safety point of view it is critical to prevent it from hurting people. To detect them, it possesses a circularly arranged sensor array (Figure 4). The sensor array returns 8 values, `nne`, `ene`, `ese`, `sse`, `ssw`, `wsw`, `wnw` and `nnw`. The value `nne` (north north east) is the distance to a nearest person, detected in the region from north (compass angle `0` minutes) to north east (`2700` minutes), `ene` (east north east) the distance in the region from `2700`` to `5400`` and so on. If no person is detected in a region, then the corresponding value is equal to the maximum distance, measurable by the array (`maxMsrblDst`). The robot may hurt someone when cleaning as the cleaning process is harmful for anyone close enough or when a collision occurs during movement. To prevent this, the control program should obey safety critical properties, which can be formulated as follows:

1. The cleaning cannot start or continue if anyone gets as close or closer to the robot as the distance safeDstCl.
2. The robot cannot move if anyone gets as close or closer to its front as the distance safeDstMov.

According to this specification, we design and implement the control program part with a method `updateAndEvaluate` as its interface (Figure 5). The first three parameters of the method specify what the robot should do in the given situation, and the rest (`nne` to `nnw`) are readings from the sensor array. The parameter command defines whether the robot should:

- switch to standby mode immediately (value 0)
- go to a specified position and then clean (1) or
- go to a specified position and stand by (3).

The specified position is given as a compass angle (`p2gAngleInMin`) and distance from the current position (`p2gDst`). The method evaluates the situation and issues instructions for the robot to follow. These include commands to turn the robot on or switch it to standby (the output parameter `botOn`), to start or to stop the cleaning process (`botCleaning`) and the destination where the robot should go (`angleInMin` and `dst`).

Now, assume that the control program part has been developed using formal methods such as B-Method [1], so we are sure that the safety critical properties hold in its implementation. What we are not sure is whether the distances `safeDstCl` and `safeDstMov` are set optimally. They should be large enough to prevent the robot from hurting people, but not too large, as it will cause the robot to interrupt its operation too often. To establish the distances, simulation experiments can be used.

3.1 Jadex/JBdiEmo/jME as simulation platform

We consider Jadex/JBdiEmo/jME to be a suitable platform for such simulation because of the following reasons:

1. Support for quick construction of a virtual environment where the robot will operate, thanks to the editors of jMonkeyEngine SDK and ability to import models already available online or created in different applications. For example, most of the buildings in JFireEmSim2 are freely available models and the house under fire has been created in SketchUp (ww.sketchup.com). NPC models were created in MakeHuman (http://www.makehuman.org/) and their animations in Blender (https://www.blender.org/).
2. Built-in physics simulation in jME, provided by jBullet, a Java port of the Bullet engine. Bullet is used in several simulation platforms, such as Gazebo (http://gazebosim.org/) and V-REP (http://www.coppeliarobotics.com/). The jBullet port has been used for simulation purposes as well, for example in [3] for cells and surrounding fluid.
3. Possibility to populate the virtual environment with characters with complex personality, thanks to the integration of Jadex and JBdiEmo. In addition, the characters can be developed separately and integrated to the environment afterwards.

Both jME and Jadex have already been used for simulation purposes. The Jadex case is described in [6] and jME is the basis of several robotics simulation environments, such as jmeSim [8] and MARS [18].

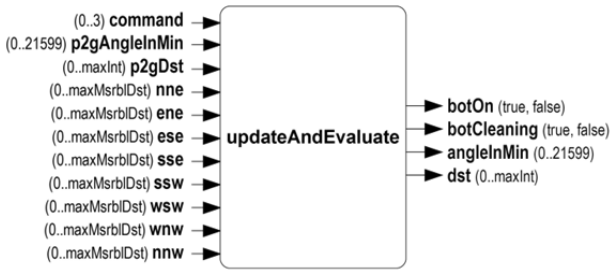


Figure 5: Cleaning bot interface.

3.2 Simulation vs. GwP

In principal, an environment created with Jadex/JBdiEmo/jME can be used in two ways:

1. For simulation experiments.
2. As a game with purpose.

The first case is de facto a game without player, with properties and behavior of entities inhabiting the environment given by their code and models only. So, different simulation experiments under equal conditions can easily be performed.

In the second case, a human player is involved and may interact with an entity representing the control program. The GwP element is that the interaction provides data essential for evaluation or adjustment of the control program, e.g. whether certain values of `safeDstCl` and `safeDstMov` may endanger persons that come close to the cleaning bot. This interaction can be completely natural, for example the cleaning bot can be situated in an area, which the player will visit either way.

3.3 Cleaning bot in JFireEmSim2

While the Jadex/JBdiEmo/jME platform allows to build a virtual environment relatively quickly, the more economical option is to embed the control system to be evaluated to an already existing game. This requires two tasks to be performed:

1. Create an entity that will represent the controlled system in the game.
2. Adjust the code of the game to integrate the controller.

To minimize the effort put into the first task we should consider reusing the assets already available in the game. For example, provided that the basic parameters like dimensions, weight and speed of the cleaning bot are similar to an average human, it can have a form of an additional fireman in JFireEmSim2 and the positions to be cleaned can be represented as fire sites to be extinguished (Figure 6).

Figure 6: The second fireman representing the robot approaching the fire sites, i.e. places to clean.

The controller can be integrated as it is shown in Figure 3 (classes with dark gray background). The controller itself consists of classes `CBotController`, `CBotControllerCore` and `ProximSensors`, which have been generated from formal specification, created and verified using B-Method. The `updateAndEvaluate` method can be found in `CBotController`.

The connection to the game is implemented via the class `Robot`. It serves the similar purpose as the class `Character` for NPCs, i.e. it provides visual representation of the robot (as a fireman). It is also responsible for executing the `updateAndEvaluate` method and performing actions according to the values it returns. The input parameters of `updateAndEvaluate` are computed from actual positions of the robot, place to clean, player and NPCs by the class `RelativePosition`.

4 Related Work

Several aspects presented here can be found in other sources, but the combination of using a game engine for the basic gameplay and simulation of the physical world, emotional agents for characters, behavior and the resulting game for evaluation of the control systems via seemingly ordinary gameplay remains unique to this work. Regarding jBdiEmo, to our knowledge, it is the only existing emotion-implementing extension for Jadex.

The idea of turning a computer game into a testing ground for control systems is in great extent realized with the real-time strategy game Starcraft. According to the survey [15], most of these works are implemented via the Brood War Application Programming Interface (http://bwapi.github.io/), which allows replacing a human player with a computer program and competitions are organized where bots play against other bots or human players.

A toolchain similar to ours has been used in [9], to implement a serious game that teaches players about energy consumption. To create the game the authors of [9] used jME, MakeHuman and Blender, too. The game doesn't include an agent system, but uses co-simulation via the Functional Mock-up Interface (FMI) to integrate thermal and physical models of a building and appliances. FMI should also be considered for a future version of the jME/Jadex interface or an interface between control programs and games.

The GwP idea has been formulated in [2] and GwP usually contain gameplay focused on solving specific problems, such as protein folding in Foldit [10] or finding program loop invariants in Xylem [14].

Other implementations of non-emotional or emotional artificial agents in computer games exist as well and a short overview and comparison to Jadex and JBdiEmo can be found in [13].

5 Conclusion

The new Jadex/jME interface, presented here as a part of JFireEmSim2 game, allows to use both Jadex and jME to their full potential when building virtual environments for games or simulation experiments. The experimental integration of the cleaning bot controller also proved feasibility of the idea of computer games utilization as testing environments with agent-based NPCs and active participation of players. While JFireEmSim2 in the state presented here provides only basic realization of the idea, it can be developed and experimented with in several different ways. One of them is to use Jadex and jBdiEmo to implement more complex gameplay and personalities of NPCs. The gameplay should include active interaction between the player and the robot, for example a task to adjust robot parameters for maximum performance.

Another way is to enhance the possibilities of player movements or perform experiments with selected groups of human players. We also would like to return to the importance of the idea with respect to formal methods, which is touched only lightly here, in a separate work. The JFireEmSim2 game is available by request from the authors.

References

[1] Abrial J. R. *The B-Book: Assigning Programs to Meanings*, 1st ed. Cambridge: CUP; 1996. 816 p., 1996.

[2] Ahn L. von . Games with a Purpose. *Computer*. 2006; 39(6): 92–94.

[3] Applewhite-Grosso T et al. A multi-scale, physics engine-based simulation of cellular migration, *2015 Winter Simulation Conference*; 2015 Dec; Huntington Beach, CA, 1230-1239. doi: 10.1109/WSC.2015.7408248

[4] Bratman, M.E. *Intentions, Plans, and practical reason*.1st ed. Cambridge, MA.: HUP; 1987. 200 p.

[5] Braubach L, Pokahr A, Jander K.. The Jadex Project: Programming Model. In: Ganzha M, Jain C L, editors. *Multiagent Systems and Applications: Volume 1:Practice and Experience*. Berlin, Heidelberg. Springer; 2013. p 33.

[6] Braubach L, Pokahr A. The Jadex Project: Simulation. In: Ganzha M, Jain C L, editors. *Multiagent Systems and Applications: Volume 1:Practice and Experience*. Berlin, Heidelberg. Springer; 2013. p 22.

[7] Erez T, Tassa Y, Todorov E. Simulation tools for model-based robotics: Comparison of bullet, havok, mujoco, ode and physx, *ICRA*, 2015, IEEE. pp. 4397–4404.

[8] Haber A, McGill M, Sammut C. Jmesim: An open source, multi platform robotics simulator. *Australasian Conference on Robotics and Automation*; 2012 Dec; Wellington. 270-276.

[9] Kashif A et al. Virtual Simulation with Real Occupants Using Serious Games. *14th International Conference of the International Building Performance Simulation Association*; 2015 Dec; Hyderabad. 2712-2719.

[10] Khatib F et al. Crystal structure of a monomeric retroviral protease solved by protein folding game players. *Nat Struct Mol Biol*. 2011;18(10): 1175–1177. doi:10.1038/nsmb.2119.

[11] Korečko Š, Herich T. On Some Concepts of Emotional Engine for BDI Agent System, *14th IEEE International Symposium on Computational Intelligence and Informatics*; 2013 Nov; Budapest. 527-532. doi: 10.1109/CINTI.2013.6705254.

[12] Korečko Š, Herich T., Sobota B. JBdiEmo – OCC Model Based Emotional Engine for Jadex BDI Agent System, *12th International Symposium on Applied Machine Intelligence and Informatics*; 2014 Jan; Herľany. 299-304. doi: 10.1109/SAMI.2014.6822426.

[13] Korečko Š, Sobota B, Čurilla P. Emotional Agents as Non-playable Characters in Games: Experience with Jadex and JBdiEmo, *15th IEEE International Symposium on Computational Intelligence and Informatics*; 2014 Nov; Budapest. 471-476. doi: 10.1109/CINTI.2014.7028721.

[14] Logas H. et al. Software Verification Games: Designing Xylem, The Code of Plants. *9th Int. Conf. Foundations of Digital Games*; 2014 April; Ft. Lauderdale.

[15] Ontanon S. et al. A Survey of Real-Time Strategy Game AI Research and Competition in StarCraft. *IEEE Trans. on Computational Intell. and AI in Games*. 2013; 5(4):1-19.

[16] Orthony A, Clore G, Collins A. *The cognitive structure of emotions*. Cambridge: CUP; 1988. 207 p.

[17] Steunebrink B.R, Dastani M, Meyer J.J.Ch. The OCC model revisited. In: Reichardt D, editor. *Proc. of the 4th Workshop on Emotion and Computing - Current Research and Future Impact*, 2009.

[18] Tosik T, Maehle E. MARS: A simulation environment for marine robotics, *OCEANS 14 MTS/IEEE*; 2014 Sept; St. John's. doi: 10.1109/OCEANS.2014.7003008

Evolving Fuzzy Model (eFuMo) Method for on-line Fuzzy Model Learning with Application to Monitoring System

Dejan Dovžan*, Vito Logar, Igor Škrjanc

Faculty of Electrical Engineering, University of Ljubljana, Tržaška 25, 1000 Ljubljana, Slovenia;
*dejan.dovzan@fe.uni-lj.si

SNE Simulation Notes Europe SNE 26(4), 2016, 205-220
DOI: 10.11128/sne.26.tn10352
Received: November 15, 2016
Accepted: December 5, 2016 (Special Issue Review)

Abstract. With evermore complex system the monitoring and fault detection is becoming a crucial part of control systems. They allow fast and effective fault diagnosis and can decrease the cost of system maintenance. Modelling of processes plays a crucial part when designing a monitoring system. In this paper an on-line approach for modelling of fuzzy model is presented (Evolving Fuzzy Model - eFuMo). As demonstrated in the paper, the method can be used in the design of model based fault detection system.

Introduction

Increasing demands of productivity and reliability call for extending the ability of a common SCADA systems with the monitoring and fault detection systems. There are several approaches for designing the fault detection system. In our paper the monitoring system is based on a process model. The model is based on a evolving fuzzy model method (eFuMo). The presented method is able to build Takagi-Sugeno fuzzy model (TS) from scratch, starting with one cluster and a local model. The TS fuzzy models are a powerful practical engineering tool for modelling and control of complex systems. They expand and generalize the well-known concept of gain scheduling. They utilize the idea of linearization in a fuzzily defined region of the state space. Due to the fuzzy regions (clusters), the nonlinear system is decomposed into a multi-model structure consisting of linear local models [1].

This enables the T-S fuzzy model to approximate virtually any nonlinear system within a required accuracy, provided that enough regions are given [2].

The eFuMo method is an on-line learning method that is also able to adapt models during the functioning of the system. Depending on the learning abilities, the on-line fuzzy-identification methods can be divided into: *Adaptive methods* (e.g., ANFIS [3], GANFIS [4], rFCM [5], rGK [6]), where the initial structure of the fuzzy model must be given. The number of space partitions/clusters does not change over time, only the parameters of the membership functions and local models are adapted; *Incremental methods* (e.g., RAN [7], SONFIN [8], SCFNN [9], NeuroFAST [10], DENFIS [11], eTS [12], FLEXFIS [13], PANFIS [14]), where only adding mechanisms are implemented; *Evolving methods* (e.g., SAFIS [15], SOFNN [16], GAP-RBF [17], EFuNN [18, 19], D-FNN [20], GD-FNN [21], ENFM [22], eTS+ [23], ENFM [22], FLEXFIS++ [24], AHLTNM [25], SOFMLS [26]) which, besides an adding mechanism, implement removing and some of them also merging and splitting mechanisms. More on evolving methods can be found in [27] and [28], where concepts and open issues regarding these methods are presented.

The paper is organized in the following order. First, the eFuMo learning method is described, next the monitoring problem is given followed by results and conclusions.

1 eFuMo Structure

The eFuMo method has two types of mechanisms for identifying the fuzzy model: the adaptation algorithm and the evolving mechanisms. The first is responsible for parameter adaptation, such as cluster centers and lo-

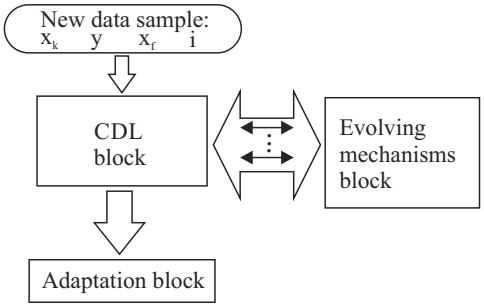


Figure 1: The eFuMo top scheme.

cal models' parameters; the second is responsible for structure update: adding, removing, merging and splitting of clusters. A central decision logic (CDL) decides which type of mechanism will be used at current sample. The block scheme is presented on Figure 1. In the following subsection, the adaptation and evolving mechanisms will be presented and at the end the CDL will be described.

1.1 Adaptation mechanisms

In order to build the TS fuzzy model clusters and local linear models must be identified. Adaptation mechanisms are responsible for identifying clusters' and local models' parameters and for their adaptation. To partition input-output data space recursive clustering algorithm is used and for identifying the local models' parameters the fuzzy recursive least squares is used.

Space partitioning. For data space partitioning, the cluster centers and fuzzy covariance matrix must be calculated. The centers are adapted with the following equation:

$$\mathbf{v}_i(k+1) = \mathbf{v}_i(k) + \Delta\mathbf{v}_i(k) \tag{1}$$

$$\Delta\mathbf{v}_i(k) = \frac{\mu_i(k)^\eta \left(\mathbf{x}_f(k) - \mathbf{v}_i(k)\right)}{s_i(k)} \tag{2}$$

where η is fuzziness factor, $\mathbf{v}_i$ is the center position vector $\mathbf{x}_f$ is clustering vector, μ_i is membership degree of the current clustering vector to the i-th cluster also called the firing degree of the i-th cluster and $s_i(k+1)$ is the sum of past membership degrees / firing levels of the i-th cluster:

$$s_i(k) = \lambda_c s_i(k-1) + \mu_i(k)^\eta. \tag{3}$$

where λ_c was introduced as a forgetting factor to enable the adaptation of centers. The membership degrees μ_i can be calculated as in equation 4 (c is the number of existing clusters), either based on rFCM [5] (equation 5), rGK [6] (equation 6) or Mahalanobis distance (equation 7).

$$\mu_i(k) = \begin{cases} \dfrac{1}{\sum_{j=1}^{c}\left(\frac{d_i(k)}{d_j(k)}\right)^{\frac{2}{\eta-1}}} & \text{if } \mathbf{x}_f(k) \neq \mathbf{v}_i; i = 1,...,c \\ 1 & \text{if } \mathbf{x}_f(k) = \mathbf{v}_i \\ 0 & \text{if } \mathbf{x}_f(k) = \mathbf{v}_j; i \neq j \end{cases} \tag{4}$$

$$d_i(k) = \left(\left(\mathbf{x}_f(k) - \mathbf{v}_i(k)\right)^T \left(\mathbf{x}_f(k) - \mathbf{v}_i(k)\right)\right)^{0.5} \tag{5}$$

$$d_i(k) = \left(\left(\mathbf{x}_f(k) - \mathbf{v}_i(k)\right)^T \det(\mathbf{F}_i)^{\frac{1}{p}} \mathbf{F}_i^{-1} \left(\mathbf{x}_f(k) - \mathbf{v}_i(k)\right)\right)^{0.5} \tag{6}$$

$$d_i(k) = \left(\left(\mathbf{x}_f(k) - \mathbf{v}_i(k)\right)^T \mathbf{F}_i^{-1} \left(\mathbf{x}_f(k) - \mathbf{v}_i(k)\right)\right)^{0.5} \tag{7}$$

To get the area of cluster influence the fuzzy covariance matrix $\mathbf{F}_i$ is calculated. The recursive equation for $\mathbf{F}_i$ is the following:

$$\mathbf{F}_i(k+1) = \gamma_c \frac{s_i(k-1)}{s_i(k)} \mathbf{F}_i(k) + \frac{\mu_i(k)^\eta}{s_i(k)} \mathbf{D}_{F_i}(k)$$
$$\mathbf{D}_{F_i}(k) = \left(\mathbf{x}(k) - \mathbf{v}_i(k)\right)\left(\mathbf{x}(k) - \mathbf{v}_i(k)\right)^T. \tag{8}$$

where γ_c is the forgetting factor. To be able to calculate the Gustafson-Kessel clustering distance (equation 6) the inverse and determinant of fuzzy covariance matrix must be calculated. The recursive equation for the inverse matrix is obtained by using the Woodbury matrix identity lemma. The equation is following:

$$\left[\mathbf{F}_i(k+1)\right]^{-1} = \frac{1}{\gamma_c} \frac{s_i(k)}{s_i(k-1)} \left[\left[\mathbf{F}_i(k)\right]^{-1} - \frac{\mathbf{B}}{C}\right] \tag{9}$$

$$\mathbf{B} = \left[\mathbf{F}_i(k)\right]^{-1} \mathbf{D}_{F_i} \left[\mathbf{F}_i(k)\right]^{-1} \tag{10}$$

$$C = \gamma_c \frac{s_i(k-1)}{\mu_i(k)^\eta} + d_{F_i}^T \left[\mathbf{F}_i(k)\right]^{-1} d_{F_i} \tag{11}$$

$$d_{F_i} = \mathbf{x}_f - \mathbf{v}_i(k). \tag{12}$$

The determinant is obtained using determinant lemma (equation 13):

$$\det(\mathbf{A}+\mathbf{u}\mathbf{v}^{\mathrm{T}}) = (1+\mathbf{v}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{u})\det(\mathbf{A}). \tag{13}$$

The recursive equation for determinant calculation is:

$$\det(\mathbf{F}_i(k+1)) = \left(\gamma_c \frac{s_i(k-1)}{s_i(k)}\right)^p \det(\mathbf{F}_i(k))(1+A) \tag{14}$$

$$A = \frac{1}{\gamma_c}\frac{\mu_i(k)^\eta}{s_i(k-1)}\left(d_{F_i}^T[\mathbf{F}_i(k)]^{-1}d_{F_i}\right), \tag{15}$$

where p is the number of rows/colmuns of fuzzy covariance matrix. The detailed derivations of equations are given in [6]. The eFuMo method implements the method for stopping the cluster parameters adaptation if the clusters firing level is below a certain user defined threshold $\beta_{cut_{trh}}$. This prevents clusters, that are far from current clustering vector, to converge to that area. The clusters' firing levels 4 that are below the threshold are set to zero. The rest of the firing levels are then normalized.

Local models' parameters identification.

Each cluster has a linear local model, that is valid in that area. The output of the local model is calculated as:

$$y_{m_i}(k) = \theta_i^T[1 \;\; \mathbf{x}_k(k)^T]^T, \tag{16}$$

where $\mathbf{x}_k(k)$ is the regression vector and θ_i^T are the local model parameters. The regression vector is usually the input part of the clustering vector:

$$\mathbf{x}_f(k) = [\mathbf{x}_k(k)^T y(k)]^T, \tag{17}$$

where y is the process output. However unlike many existing on-line fuzzy identification methods the eFuMo method allows the clustering vector to be different than the regression vector.

The eFuMo has different fuzzy least squares based identification methods included ([12], [22], [5] and [29]). Usually best results are obtained using local fuzzy weighted least squares presented in [12]:

$$\begin{aligned}\mathbf{x}_e(k) &= [1 \;\; \mathbf{x}_k(k)^T]^T \\ \mathbf{P}_i(k+1) &= \frac{1}{\lambda_r}\left(\mathbf{P}_i(k) - \frac{\beta_i \mathbf{P}_i(k)\mathbf{x}_e(k)\mathbf{x}_e^T(k)\mathbf{P}_i(k)}{\lambda_r + \beta_i \mathbf{x}_e^T(k)\mathbf{P}_i(k)\mathbf{x}_e(k)}\right) \\ \theta_i(k+1) &= \theta_i(k) + \mathbf{P}_i(k)\beta_i\mathbf{x}_e(k)\left(y(k) - \mathbf{x}_e^T(k)\theta_i(k)\right)\end{aligned} \tag{18}$$

where i is the cluster index, θ is the vector of local model's parameters and β is the firing level of cluster and the y is the process output. The firing levels are calculated on the input space. Usually the methods use Gaussian functions equation 19 or equation 20:

$$\begin{aligned}\mu_i(k) &= e^{-\frac{(x_{f_k} - v_{i_k})^2}{2\eta_m \mathbf{F}_{i_{k,k}}}} \quad k = 1,2,...p-1 \;\; i = 1,2,...,c \\ \beta_i &= \prod_{k=1}^{p-1}\mu_i(k)\end{aligned} \tag{19}$$

$$\begin{aligned}\beta_i &= e^{\frac{-D_{ik}^2}{2\eta_m}} \quad i = 1,2,...,c, \\ D_{ik}^2 &= \left(\mathbf{x}_{f_{in}}(k) - \mathbf{v}_{i_{in}}\right)^T \mathbf{F}_{i_{in}}^{-1}\left(\mathbf{x}_{f_{in}}(k) - \mathbf{v}_{i_{in}}\right).\end{aligned} \tag{20}$$

where η_m is the overlapping factor usually set to 1, $\mathbf{F}_{i_{k,k}}$ is diagonal element k of fuzzy covariance matrix, x_{f_k} and v_{i_k} are the k-th element of clustering vector and k-th element of i-th cluster center vector, respectively. The $\mathbf{F}_{i_{in}}$ is the input fuzzy covariance matrix, $\mathbf{x}_{f_{in}}$ is the clustering vector containing only the input variables and $\mathbf{v}_{i_{in}}$ is the cluster center in an input space. The obtained firing levels are then normalized:

$$\beta_i = \frac{\beta_i}{\sum_{k=1}^c \beta_k} \quad i = 1,2,...,c. \tag{21}$$

One can also use the same equation for firing level calculation as with clustering algorithm. However, with Gussian functions the transitions between local models (clusters) are more smooth.

When building the simulation model, the model parameters can be identified more accurately using the instrumental version of least squares [30]. The instrumental variable adaptation algorithm for equation 22 can be written as:

$$\begin{aligned}\mathbf{x}_e(k) &= [1 \;\; \mathbf{x}_k(k)^T]^T \\ \mathbf{x}_{e_m}(k) &= [1 \;\; \mathbf{x}_{k_m}(k)^T]^T \\ \mathbf{P}_i(k+1) &= \frac{1}{\lambda_r}\left(\mathbf{P}_i(k) - \frac{\beta_i \mathbf{P}_i(k)\mathbf{x}_{e_m}(k)\mathbf{x}_e^T(k)\mathbf{P}_i(k)}{\lambda_r + \beta_i \mathbf{x}_e^T(k)\mathbf{P}_i(k)\mathbf{x}_{e_m}(k)}\right) \\ \theta_i(k+1) &= \theta_i(k) + \mathbf{P}_i(k)\beta_{i_m}\mathbf{x}_{e_m}(k)\left(y(k) - \mathbf{x}_e^T(k)\theta_i(k)\right)\end{aligned} \tag{22}$$

where $\mathbf{x}_{e_m}(k)$ is the regression vector where the delayed process outputs were replaced with model outputs and β_{i_m} is the membership degree of vector $\mathbf{x}_{f_m}(k)$, which is the clustering vector, where delayed process outputs

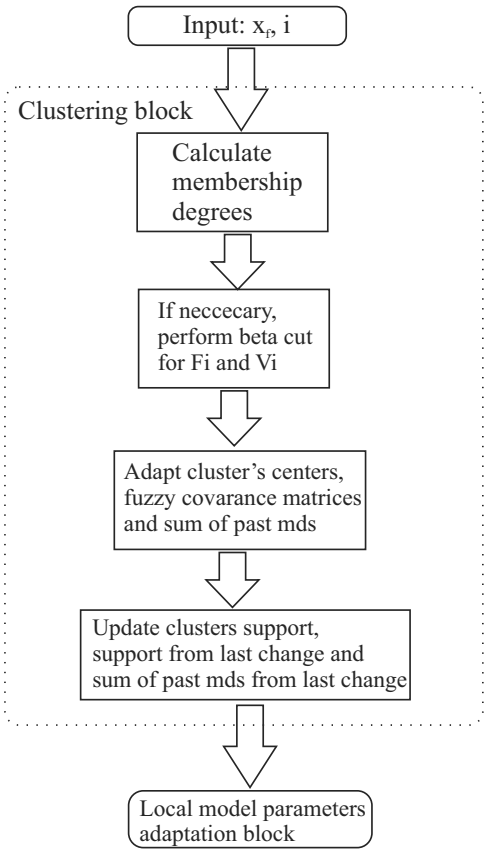


Figure 2: The clustering algorithm.

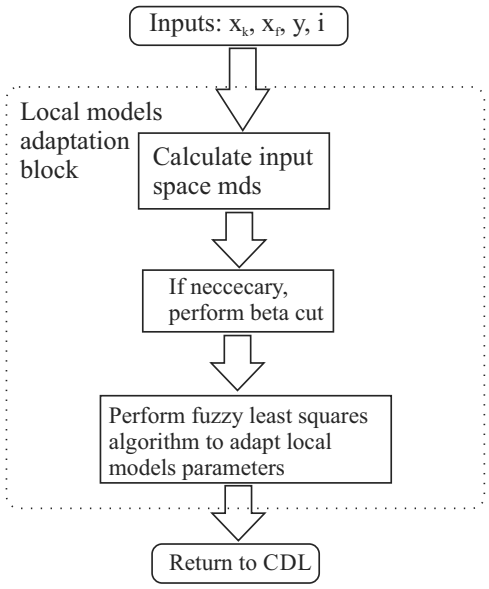


Figure 3: The parameter adaptation algorithm.

were replaced with model outputs:

$$\begin{aligned}\mathbf{x}_f(k) &= [u(k-n)\ \ldots u(k)\ \ y(k-r)\ \ \ldots\ \ y(k-1)]\\ \mathbf{x}_{f_m}(k) &= [u(k-n)\ \ldots u(k)\ \ y_m(k-r)\ \ \ldots\ \ y_m(k-1)]\end{aligned} \tag{23}$$

where y_m is the model output and y is the real output. In both cases the dead zone for adaptation can be considered [31].

The adaptation procedure can be represented by the diagrams as shown on figure 2 and figure 3. In figure 2 the clustering procedure is represented and in figure 3 the local model parameters identification algorithm is presented.

1.2 Evolving mechanisms

To upgrade the fuzzy model structure evolving mechanisms, such as adding and removing the clusters is implemented in the eFuMo method.

Adding mechanism. This is one of the most important mechanisms. It adds new clusters to the fuzzy model structure and improves the fuzzy model performance. In the literature, there are several different conditions of adding new clusters based on model output error, distance of current sample to existing cluster and ε-completeness which is based on current samples membership degree to existing clusters.

In [32] (DFKNN) a cluster adding is based on Euclidian distance to the existing cluster centers and the change of variance that the new sample brings to the closest cluster. The distance and variance change must be greater than the predefined threshold. A new cluster is added if a certain number of sequential samples satisfy this condition. In [11] (DENFIS) adding is based on an Euclidian distance. If the distance of current sample to closest cluster is grater than two times the threshold a new cluster is added. In [20] (D-FNN) and [21] (GD-FNN) adding is based on model error and distance of new sample vector to closest cluster. If both are grater than a user defined threshold the cluster is added. The threshold is decreasing with time. In [17] (GAP-RBF) and [15] (SAFIS) a new cluster is added if the model error and distance of the current sample to existing clusters is over a threshold. They calculate the decrease in error if current sample would be taken for a new cluster. If the decrease is large enough new cluster is created. In [18, 19] (EFuNN) the adding is based on sensitivity calculated based on normalized fuzzy differ-

ence distances. The eTS method [12] adds a new cluster when the potential of current sample is higher than a potential of existing clusters and if it is distanced enough from the nearest cluster. In [33] (NFCN), [22] (ENFM), [8] (SONFIN), [9] (SCFNN), [16] (SOFNN) adding is based on ε-completeness principle. In [13] (FLEXFIS) the adding is based on distance and vicinity quotient.

In practice, the distance conditions work best. Therefore, the eFuMo implements two conditions for adding: the distance conditions and the consequent samples conditions. Both conditions must be satisfied in order for a new cluster to be added. The consequent samples condition is to prevent a new cluster being created based on outlier sample. This condition means that several consecutive samples must satisfy the distance condition before a new cluster is added. The condition is explained in [32].

The distance adding condition is based on a normalized distance. There is an option of choosing the component distances or Mahanalobis distance. The normalized component distances are calculated as:

$$d_{ij} = \frac{|x_{f_j}(k) - v_{i_j}|}{k_n \sqrt{f_{i_{jj}}}}, \quad j = 1,...,p \;\; i = 1,...,c \qquad (24)$$

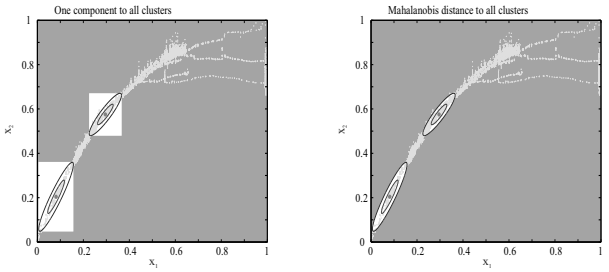
where $x_{f_j}(k)$ is the j-th element of clustering vector, v_{i_j} is the j-th component of i-th cluster center, p is the length of clustering vector, c is the number of clusters, $f_{i_{jj}}$ is the j-th diagonal element of i-th cluster's fuzzy covariance matrix and k_n is the user defined constant, usually set to 2. When using normalized Mahanalobis distance the normalization vector is formed from diagonal elements of fuzzy matrix:

$$\mathbf{s}_{i_{norm}} = [\sqrt{f_{i_{11}}} \;\; \sqrt{f_{i_{22}}} \;\; \cdots \;\; \sqrt{f_{i_{pp}}}]^T, \qquad (25)$$

The normalized distance is then calculated as:

$$d_{i_{norm}} = \frac{((\mathbf{x}_f(k) - \mathbf{v}_i)^T \mathbf{F}_i^{-1} (\mathbf{x}_f(k) - \mathbf{v}_i))^{0.5}}{k_n (\mathbf{s}_{i_{norm}}^T \mathbf{F}_i^{-1} \mathbf{s}_{i_{norm}})^{0.5}} \qquad (26)$$

With the first condition, a cluster can be added is any of the component distance equation 24 is larger than 1. The same component distance must be larger than 1 for all existing clusters. With the second condition a cluster can be added if the distances equation 26 to all clusters are larger than 1. Figures 4(a) show the possible adding space for the component distance conditions and figure 4(b) for the Mahanalobis distance condition. Both figures show the possible adding space (orange) for two



(a) Component adding distance.

(b) Mahalanobis adding distance.

Figure 4: Different adding distance conditions.

dimensional space. When a cluster is added, the parameters of the cluster must be initialized. The center of a new cluster is set at the position of current clustering vector. The fuzzy covariance matrix is initialized as diagonal matrix where the distances to closest cluster are considered. The diagonal elements are defined as:

$$f_{new_{jj}} = -\frac{d_{ij}^2}{2\eta_m \ln(\varepsilon_\beta)}, \qquad (27)$$

where ε_β is a user defined constant, normally set to 0.15. If the distance d_{ij} is smaller than standard deviation ($\sqrt{f_{i_{jj}}}$), then this diagonal element is equal to a diagonal element of the closest cluster's fuzzy covariance matrix ($f_{new_{jj}} = f_{i_{jj}}$).

The first cluster is added at the position of the first clustering vector. Its fuzzy covariance is initialized in the similar manner considering the input-output space boundary and expected number of clusters:

$$d_{max_j} = \max(x_j) - \min(x_j), \quad j = 1,...,p \qquad (28)$$

where d_{max_j} is an expected range of j-th element of clustering vector. The influence zone of the j-th component is then calculated as:

$$d_{influence_j} = \frac{d_{max_j}}{2\,c}, \quad j = 1,...,p \qquad (29)$$

where c is the expected number of clusters. The diagonal j-th element of fuzzy covariance matrix is then calculated as:

$$\sigma_j^2 = -\frac{d_{influence_j}^2}{2\eta_m \ln(\varepsilon_\beta)} \quad j = 1,...,p \qquad (30)$$

The fuzzy covariance is built with σ_j^2 as:

$$\mathbf{F}_i = \begin{bmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_p^2 \end{bmatrix} \tag{31}$$

The parameters of new local model can be initialized using weighted mean:

$$\theta_{i+1_j} = \frac{\sum_{i=1}^{c} \omega_{i_j} \theta_{i_j}}{\sum_{i=1}^{c} \omega_{i_j}} \tag{32}$$

where i is the index of cluster and j is the parameter index. Weights ω_{i_j} can be equal to normalized firing levels of clusters, or equal to normalized firing levels of clusters combined with parameters variances:

$$\omega_{i_j} = \beta_i \frac{1}{\sigma^2_{P_{i_{jj}}}}, \tag{33}$$

where $\sigma^2_{P_{i_{jj}}}$ is the j-th diagonal element of least squares covariance matrix of i-th cluster.

Removing mechanism. It is meant to remove old clusters and clusters created based on outliers. In eFuMo method, this mechanism is not so important as the method incorporates the forgetting factors that ensure the adaptation of the structure to the new data. However, it may happen that a cluster is created in a partition of input-output space that doesn't have much samples and is not very important for the model accuracy. This mechanism ensures that these clusters are removed from the model structure. In literature, different ideas are presented. In [34] the cluster is removed if in a certain time the cluster doesn't receive any support sample. Cluster receives a support sample if it has greater firing level than other clusters. This might be a problem with industrial processes, where it might happen that the process is in one working point for a longer period of time. In this case, other clusters, that describe different working points, might be removed from the structure. In [20], [21], [17], [15] in [16] (D-FNN, GD-FNN, GAP-RBF, SAFIS in SOFNN) the removing is based on model error. In [20] (D-FNN) the *error reduction* ratio is introduced. The amount of error, that a certain cluster brings to the overall model error is calculated. If this is small, the cluster is considered as redundant and is therefore deleted. Similar concept is used in [21] (GD-FNN), where *sensitivity index* is introduced. In [15] (SAFIS) an equation is introduced to estimate the error change if a certain cluster is removed from the structure. If this change is small, the cluster is removed. In [16] (SOFNN) removing is based on *optimal brain surgeon approach* [35, 36]. In [37] (Neural gas) the clusters are removed based on their age. All clusters that are older than an user defined age are removed from the structure. In [38, 39] (exTS) the removing is based on cluster's support and cluster's age. The clusters are removed based on support-age ratio. Similar conditions are introduced in +eTS [23], where also the *utility* condition is added. This condition is based on the ratio of sum of firing levels and age of cluster. The threshold values are defined with standard deviation and mean values of the ratios. In general, this is not adequate, since there is usually small number of clusters; therefore, using standard deviation and mean value are not really representative. In +eTS also minimal existence condition is introduced. With this condition, a newly created cluster must gather a certain amount of support samples in a certain time period after creation. If the gathered support is lower than a predefined threshold, the cluster is removed from the structure. In [18, 19] (EFuNN) the removing is based on cluster's age and sum of cluster's firing levels. In [32] (DFKNN) removing is based on minimal support and time period. If the cluster has lower support than an user defined threshold the cluster is deleted. The cluster is also removed if in certain time period after cluster's creation, no support sample is assigned to it.

The proposed eFuMo method has two conditions for removing: A minimal existence condition and support-age ratio condition. The minimal existence condition is the same as in [23]. It simply removes clusters that in certain period after creation (k_{delay}) don't receive enough support samples (Ns_i). The time period (k_{delay}) and support threshold (Ns_{trh}) are user defined constant usually set to 20 and 10, respectively. The support-age ratio condition is based on clusters' supports Ns_i normalized with clusters' age (equation 35). Cluster with the ratio lower than a percent ε of mean ratio is deleted. Age a_i is defined as a number of samples from the cluster's creation k_i and current sample k:

$$a_i = k - k_i \tag{34}$$

$$S_{n_i} = \frac{Ns_i}{a_i}. \tag{35}$$

Both conditions for removing can be written as:

$$\begin{aligned} &\textbf{IF } S_{n_i} < \varepsilon \text{ mean}(S_n) \\ &\textbf{OR } \left(Ns_i < Ns_{trh} \textbf{ AND } k > k_i + k_{delay}\right) \\ &\textbf{THEN } \text{remove } i\text{-th cluster.} \end{aligned} \tag{36}$$

Splitting mechanism. It is in our case meant for fine tuning the evolving fuzzy model. It can add clusters in input-output space, where the output model error is higher than predefined threshold. The concept of splitting was used in the on-line incremental learning of Gaussian Mixture Models in [40], where the Chernoff bound is used and in [41], where fidelity measure is used. It is argued in [42] that these methods are slow and don't produce good results. Therefore they propose an integrating a joint incremental on-line split-and-merge scenario, that should overcome under- and over-clustered partitions. The splitting is based on a BIC value. The BIC is a combination of Gaussian density function function and cluster overlapping. The clusters that are split are found using trail and error procedure. In [10] (NeuroFAST) clusters are split based on mean squared error (MSE). The error is checked every P step. The cluster that has the highest MSE and is at least P-times activated is split.

The eFuMo's splitting mechanism is based on relative model error, that clusters gather over time. The error is updated every time the splitting mechanism is called and the current sample doesn't satisfy the distance adding condition. First the relative model error is calculated:

$$e(k) = \frac{|y_m(k) - y(k)|}{3.4\sigma_y}, \tag{37}$$

where y is the real output and y_m is the model output. The σ_y is calculated by CDL block and represents current standard deviation of the process output. The error is then divided among the existing clusters and added to the previous error:

$$e_{sum_i}(k) = e_{sum_i}(k-1) + \beta_i e(k), \tag{38}$$

where β_i is the firing level of i-th cluster. The splitting mechanism checks the cluster with the highest error. If its support from the last change in cluster number till now is higher than a threshold (usually set to 20) and its error normalized with N (number of samples used to calculate the error) is larger than a threshold value, the cluster is split. The error threshold is set by the user, specifying the maximal and minimal error threshold and the decay constant. The current threshold is calculated as:

$$e_{trh} = max(e_{max} exp(-N/T), e_{min}), \tag{39}$$

where e_{trh} is the current threshold, e_{max} is the maximal error threshold, e_{min} is the minimal error threshold, N and T are the number of samples that are used for error calculation and decay constant, respectively.

The positions of the split clusters are calculated using diagonal elements (vector $\mathbf{s}_{i_{norm}}$) of the fuzzy covariance matrix.

$$\begin{aligned} \mathbf{v}_{i1} &= \mathbf{v}_i + 0.5\mathbf{s}_{i_{norm}} \\ \mathbf{v}_{i1} &= \mathbf{v}_i - 0.5\mathbf{s}_{i_{norm}} \end{aligned} \tag{40}$$

where i is the index of the cluster that is split. The new center positions can also be calculated using the singular value decomposition as in [43]. The fuzzy covariance matrix, support and sum of past membership degrees are set to half of their original value for both clusters. The time of cluster creation is for both clusters initialized as the creation time of the original cluster.

Merging mechanism. There are two types of merging algorithms implemented in eFuMo: supervised and unsupervised. In literature different concept of merging techniques can be found. In [32] (DFKNN), the center positions are monitored. If the centers are converging to the same area the clusters are merged. The used similarity measure is based on samples membership degrees and is similar to the correlation between clusters firing levels. It is presented in detail in [44]. In [18] (EFuNN), the merging is done based on clusters' firing levels correlation. The method merges neighborhood clusters, where after merging the total radius does not exceed the predefined maximal radius. In [22] (ENFM),the clusters are merged if the membership degree of the first cluster to the second and vice versa is higher than a predefined threshold. In [16] (SOFNN) clusters are merged if the cluster centers of the two clusters are the same. The possibility of using similarity measure from [45] is mentioned. In [46] (FLEXFIS+), the merging based on membership function intersections is proposed and the overlapping index is calculated. If this index for the two clusters is higher than a predefined threshold and the angles between the local models' parameters are small the clusters are merged.

The eFuMo unsupervised merging is based on most commonly used principle of merging. It merges clusters

that are close together. The similarity and the vicinity of the two clusters are measured by the normalized distance:

$$d_{ik}^2 = (\mathbf{v}_i - \mathbf{v}_k)^T F_i^{-1} (\mathbf{v}_i - \mathbf{v}_k), \;\; i,k = 1,...,c \;\; i \neq k. \quad (41)$$

$$d_{norm_{ik}} = \sqrt{\frac{d_{ik}^2}{2\mathbf{s}_{i_{norm}}^T F_i^{-1} \mathbf{s}_{i_{norm}}}} \quad (42)$$

The distances are calculated only for clusters that have higher support from last change in cluster number than an user defined threshold (usually set to 20 for both values). The clusters are considered for merging if both normalized distances $d_{norm_{ik}}$ and $d_{norm_{ki}}$ are shorter than the predefined threshold ε_β:

$$d_{norm_{ik}} < \sqrt{-\ln(\varepsilon_\beta)} \quad (43)$$

If this criterion is satisfied, the distance ratio is checked:

$$\left| 1 - \frac{min(d_{norm_{ik}}, d_{norm_{ki}})}{max(d_{norm_{ik}}, d_{norm_{ki}})} \right| < k_{d_{merge}} \quad (44)$$

if the ratio is above the user defined threshold $k_{d_{merge}}$ (usually 10 percent) clusters are merged.

The parameters of new cluster are initialized as a weighted mean. The fuzzy covariance as proposed in [22]:

$$\begin{aligned} \mathbf{F}_{new} = \frac{1}{(Ns_i + Ns_k)^3} ((Ns_i^3 + 2Ns_i^2 Ns_k + Ns_i Ns_k^2)\mathbf{F}_i + \\ + (Ns_k^3 + 2Ns_k^2 Ns_i + Ns_k Ns_i^2)\mathbf{F}_k + \\ + (Ns_i^2 Ns_k + Ns_i Ns_k^2)(\mathbf{v}_i - \mathbf{v}_k)(\mathbf{v}_i - \mathbf{v}_k)^T) \end{aligned} \quad (45)$$

The new center is calculated as:

$$\mathbf{v}_{new} = \frac{Ns_i \mathbf{v}_i + Ns_k \mathbf{v}_k}{Ns_i + Ns_k} \quad (46)$$

In the same manner a the new sum of past membership degree is calculated. New support of the cluster 47 and time of creation are calculated as weighted mean where weights are sum of past membership degrees (s_i, s_k):

$$Ns_{new} = \frac{Ns_i s_i + Ns_k s_k}{s_i + s_k}. \quad (47)$$

The local linear model parameters are calculated as weighted mean:

$$\theta_{new_j} = \frac{\omega_{i_j} \theta_{i_j} + \omega_{k_j} \theta_{k_j}}{\omega_{i_j} + \omega_{k_j}} \;\; j = 1,...p, \quad (48)$$

where weights ω are the cluster supports Ns combined with a variance of the parameters.

The supervised merging considers the prediction model error. The supervised merging has three different measures to detect the clusters that could be merged together. It uses angles between local models' parameters (angle merging condition), correlation between clusters firing levels (correlation merging condition) and distance ratio (distance ratio merging condition). Only clusters that gathered higher support and sum of past membership degrees than a predefined threshold can be considered for supervised merging. The correlation coefficient is calculated based on monitoring of firing levels and their products $\beta_{ij}(k) = \beta_{ij}(k-1) + \beta_i(k)\beta_j(k)$, $\beta_{ii}(k) = \beta_{ii}(k-1) + \beta_i(k)\beta_i(k)$ and is calculated as:

$$C_{ij}(k) = \frac{\beta_{ij}}{\beta_{ii}^{0.5} \beta_{jj}^{0.5}} \quad (49)$$

If the coefficient $C_{ij}(k)$ is above user-defined threshold (usually set to 0.9) the clusters i and j are considered for merging.

The distance ratio criterion for merging is similar than with the unsupervised merging. The distance ratio is calculated as:

$$\begin{aligned} d_{ik} = \sqrt{(\mathbf{v}_i - \mathbf{v}_k)^T \det(F_i)^{\frac{1}{p}} F_i^{-1} (\mathbf{v}_i - \mathbf{v}_k)} \\ K_d \frac{|1 - min(d_{ik}, d_{ki})|}{max(d_{ik}, d_{ki})} \end{aligned} \quad (50)$$

The clusters are considered for merging if the distance ratio K_d is lower than an user defined threshold (usually 0.05) and the correlation coefficient is at least half of the threshold defined for correlation merging condition.

The angle merging criterion is based on local models' angles. First the parameters are normalized. The algorithm sweeps all local models' parameters to find the vector of the largest absolute value of parameters. Then the parameters of local models are normalized with this vector. The angles for the two clusters for all parameters are calculated:

$$\alpha_{ij_k} = |\arctan(\theta_{i_k}) - \arctan(\theta_{j_k})| \quad (51)$$

where k is the parameter index. The clusters are consid-

ered for merging if all angles $\alpha_{ij_k}, k = 1,..,r$, where r is the number of local model's parameters, are below the user-defined threshold (usually set to 2 degrees) and the correlation coefficient is at least half of the threshold defined for correlation merging condition.

After the eFuMo identifies the possible merging pairs with the correlation, angle and distance ratio merging conditions it then checks the local models for the error:

$$\begin{aligned} \mathbf{x}_1 &= [1, \bar{u}_1, ..., \bar{u}_{p-1}]^T \\ e_1 &= |\theta_i^T \mathbf{x}_1 - \theta_j^T \mathbf{x}_1| \\ e_2 &= \sum_{r=1}^{p} |\theta_{i_r}(x_{1_r} + 2\sigma_{u_{r-1}}) - \theta_{j_r}(x_{1_r} + 2\sigma_{u_{r-1}})| \\ e_3 &= \sum_{r=1}^{p} |\theta_{i_r}(x_{1_r} - 2\sigma_{u_{r-1}}) - \theta_{j_r}(x_{1_r} - 2\sigma_{u_{r-1}})| \\ e &= \frac{1}{10.2\ \sigma_y} \sum_{r=1}^{3} e_r \end{aligned} \tag{52}$$

where $\bar{u}$ is the mean value of a certain input variable $\sigma_{u_{r-1}}$ is its standard deviation, σ_y is the standard deviation of the process output, $p-1$ is the number of inputs, j and i are the cluster indexes θ_i is the i-th cluster's local model parameter vector and θ_{i_r} is the r-th parameter of the i-th local model.

The pair that has the lowest error and the error is below the threshold is merged. The center of the merged cluster is positioned in the middle between maximum and minimum border of both clusters:

$$\begin{aligned} \mathbf{d}_1 &= \mathbf{v}_i - \mathbf{v}_j \\ \mathbf{v}_i' &= \mathbf{v}_i + \text{sign}(\mathbf{d}_1)\mathbf{s}_{i_{norm}} \\ \mathbf{v}_j' &= \mathbf{v}_j - \text{sign}(\mathbf{d}_1)\mathbf{s}_{j_{norm}} \\ \mathbf{d}_2 &= \frac{\mathbf{v}_i' - \mathbf{v}_j'}{2} \\ \mathbf{v}_{new} &= \mathbf{v}_j + \mathbf{d}_2 \end{aligned} \tag{53}$$

The fuzzy covariance matrix and support of a new merged cluster is initialized as a sum of both clusters' fuzzy covariance matrices and supports, local model parameters are initialized as a mean of both local models' parameters and the creation time is initialized to the creation time of the oldest cluster. The sum of past membership degrees is initialized to the max sum of past membership degrees of both clusters.

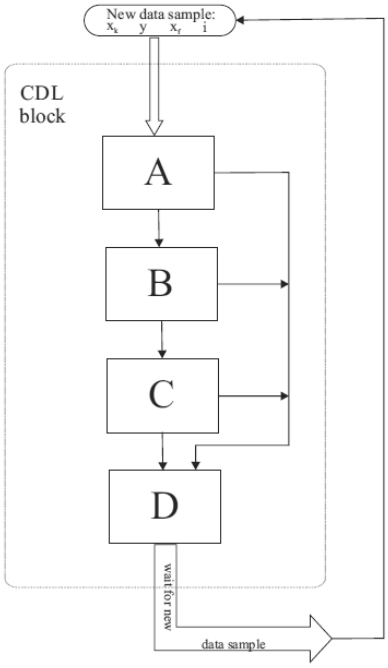


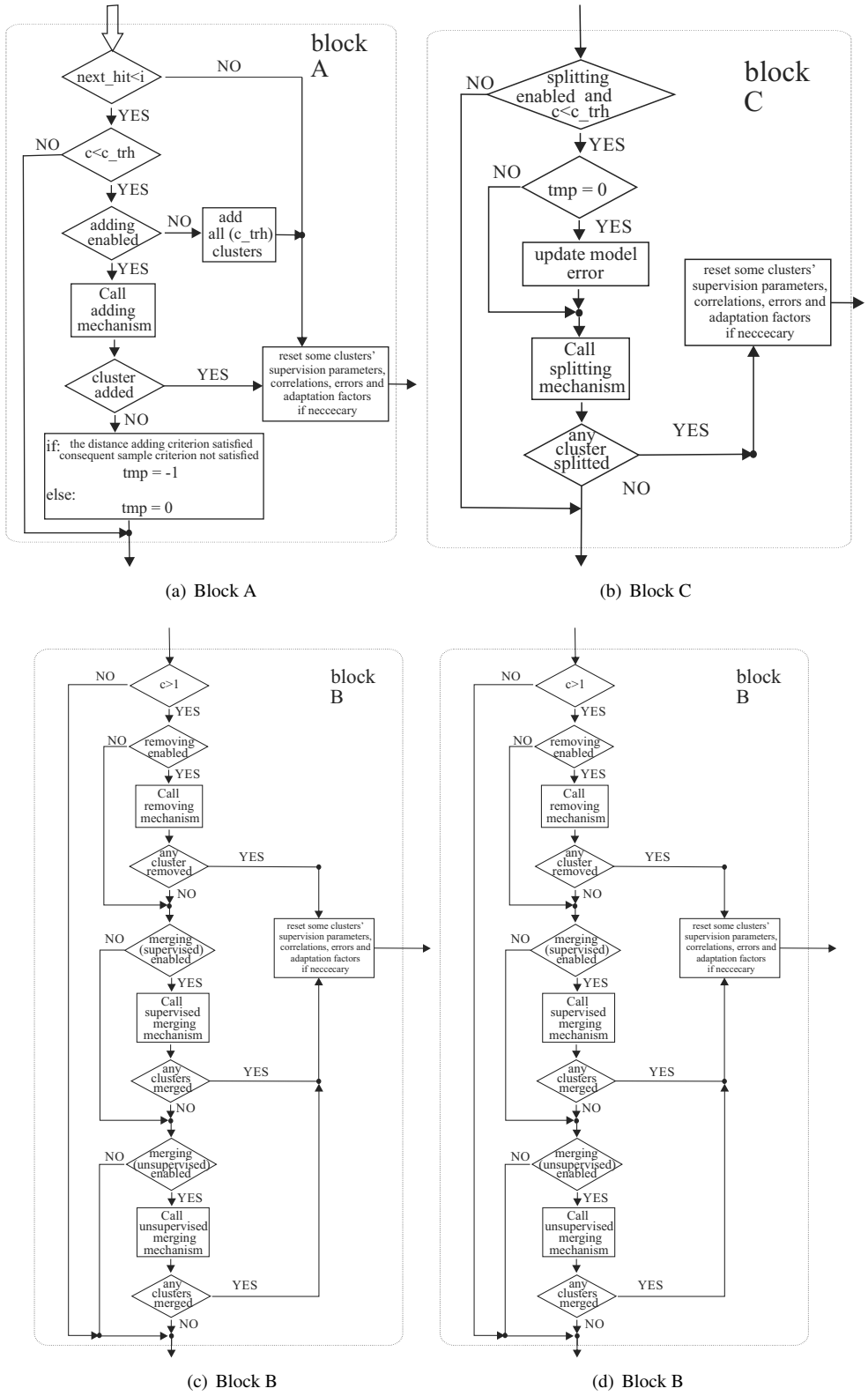
Figure 5: Scheme of the CDL.

1.3 Central decision logic

The CDL is responsible for proper flow of the operations. It controls the calls to evolving mechanisms and adaptation mechanisms. It also calculates the mean and standard deviations of the inputs and output of the process, that is identified with eFuMo. The scheme of the CDL block is shown on figure 5 and the sub-blocks are shown on figure 6.

The input to the eFuMo identification method are clustering vector ($\mathbf{x}_f$), regression vector ($\mathbf{x}_k$), output of the process (y) and number of current sample (i). The CDL block first checks the current sample number (i) to the sample number when the last change in cluster number was made and the user defined time delay. If the sum of these two values are smaller than a current sample number, the evolving mechanisms may be called. Otherwise the CDL skips the call to evolving mechanisms.

The CDL first calls the adding mechanism, then the removing mechanism, follows the supervised merging mechanism and unsupervised merging mechanism and at the end the CDL calls the splitting mechanism. If one of the mechanisms changes the cluster number other evolving mechanisms that follow are not called and the eFuMo continues with the adaptation algorithm.



(a) Block A (b) Block C

(c) Block B (d) Block B

Figure 6: The CDL scheme blocks. The c is the number of clusters, c_trh is the maximal allowed number of clusters, age_trh is the age threshold for minimal existence condition and $last\_change$ is the sample number when the last change in cluster number occurred.

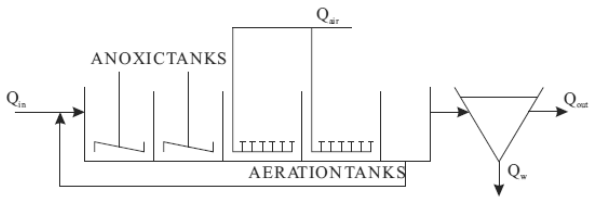


Figure 7: Scheme of the MBBR.

The CDL algorithm is also responsible for calculating the variance and mean of the input variables and output. The variance σ^2 is calculated on line by the following equation:

$$\sigma_x^2(k) = \frac{1}{k}\left((k-1)(\sigma^2(k-1)+\bar{x}(k-1)^2)+x(k)^2\right) - \\ -\frac{1}{k^2}\left((k-1)\bar{x}(k-1)+x(k)\right)^2 \tag{54}$$

where x is the variable and $\bar{x}$ is the mean of it, calculated as:

$$\bar{x}(k) = \frac{1}{k}\left((k-1)\bar{x}(k-1)+x\right) \tag{55}$$

If the splitting is enabled, the CDL also calls the error update algorithm. The algorithm updates a cluster error equation 38. This algorithm is only called if the current data sample doesn't satisfy the distance adding condition. The CDL also calculates the clusters' firing levels products used to calculate the correlation coefficient 49.

2 Monitoring Example

2.1 Monitoring system idea

The monitoring system that includes the evolving fuzzy model was tested on measured data from a pilot wastewater treatment plant, shown in figure 7. The pilot plant consists of two anoxic reactors, two aerobic reactors and an additional reactor, where the water is collected before returning as an internal recycle or passing down to the settler. To ensure the homogeneity, the waste water is mixed by mixers in the anoxic reactors and by air flow in the aerobic reactors. In this example the monitoring of oxygen concentration in anoxic reactors will be done. The monitoring system is based on Takagi-Sugeno (TS) fuzzy model that estimates the relations between the input and output variables. The oxygen concentration is estimated from the air flow, the temperature in the reactor and the previous measurement of the oxygen concentration. First order local models are used. The inputs were selected by a backward selection. The idea is to detect the error in the process output based on the inputs. The outputs of the FDS are $y_{soft}(k)$ and $alarm(k)$. The output $alarm(k)$ indicates the presence of the fault in the measured signal ($alarm(k) = 1$: fault detected). The output $y_{soft}(k)$ is the process output with the removed fault. If there is no fault detected the output $y_{soft}(k)$ is equal to the process output $y(k)$. If the fault is detected, the output $y_{soft}(k)$ is calculated based on a fuzzy model that describes the proper relations between the input signals and the monitored signal.

The FDS determines the fault based on the internal fuzzy model of the signal relations. For monitored signal, three models are kept in the FDS's memory: a full evolving fuzzy model, an adaptive fuzzy model (parameters of clusters and local models are adapted) and a fuzzy model with fixed parameters that holds the information about the last good known parameters. The learning of the fuzzy models is delayed for 200 samples. The delay was introduced for future research to cope with slower faults. The data sample is used for learning if there was no fault detected. For each sample and each model the relative prediction error is calculated. The calculated error (its absolute value) is assigned to the model. The prediction error assigned to the fuzzy model is combined with the simulation error, which is calculated periodically on every 200-th sample using the 200 samples in the buffer. The prediction error is also used for learning the prediction-error fuzzy model. Namely, each model that describes the signal relations is accompanied by the error model. The error model is used to calculate the allowed difference between the estimated and measured signals. For estimating the sensor output during the failure, the model with the lowest assigned error is used.

The adaptive and fixed model structure and parameters can be replaced when a cluster is added or removed from the evolving fuzzy model's structure. Before the number of cluster changes, the error of each model is checked. If the evolving model has the smallest error, the adaptive and fixed model structure is replaced by the evolving model's structure. In addition their error models are replaced. The simplified diagram of the procedure is shown in figure 8.

The variances denoted as $\sigma\_evolving$, $\sigma\_adaptive$ and σ_fixed are calculated from the error model:

$$\sigma = \sum_{i=1}^{c} \beta_i \sqrt{F_{i_{r,r}}}, \tag{56}$$

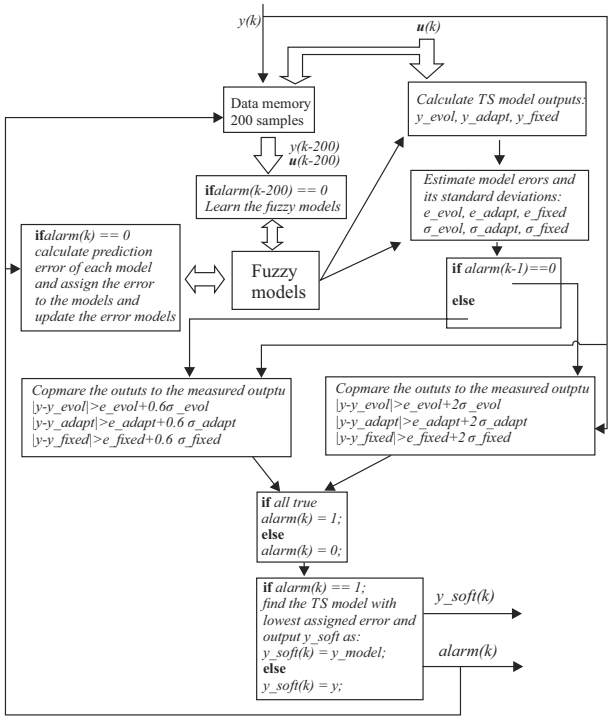


Figure 8: Scheme of the FDS for a subprocess.

where $F_{i_{r,r}}$ is the last diagonal element of the error fuzzy model's cluster i. This element represents the variance of the error. As seen in figure 8, the alarm is raised if the difference between the estimated output and the measured output is higher than the maximum allowed difference. Note that the alarm is turned off when for at least 30 consecutive samples the difference is below the defined threshold for turning off the alarm. To ensure a smooth transition from the estimated output to the measured output, when the alarm is turned off a filter was implemented that calculated the output of the FDS as:

$$y_{soft} = \frac{((30 - k_{alarm})y_{model} + k_{alarm}y)}{30}, \qquad (57)$$

where k_{alarm} is the number of samples from the sample when the condition for turning the alarm off was reached. The maximum number of k_{alarm} is 30 and its value is reset to 0 every time a new alarm is raised.

2.2 Detecting the false alarms due to manual calibration

Manual tuning and offset repairs of the oxygen concentration signal is performed every few months. This is seen on the upper graph in figure9. The drift of the sensor was manually reduced by the operator, causing the FDS to report an error. It can be seen that the shapes of the estimated and measured outputs are practically the same. However, due to an offset of the signal the FDS detects the error. To automatically turn off such alarms, an additional algorithm was implemented to the FDS. This algorithm is turned on when a new alarm is detected. With this procedure the algorithm starts to calculate the variances of the estimated output, the measured output and the variance of their difference when the alarm is raised. The idea behind this solution is that the variance of the estimated and measured output (if they are only shifted) should be higher than the variance of their difference, under the assumption that the model used for estimating the output is not biased and the process output changes (there is an excitation present). The variances are calculated recursively with equations 55 and 54. When the variance of the difference between the estimated and measured outputs falls under the variance of both, the estimated and the measured outputs the raised alarm is turned off. The algorithm starts to check this condition after the alarm is present for some time (in our case 300 samples). The algorithm is turned off when, for at least a certain number of consecutive samples (in our case 100), the variance of error is below the model and process variance. The algorithm is also turned off if its maximum functioning time is reached.

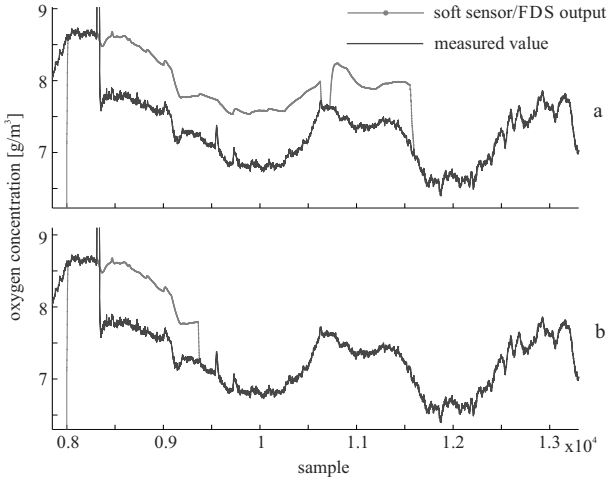


Figure 9: Effect of sensor calibration.

3 Results and Discussion

The presented idea was tested on real data. To estimate the performance of the system during a sensor's

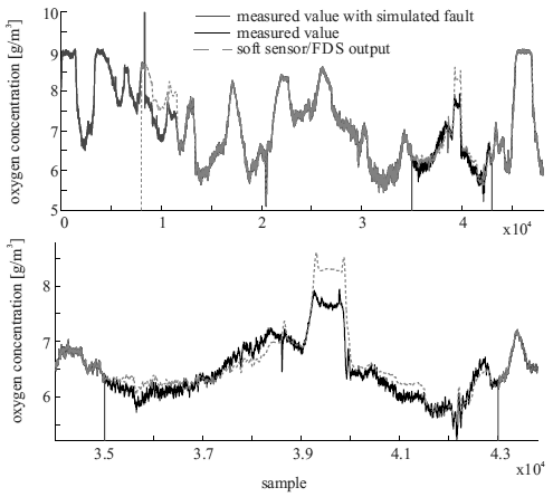


Figure 10: Oxygen-concentration fault detection.

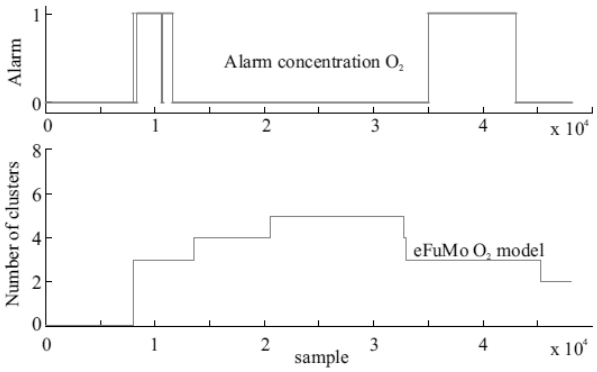


Figure 11: Alarm signal and number of clusters over the experiment.

Estimation Error	Concentration O_2
NDEI	0.488
min. abs.	2.83e-5 $[g/m^3]$
max. abs.	1.347 $[g/m^3]$
avg. abs.	0.189 $[g/m^3]$
signal range	2.72 $[g/m^3]$
min. rel.	1.04e-5
max. rel.	0.495
avg. rel.	0.0695
faulty samples $[10^3]$	$35-43$

Table 1: Estimation error during the simulated fault.

malfunction a failure was simulated on a known part of the data. Note that the duration of the simulated fault was exaggerated in order to test the system. The simulated faults lasted for about 7000 samples (around 39 hours). Usually, the faults last from about a few minutes up to 6 hours. The settings of the evolving method were obtained based on trail and error. The fault was simulated between the samples 35000 and 43000. The whole experiment is shown in figure 10. The first 8000 data points were used for the initial learning of the fuzzy model. The learning was performed using the eFuMo method. The alarm signal and the number of fuzzy model clusters are shown in figure 11. Besides the simulated fault, the system also detected some faults that were not added to the signals. These faults were caused by sudden spikes in the monitored signals and therefore the detection of the fault seems justified.

Even though the estimated signal is not entirely covering the measured signal, we believe that the estimation accuracy is still good enough. The error between the measured and estimated signal during the fault is given in Table 1. This table also includes the NIDE index, the minimum, maximum and mean absolute error, the signal range for the faulty samples, the minimum, maximum and mean relative error, and the samples where the fault was simulated are given.

As can be seen on the upper graph in figure 9, the manual tuning creates an offset of the measured signal, resulting in the detection of a fault. At around sample 8400 a real fault occurs, which then quickly vanishes. Later on the measured signal is shifted. The FDS detects the alarm. Because the signal is shifted after the fault, the alarm is still present. The alarm is finally turned off at sample 11500, when the measured signal comes into the allowed difference zone and stays there long enough for the fuzzy model to adapt itself to the signal shift. On the lower graph in figure 9, the false-alarm detection was implemented. It can be seen that the signal shift is successfully detected and the alarm is turned off more quickly than without the implemented false-alarm detection algorithm.

On figure 12, the course of variances are shown. The variance of the difference (between the estimated and measured signal) falls under the measured signal's variance very quickly. This is partly because the initial fault of the measured signal is included in the variance calculation. The variance of the difference falls under the variance of the estimated signal at sample 9475. With

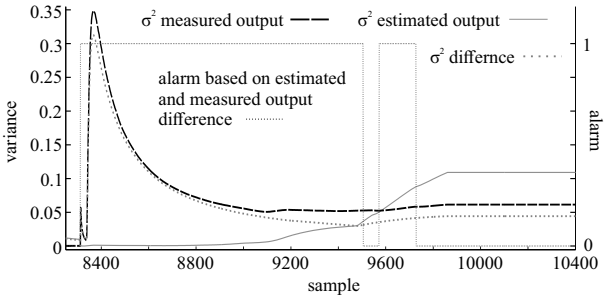


Figure 12: The course of variances.

this, the conditions for overriding the original alarm are met. The last alarm based on the output differences is raised at sample 9740. Therefore, the variance procedure is switched off at sample 9840. The procedure successfully detected the signal offset caused by the manual calibration.

4 Conclusion

In this paper an evolving fuzzy model method for on-line learning of fuzzy models was presented. The method is useful when dealing with nonlinear time-varying processes. The method was used in an example of fault detection system. The presented results show that the approach can be successfully used for such tasks. The only issue of the method and all such methods is in its tuning. There are a number of parameters that need to be tuned. Their tuning highly depends on a problem and require an expert to tune them. Further research will be focused on lowering the number of tuning parameters and on self tuning of the method.

References

[1] Johanson TA, Murray-Smith R. *Operating regime approach to nonlinear modeling and control*. UK: Taylor Francis. 1981.

[2] Hwang CL, Chang LJ. Fuzzy neural-based control for nonlinear time-varying delay systems. *IEEE Trans Syst Man Cyber part B*. 2007;37(6):1471–1485.

[3] Shing J, Jang R. ANFIS : Adaptive-Network-Based Fuzzy Inference System. *IEEE Trans Syst Man Cyber*. 1993;23(3):665–685.

[4] Azeem MF, Hanmandlu H, Ahmad N. Structure identification of generalized adaptive neuro-fuzzy inference systems. *IEEE Trans on Fuzzy Syst*. 2003; 11(5):666–681.

[5] Dovzan D, Skrjanc I. Recursive fuzzy c-means clustering for recursive fuzzy identification of time-varying processes. *ISA Transactions*. 2011; 50(2):159 – 169.

[6] Dovzan D, Skrjanc I. Recursive clustering based on a Gustafson-Kessel algorithm. *Evolving Systems*. 2011; 2:15–24.

[7] Patt J. A resource allocating network for function interpolation. *Neural Computat*. 1991;3(2):213–225.

[8] Juang CF, Lin CT. An on-line self-constructing neural fuzzy inference network and its applications. *IEEE Trans on Fuzzy Syst*. 1998;6(1):12–32.

[9] Lin FJ, Lin CH, Shen PH. Self-constructing fuzzy neural network speed controller for permanent-magnet synchronous motor drive. *IEEE Trans on Fuzzy Syst*. 2001;9(5):751–759.

[10] Tzafestas SG, Zikidis KC. NeuroFAST: On-line neuro-fuzzy ART-based structure and parameter learning TSK model. *IEEE Trans Syst Man Cyber Part B*. 2001;31(5):797–802.

[11] Kasabov NK, Song Q. DENFIS: Dynamic Evolving Neural-Fuzzy Inference System and Its Application for Time-Series Prediction. *IEEE Trans on Fuzzy Syst*. 2002;10(2):144–154.

[12] Angelov PP, Filev DP. An approach to on-line identification of Takagi-Sugeno fuzzy models. *IEEE Trans Syst Man Cyber Part B*. 2004;34(1):484–497.

[13] Lughofer E, Klement EP. FLEXFIS: A Variant for Incremental Learning of Takagi-Sugeno Fuzzy Systems. In: *Proceedings of The 2005 IEEE International Conference on Fuzzy Systems*. Reno, Nevada, USA. 2005; pp. 915 – 920.

[14] M Pratama PA S G Anavatti, Lughofer E. A novel incremental learning machines. *IEEE Transactions on*

Neural Networks and Learning Systems. 2014; 25(1):55–68.

[15] Rong HJ, Sundararajan N, Huang GB, Saratchandran P. Sequential adaptive fuzzy inference system (SAFIS) for nonlinear system identification and prediction. *Fuzzy Sets and Syst*. 2006;157(9):1260–1275.

[16] Leng G, Prasad G, McGinnity TM. An on-line algorithm for creating self-organizing fuzzy neural networks. *Neural Networks*. 2004;17:1477 – 1493.

[17] Huang GB, Saratchandran P, Sundararajan N. A Recursive Growing and Pruning RBF (GAP-RBF) Algorithm for Function Approximations. In: *Proceedings of The Fourth International Conference on Control and Automation (ICCA-03)*. Montreal, Canada. 2003; pp. 10 – 12.

[18] Kasabov NK. Evolving fuzzy neural networks for supervised/unsupervised on-line knowledge-based learning. *IEEE Trans Syst Man Cyber Part B*. 2001; 31(6):902–918.

[19] Kasabov N. Evolving fuzzy neural networks-Algorithms, applications and biological motivation. In: *Methodologies for the Conceptation, Design and Application of Soft Computing*. Japan. 1998; pp. 271–274.

[20] Wu S, Er MJ. Dynamic fuzzy neural networks - A novel approach to function approximation. *IEEE Trans Syst Man Cyber Part B*. 2000;30(2):358–364.

[21] Wu S, Er MJ, Gao Y. A fast approach for automatic generation of fuzzy rules by generalized dynamicc fuzzy neural networks. *IEEE Trans on Fuzzy Syst*. 2001;9(4):578–594.

[22] Soleimani-B H, Lucas C, Araabi BN. Recursive Gath-Geva Clustering as a Basis for Evolving Neuro-Fuzzy Modeling. *Evolving Systems*. 2010; 1(1):59 – 71.

[23] Angelov P. *EVOLVING INTELLIGENT SYSTEMS: Methodology and Applications*, chap. Evolving Takagi-Sugeno Fuzzy Systems From Streaming Data (eTS+), pp. 21 – 50. New Jersey: Wiley. 2010;.

[24] Lughofer E. *Learning in Non-Stationary Environments: Methods and Applications*. Springer.

[25] Kalhor A, Araabi B, Lucas C. An online predictor model as adaptive habitually linear and transiently nonlinear model. *Evolving Systems*. 2010;1(1):29–41.

[26] Rubio J. SOFMLS: Online self-organizing fuzzy modified least-squares network. *IEEE Transactions on Fuzzy Systems*. 2009;17(6):1296–1309.

[27] Lughofer E. On-line assurance of interpretability criteria in evolving fuzzy systems - achievements, new concepts and open issues. *Information Sciences*. 2013; 251:22 – 46.

[28] Lughofer E. *Evolving Fuzzy Systems – Methodologies, Advanced Concepts and Applications*, vol. 266 of *Studies in Fuzziness and Soft Computing*. Springer.

[29] de Jesús Rubio J. *EVOLVING INTELLIGENT SYSTEMS Methodology and Applications*, chap. STABILITY ANALYSIS FOR AN ONLINE EVOLVING NEURO-FUZZY RECURRENT NETWORK, pp. 173–199. Hoboken, New Jersey: John Wiley & Sons. 2010;.

[30] Blažič S, Škrjanc I, Gerkšič S, Dolanc G, Strmčnik S, Hadjiski MB, Stathaki A. Online fuzzy identification for an intelligent controller based on a simple platform. *Engineering Applications of Artificial Intelligence*. 2009;22:628–638.

[31] Dovžan D, Škrjanc I. Predictive functional control based on an adaptive fuzzy model of a hybrid semi-batch reactor. *Control Engineering Practice*. 2010;18(8):979 – 989.

[32] Hartert L, Mouchaweh MS, Billaudel P. A semi-supervised dynamic version of Fuzzy K-Nearest Neighbours to monitor evolving systems. *Evolving Systems*. 2010;1:3–15.

[33] Lin CT. A neural fuzzy control system with structure and parameter learning. *Fuzzy Sets and Syst*. 1995; 70:183–212.

[34] Crespoa F, Weberb R. A methodology for dynamic data mining based on fuzzy clustering. *Fuzzy Sets and Systems*. 2005;150:267–284.

[35] Hassibi B, Stork DG. Second-order derivatives for network pruning: Optimal brain surgeon. *Advances in Neural Information Processing*. 1993;4:164–171.

[36] Leung CS, Wong KW, Sum PF, Chan LW. A pruning method for the recursive least squared algorithm. *Neural Networks*. 2001;14:147–174.

[37] Fritzke B. A growing neural gas network learns topologies. *Adv Neural Inform Processing Syst*. 1995; 7:845–865.

[38] Asif M, Angelov P, Ahmed H. An Approach to Real-time Color-based Object Tracking. In: *Evolving Fuzzy Systems, 2006 International Symposium on*. 2006; pp. 86 –91.

[39] Angelov P, Zhou X. Evolving Fuzzy Systems from Data Streams in Real-Time. In: *Evolving Fuzzy Systems, 2006 International Symposium on*. 2006; pp. 29 –35.

[40] Hall P, Hicks Y. A method to add Gaussian mixture models. *Tech. rep.*, Department of Computer Science, University of Bath. 2004.

[41] Declercq A, Piater J. Online learning of gaussian mixture models - a two-level approach. In: *Proceedings of the 3rd international conference on computer vision theory and applications VISAPP*. Funchal, Portugal. 2008; pp. 605–611.

[42] Lughofer E. A dynamic split-and-merge approach for evolving cluster models. *Evolving Systems*. 2012;DOI: 10.1007/s12530-012-9046-5.

[43] Skrjanc I, Hartmann B, Banfer O, Nelles O, Sodja A, Teslic L. Supervised Hierarchical Clustering in Fuzzy Model Identification. *Fuzzy Systems, IEEE Transactions on*. 2011;PP(99):1.

[44] Frigui H, Krishnapuram R. A robust algorithm for automatic extraction of an unknown number of clusters from noisy data. *Pattern Recognition Letters*. 1996; 17:1223–1232.

[45] Wang WJ. New similarity measures on fuzzy sets and on elements. *Fuzzy Sets and Systems*. 1997;85:305–309.

[46] Lughofer E, Bouchot JL, Shaker A. On-line elimination of local redundancies in evolving fuzzy systems. *Evolving Systems*. 2011;(2):165–187.

Model-based Prediction of the Remaining Useful Life of the Machines

Pavle Boškoski[1*], Boštjan Dolenc[1,2], Bojan Musizza[1], Đani Juričić [1,3]

[1] Department of Systems and Control, Jožef Stefan Institute, Jamo va 39, 1000 Ljubljana, Slovenia;*pavle.boskoski@ijs.si

[2] Jo žef Stefan International Postgraduate School, Jamova 39, 1000 Ljubljana, Slovenia;

[3] University of Nova Gorica, Vipavska 13, 5000 Nova Gorica, Slovenia

SNE Simulation Notes Europe SNE 26(4), 2016, 221 - 228
DOI: 10.11128/sne.26.tn.10353
Received: November 10, 2016
Accepted: December 5, 2016 (Special Issue Review)

Abstract. Accurate anticipation of the remaining useful life (RUL) of a machine is becoming mandatory for efficient exploitation of the asset and avoiding the unplanned downtimes. This should be achieved without extra investments in additional sensors and processing power. In this paper we present an approach to the RUL prediction of a shot blasting machine by using recordings from inexpensive vibrational sensors. The key idea consists in (i) employing generalised Jensen-Rényi divergence (JRD) as a measure of change in the vibrational pattern and (ii) associating JRD with the abrasive wear in rotor blades. It is essential to note that these two show monotonic relationship. Hereupon, a simple hidden Markov model with stochastic inputs and JRD as output is proposed. The hidden states of the model are updated on-line by means of Kalman filter. Prediction of the remaining useful life is done by executing Monte Carlo simulations on the updated model and evaluation of the first passage time of the JRD. The approach is successfully validated experimentally by running the machine up to failure, hence allowing for naturally evolving wear progression and breakdown.

Introduction

Stable and anticipative condition of process equipment, high availability and reliability, along with product quality are key factors that allow companies to stay competitive on the market. However, wear, material stress and environmental factors cause equipment to fail. The problem occurs if that happens unexpectedly, since the consequence can be partial or total breakdown of a production line, destroyed equipment and even catastrophes.

Migrating towards more cost effective condition-based and predictive maintenance (instead of sticking to the outdated concepts of reactive and periodic maintenance) has become a way to raise the overall process performance and cost efficiency. To accomplish this goal, systems for on-line and non-destructive condition monitoring (CM) have to be employed to timely alert about the onset and location of fault in the early stage [1]. Indeed, the degradation of an asset usually goes through a distinct incipient phase with some noticeable indicators, which provide advanced warning about onset of failure. However, what the operators and maintenance people indeed want to know is when to stop the machine and take accommodation actions. Reliable estimate of the remaining useful life (RUL) becomes indispensable.

In spite of significant advances in condition monitoring in the last decade in terms of methodology and key enabling technologies, yet no massive use in industrial sector has been witnessed to date [2]. There are several reasons for that, including (i) (still) relatively high cost of the design and commissioning, especially when domain specific solutions have to be adopted and (ii) the fact that traditional approaches require additional instrumentation (e.g. for rotational speed) to be implemented hence rising the cost.

Compared to CM, predicting RUL is by far more difficult problem. Only limited success has been achieved in special cases like in aeronautics and defence systems. The problem is notoriously demanding for several reasons: (i) data about overall useful life from similar items of equipment are seldom available, (ii) knowledge about degradation, i.e. wear mechanisms is incomplete and (iii) comprehensive knowledge of operating history, disturbances and past maintenance actions is usually unavailable.

The objective of the design approach presented below is to comply with the three main requirements: (i) to come up with signatures sufficiently robust to variations in the operating conditions; (ii) to set up the alarm threshold the required prior knowledge should be minimal (meaning that all the required information should

be extracted from data in fault-free operation) and (iii) to perform condition monitoring (CM) using minimal number of sensors thus making the method both broadly applicable and financially viable.

In this paper we propose an approach to the RUL prognosis based solely on vibrational records. The idea is to exploit the relationship between the degradation phenomena in the material, the remaining life and characteristic information patterns in measured signals. The latter are obtained by statistical signal processing of signals from vibrational sensors in a way to accomplish monotonous dependance with the level of machine degradation. Evaluation of the vibrational features is based on statistical analysis of the envelope of the generated vibration [3]. State of health of the machine is determined from change in the vibrational signature by calculating the "distance" between initial and current signatures. That is achieved by evaluating the generalised Jensen-Rényi divergence of the vibrational features. Since the degradation is stochastic process, we will exploit hidden Markov models to describe the degradation phenomena. The states of the models are updated on-line and then used to simulate propagation of the future degradation and hence evaluate the probability density function of the remaining useful life.

The concept of RUL estimation above is applied to a shot blasting machine.

The rest of the paper is organised as follows. Section 2 introduces the problem related to the degradation of the machines during operation. Simple process model for RUL prediction, complemented with the health index, is presented in Section 3. Experimental results are highlighted in Section 4. The paper ends up with concluding remarks.

1 Shot Blasting Machine

Shot blasting machines are widely used in the process of surface cleaning where contaminants from the surface of castings are removed in order to prepare the metal parts for further finishing like, for example, painting, coating or mechanical treatment.

In shot blasting machines (Figure 1) small shots of abrasive material are fed to the turbine blades where the shots form a stream flowing along the blade length. Depending on the actual arrangement of the separating rotor and the sleeve, the flowing stream will be roughly uniform on the blades' width and length. As soon as the stream of shots leaves the blades, its direction is controlled by setting the wheel, whilst its shape changes both in width and length, thus forming a range of shot flow that hits the surface of object under treatment.

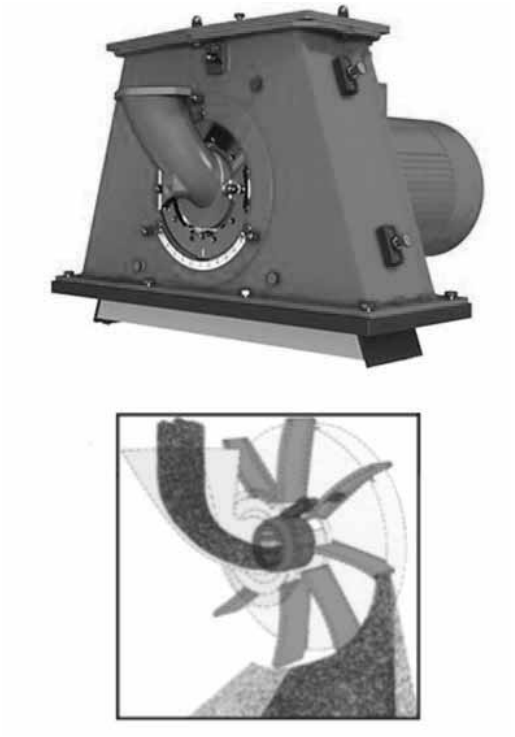


Figure 1: The shot blasting machine and illustration of the principle of operation.

The problem addressed in this paper concerns abrasive wear of the rotor blades. Abrasive grains transverse the blade from center to the periphery and their kinetic energy increases due to centrifugal forces of the rotating blade. Hence the abrasive grains scrap the surface of the blade thus forming "micro-chips", i.e. small pieces of material removed from the blade surface. With increasing number of the operating cycles the wear increases, gradually leading to the damaged blade, which can eventually break and cause downtime.

The outlook of a new blade at the beginning of the process and near failure is given in Figure 2. The problem is that it is not possible to accurately judge the level of wear on the basis of the number of cycles. Therefore it is of utmost interest for the operators to have an indicator on the level of wear in non-intrusive manner, i.e. without interrupting the blasting process.

Inference on the level of damage is done on the basis of signal analysis from vibrational sensor mounted on the housing of the machine close to the rotor bearing.

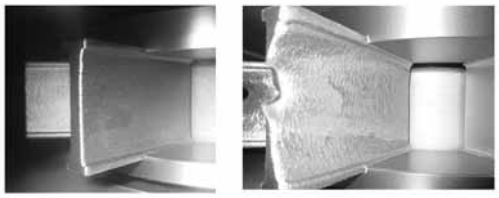


Figure 2: Turbine blade at the beginning of the operation (left) and at the end of the useful life (right).

2 Vibrational features and health index

2.1 Feature extraction from vibrational signal

Faults in the rotational machines affect the inner patterns of vibrational signals referred to as features [3]. By tracking the way these features evolve over time, it is possible to perform sufficiently accurate RUL prediction.

Wear in a turbine blade of the machine gradually results in increased imbalance of the rotor system. Vibrations resulting thereof can be viewed as the result of excitation, caused by rotor movement, on the machine eigen-structure. The resulting spectrum contains characteristic components at the frequencies $m \cdot n_{blades} \cdot f_{rot}$ where $m \in 1, 2, ..., n_{blades}$ is the number of blades and f_{rot} is rotational speed. By applying the narrow-band filtering around the characteristic frequency we get a narrow-band stochastic signal whose energy (or envelope) is Rice distributed.

Sampling of vibrational signal is performed at high frequency during short measurement sessions with an interval of 2 hours between two consecutive sessions. Changes in the probability distribution function (pdf) are characterised by calculating the "distance" between the current pdf and the reference one obtained when the machine is in nominal (healthy) state. Among several possible metrics that can be used to describe this distance, we suggest the so-called f-divergence measures, more precisely the generalised Jensen-Rényi (JR) divergence [4]. The rationale is simple. Instead of comparing two distributions, we compare two *ensembles* of distributions, one from fault-free reference condition and the other from current condition. The strength of this approach lies in the fact that comparing only two distributions is subjected to considerable fluctuations, which make final decision making difficult.

2.2 Jensen-Rényi divergence

The generalised Jensen-Rényi divergence (JRD), denoted by JR^w_α serves to quantify the dissimilarity among n pdfs $\mathscr{P}_1, ..., \mathscr{P}_n$. It reads:

$$JR^w_\alpha(\mathscr{P}_1, \ldots, \mathscr{P}_n) = H_\alpha\left(\sum_{i=1}^{n} w_i \mathscr{P}_i\right) - \sum_{i=1}^{n} w_i H_\alpha(\mathscr{P}_i) \tag{1}$$

where $\sum_{i=1}^{n} w_i = 1$ and H_α is the Rényi entropy:

$$H_\alpha(\mathscr{P}) = \frac{1}{1-\alpha} \ln \sum_{x \in \mathscr{D}} p^\alpha(x). \tag{2}$$

with $\alpha \in [0, 1]$.

The selection of weights w_i in (1) is in principle arbitrary. If w_i are selected uniformly i.e. $w_i = 1/n$, the divergence reaches maximal value [5]. JR divergence quantifies shared information among n random variables. If they are identical, i.e. $\mathscr{P}_1 = \mathscr{P}_2 = \ldots = \mathscr{P}_n$, the divergence is zero.

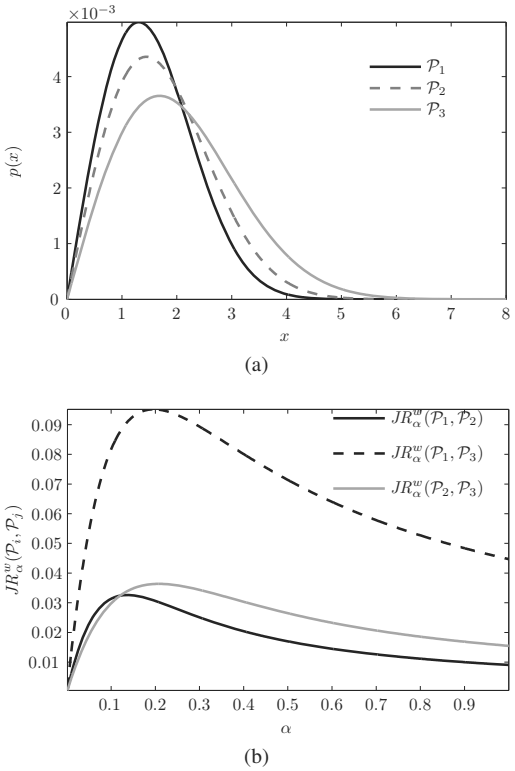


Figure 3: (a) Pdfs three random signals, and (b) pairwise JR divergence as a function of α.

The usability of the JR divergence concept can be described with a simple example. Figure 3(a) shows

three pdfs of Rician family. By considering the pairwise JR divergence with uniform weights, the relation (1) becomes:

$$JR^{w}_{\alpha}(\mathscr{P}_1,\mathscr{P}_2) = H_\alpha\left(\frac{1}{2}(\mathscr{P}_1+\mathscr{P}_2)\right) - \frac{1}{2}\left(H_\alpha(\mathscr{P}_1)+H_\alpha(\mathscr{P}_2)\right), \tag{3}$$

where $\mathscr{P}_1$ and $\mathscr{P}_2$ are pdfs of interest. As shown in Figure 3(b), the JR divergence corresponds to the dissimilarity between corresponding pdfs.

Figure 3(b) additionally shows the effect of the values of the parameter α. Low value of $\alpha \approx 0$ emphasizes dissimilarity among pdfs in the lower part of the range of random variable (approximately $x \in (6,8)$) where pdfs do not differ much, hence low divergence values. In the middle region ($x \approx 4$, $\alpha \approx 0.2$) the pdfs differ the most, hence the highest values of JR divergence. Finally, $\alpha \in (0.6,1)$ captures the region of the bulk probability masses and the divergence drops in a relatively linear manner.

2.3 The role of weights w_i

To allow tracking the changes in pdfs, the exponential weights w_i are suggested in this paper. The weights w_i are calculated using the exponential function of the following form:

$$w_i = C \cdot e^{-\frac{\lambda}{n} i} \tag{4}$$

where λ is sensitivity parameter, n is the number of pdfs (1), $i = 1,2,\ldots,n$ and C is normalising constant. One can easily see that (4) reduces to the uniform weighting for $\lambda \to 0$ and $n \to \infty$.

The influence of weights w_i on JR divergence can be illustrated by a simple simulated example. The simulation consists of 21 Gaussian pdfs with one heaving significantly different μ as shown in Figure 4(a). The JR divergence is calculated as: $JR^{w}_{\alpha}(\mathscr{P}_1,\mathscr{P}_2,\ldots,\mathscr{P}_i)$, $i = 1\ldots21$.

The rate of change in JR divergence is conditioned with the selection of weights as shown in Figure 4b. The most notable increase is observed if uniform weighting is applied, i.e. $w_i = 1/n$ [5], while exponential weighting delays the impact.

2.4 Health index

The concept of health index is widely used in system condition monitoring and serves to describe the aggregated level of health either of a component or machine as a whole. In the case of shot blasting machines the health H is perfect when the machine is new, hence $H = 0$. With evolving abrasive processes on the blades, more and more surface material is removed, which results in increased vibrations. The Jensen-Rényi divergence is viewed as an appropriate metric that reflects the change in vibrational pattern caused by the level of wear in turbine blades. To find the relationship, lifelong experiments have been run in which machine operation was periodically interrupted by operators who performed invasive measurement of the blades volume. All the time during operation, the vibrations were regularly measured. The most important result of the experiment is the finding that between JRD and the extent of damage (equivalent to removed volume of blade material)

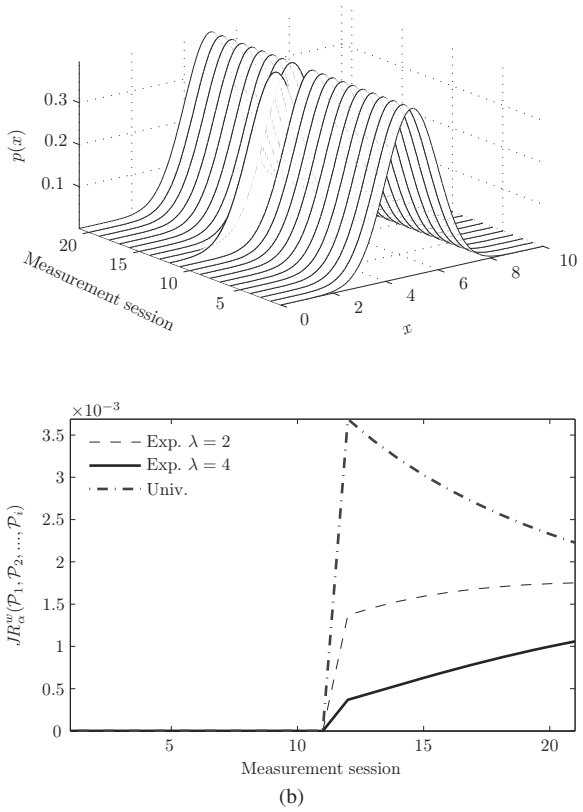


Figure 4: The evolution of JR divergence after measurement sessions. Note that all pdfs are equal except the pdf #11. (a) Pdf's of the simulated signals associated to the measurement sessions $1,\ldots,21$ (b) JR divergence. Up to $i < 11$ there is no dissimilarity in the distribution, hence $JR^{w}_{\alpha}(\mathscr{P}_1,\mathscr{P}_2,\ldots,\mathscr{P}_i) = 0$.

there exists a monotone relationship. This is indicated in Figure 5. Consequently, one can adopt the health index to be equal to the normalised JRD, i.e. $H_k = \frac{JRD_k}{JRD^*}$ where JRD^* stands for JRD when the machine turbine is considered worn out.

Note that health index H does not rise monotonically all the time, but in the period approximately $[30, 100]$ it slightly decreases. Such a behaviour looks illogical given the fact that the machine should get more and more worn with new operating cycles. The explanation lies in the fact that at the begining of the operation, the machine is not perfectly balanced. If we take into account that abrasive processes are not the same on all the blades, then asymmetry in abrasion slightly corrects the position of the center of gravity, hence resulting in lower vibrations and apparently improved condition. Such a situation changes as soon as abrasion progresses. Then asymmetrical wear in the blades results in increased imbalance and consequently raised vibrations.

3 Stochastic Model of Abrasive Wear

3.1 Abrasive wear

The key mechanism of deterioration of condition of the turbine blades is abrasive wear [6]. Each time a shot particle enters the turbine, it travels along the blade's length. Along that path it removes a small layer of the blade material of volume δV according to the Archard's law

$$\delta V = k \cdot \delta A \cdot \delta L, \tag{5}$$

where k is the wear coefficient, δA is the contact area and δL is the length of the path traversed by the shot particle on blade's surface.

In the ideal case, when all the blades were identical, the mass removed from each blade would be the same. Thus the center of gravity would stay at the rotational axis, which means negligible vibrations. However, due to irregularities in the particle size, angle of entry and variations of the blade's microstructure, there are minute variations in the mass removed from each blade. As a result, the generated vibrations tend to include amplitude modulations that depend on the number of blades and the rotational speed. Therefore, the intensity of these sidebands can be directly correlated with the removed volume of the blade material due to abrasive wear. Since there is no other source of vibrations, one can safely assume that any particular change in the vibration's signature in the lower frequency band (<2 kHz) is due to mass loss and is therefore directly related to the blades' condition.

3.2 Hidden Markov model

The Archard's law (5) describes mass loss due to the blade interaction with single shot particle. During normal operation a number of particles travel along the blade's surface. During the interval of time $[t_{k-1}, t_k]$ the loss of volume can be written as:

$$\Delta V_k = k \cdot \tilde{A}_k \cdot \tilde{L}_k, \tag{6}$$

where $\tilde{A}_k$ is the cumulative contact area of the shot particles and $\tilde{L}_k$ is the cumulative traversed distance. These two quantities are results of stochastic processes and, consequently, also ΔV_k is stochastic process. Therefore, the total volume loss at $k+1$ would be:

$$V_{k+1} = V_k + \Delta V_k. \tag{7}$$

Due to surface changes, the contact area and the traversed length are expected to change over time. Therefore, based on (7), we can assume that the volume loss ΔV_k is a process defined by the stochastic variable defined on the set of non-negative real numbers. To consistently model such a process, several options are at disposal as for example, gamma or Weibull distribution. The problem is that in such a case recursive updates can be done only by numerical techniques. A way around is to assume that the increments ΔV_k fluctuate around some mean value μ. The size of fluctuation can be described by a normally distributed white noise $w_\mu \sim \mathcal{N}(0, \sigma_\mu^2)$ such that $\sigma_\mu \ll \mu$. From here it follows that

$$\Delta V_k - \Delta V_{k-1} = w_{\mu,k} - w_{\mu,k-1}$$

and consequently one can write

$$\Delta V_k = \Delta V_{k-1} + w_{\Delta V,k}$$

where $w_{\Delta V,k} \sim \mathcal{N}(0, 2\sigma_\mu^2)$

Hence we get a state-space model with states V_k and ΔV_k. The problem now is that none of the states is available through on-line sensor reading. This can be sorted out by replacing the volume V_k by health index H_k, which is calculated on-line from acquired vibrational records.

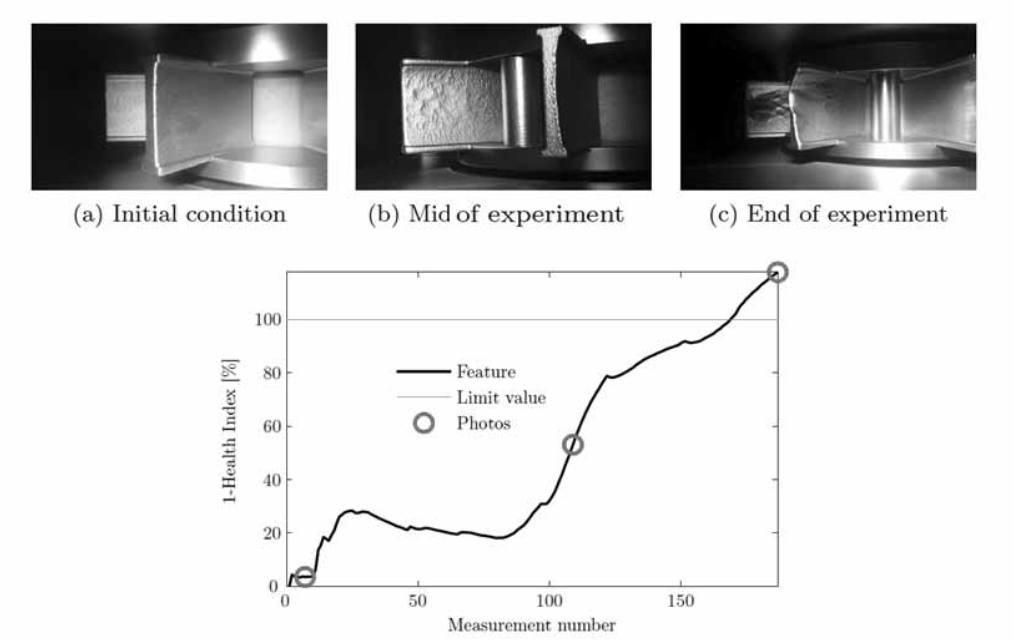


Figure 5: The relation between fault progression and evolution of health index (JR divergence): (a) at the beginning of the experiment, (b) in the middle of the experiment and (c) at the end of the experiment.

Hence the resulting state space model reads as follows

$$\underbrace{\begin{bmatrix} V_{k+1} \\ \Delta V_{k+1} \end{bmatrix}}_{\boldsymbol{x}_{k+1}} = \underbrace{\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} V_k \\ \Delta V_k \end{bmatrix}}_{\boldsymbol{x}_k} + \underbrace{\begin{bmatrix} 0 \\ w_{\Delta V,k} \end{bmatrix}}_{\boldsymbol{w}_k} \tag{8}$$

The measurement equation that relates system states and computable health index H_k reads

$$H_k = \underbrace{[1 \;\; 0]}_{\boldsymbol{C}} \begin{bmatrix} V(k) \\ \Delta V_k \end{bmatrix} + n_k \tag{9}$$

where $n_k \sim \mathcal{N}(0, \sigma_n^2)$ is white noise uncorrelated with $\boldsymbol{w}_k$.

3.3 Kalman fiter

The states of the discrete model (8) can be effectively estimated using the Kalman filter approach [7, 8]. The unknown states are updated at each measurement session resulting in the moments of the posterior distribution of system states $\boldsymbol{x}_k \sim \mathcal{N}(\hat{\boldsymbol{x}}_{k|k}, \boldsymbol{P}_{k|k})$ as follows

1. Initialisation step: set the estimates $\hat{\boldsymbol{x}}_{0|0} = \bar{\boldsymbol{x}}_0$, $\boldsymbol{P}_{0|0}, \boldsymbol{Q} = \boldsymbol{w}\boldsymbol{w}^T, \boldsymbol{R} = \sigma_n^2$ from data obtained through life-long experiments on similar machines.

2. Prediction step

$$\hat{\boldsymbol{x}}_{k|k-1} = \boldsymbol{A}\hat{\boldsymbol{x}}_{k|k-1}$$
$$\boldsymbol{P}_{k|k-1} = \boldsymbol{A}\boldsymbol{P}_{k-1|k-1}\boldsymbol{A}^T + \boldsymbol{Q}$$

3. Update step: calculate the system output vector $\boldsymbol{y}$ based on calculated JRD and then update the moments of state probability distribution function

$$\boldsymbol{K}_k = \boldsymbol{P}_{k|k-1}\boldsymbol{C}^T(\boldsymbol{C}\boldsymbol{P}_{k|k-1}\boldsymbol{C}^T + \boldsymbol{R})^{-1}$$
$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k(\boldsymbol{y}_k - \boldsymbol{C}\hat{\boldsymbol{x}}_{k|k-1})$$
$$\boldsymbol{P}_{k|k} = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{C})\boldsymbol{P}_{k|k-1}$$

4. When the next measurement session appears set $k = k+1$ and go to step 2.

3.4 RUL predictor

Having an updated model at a given measurement session k one can simulate the possible future trajectories of the state space model (8) by Monte Carlo approach. Using realisations of random processes of noise terms $w_{\Delta,k+s}$, n_{k+s}, $s > 0$ is is possible to calculate the corresponding trajectories of the state vector $\boldsymbol{x}_{k+s}$ and the predicted health index H_{k+s}. Based on that one can eas-

ily calculate the distribution of first passage time, i.e. the time s^* at which the health index H crosses the upper bound H^*.

4 Results of experiments

The RUL estimation algorithm was evaluated on a shot blasting turbine in real operating environment. The blades were subjected to 400 operational hours spread over a period of 4.5 months. Vibration signals were acquired during 10 seconds long measurement sessions every two hours while the machine was in full operation. In that period, three visual inspections were performed after 10 hours of operation, at the 120$^{\text{th}}$ hour and at the end of the experiment. Vibrations were measured on the bearing housing nearest to the turbine with sampling frequency of 10 kHz.

4.1 Evolution of the health index

The health index was calculated as JR divergence according to (1) with unifirm weights w_i. First 20 hours of operation were used as a reference point. The evolution of the health index is shown in Figure 5.

As shown in Figure 5, in the initial phase, the health index values were near zero. This is an indication that the energy distribution of the newly observed vibration is very similar to the initial 'fault-free' distribution, hence the minimal JR divergence.

The first significant increase of the JR divergence occurred around the 30$^{\text{th}}$ hour of operation. After the initial increase the JR divergence gradually decreased. As said, this effect can be attributed to the run-in phase of the turbine blades.

The onset of fault is visible at the 80$^{\text{th}}$ hour of operation. At this point the degradation of the blade condition commenced. This is clearly indicated by the increase in the calculated JR divergence. The observed degradation was confirmed by the visual inspection performed at 120$^{\text{th}}$ hour, as shown in Figure 5. The degradation trend is kept almost constant until the last fifth of the run i.e., around the 130$^{\text{th}}$ hour. The calculated health index surpassed the threshold at the 180$^{\text{th}}$ hour. The operation was halted at the 190$^{\text{th}}$ hour with the blade condition corresponding to the estimated health index, as shown in Figure 5.

4.2 RUL prediction

The evolution of the calculated health index is evaluated according to the Archard's law, as described in Section 4.1. Based on results of Kalman filtering, the trajectories of future states, and consequently health index, are calculated from a set of noise realisations.

The RUL prediction based on the first 100 measurements is shown in Figure 6. At each time moment, the Kalman filter provides estimates of the posterior probability distribution of the state vector $\boldsymbol{x}_{k+s}$ and the output y_{k+s}. To come to the distribution of the actual RUL we perform Monte Carlo simulations of the output trajectoris. The distribution of the RUL can be evaluated from the histogram of first passage times for each simulation run. As shown in Figure 6, the proposed unscented Kalman filter (UKF) provides left skewed RUL estimates. The 3σ confidence interval is sufficiently narrow and corresponds to the actual evolution of the health index.

For proper assessment of the model's accuracy, the RUL estimates should be plotted versus a theoretically expected RUL. Typically, the theoretical RUL is expected to be a linear function with gradient -1. This is shown in Figure 7. Note that during the first $2/3$ of the operational life the RUL prediction is not reliable. However, in the last third of the life, predictions become rather accurate meaning that roughly 2 months before the blades are fully worn the operators have reliable information, which could be used to plan the maintenance actions at a convenient occasion in a way that do not disturb regular production (for example, during a weekend or night shift).

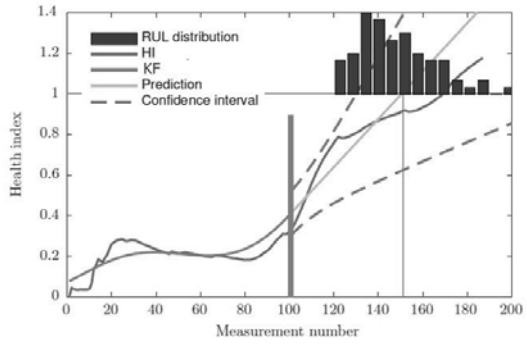


Figure 6: RUL prediction at the 100$^{\text{th}}$ measurement

5 Conclusion

The proposed feature based on JR divergence is shown to be sufficiently sensitive to perform accurate condition monitoring of shot blasting machines. Furthermore, it is shown that the evolution of the JR divergence can be directly related to the removed mass from the turbine's blades due to abrasive wear. As a result, the evolution profile can be described through Archard's law of abrasive wear. Based on this result, accurate RUL prediction is be achieved by estimating the models's states using computationally simple Kalman filter and Monte Carlo simulations over noise realisations.

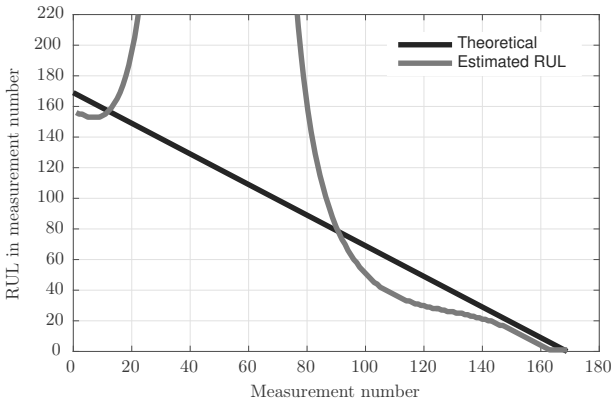


Figure 7: Performance of the RUL prediction algorithm.

Acknowledgements

The authors acknowledge the support of the Slovenian Research Agency through the Research Programme P2-0001 and the Research Project L2-7663.

References

[1] Lee J, Wu F, Zhao W, Ghaffari M, Liao L, Siegel D. Prognostics and health management design for rotary machinery systems—reviews, methodology and applications. *Mechanical Systems and Signal Processing*, vol. 42, no. 1–2, pp. 314 – 334, 2014.

[2] Jablonski A. Condition monitoring systems readressed and revised. vol. Nov., p. 2, 2015.

[3] Boškoski P, Juricic D. Fault detection of mechanical drives under variable operating conditions based on wavelet packet rényi entropy signatures. *Mechanical Systems and Signal Processing*, vol. 31, pp. 369—381, 2012.

[4] Basseville M. Divergence measures for statistical data processing - an annotated bibliography. *Sig-nal Processing*, vol. 93, pp. 621–633, 2013.

[5] Hamza A, Krim H. Image registration and segmentation by maximizing the jensen-rényi divergence. In *Energy Minimization Methods in Computer Vision and Pattern Recognition* (A. Rangarajan, M. Figueiredo, and J. Zerubia, eds.), vol. 2683 of *Lecture Notes in Computer Science*, pp. 147–163, Springer Berlin Heidelberg, 2003.

[6] Bisson EE. Adhesive, abrasive, corrosive, and surface fatigue wear modes. Tech. Rep. TM X-52426, NASA Lewis Research Center, Cleveland, OH, United States, 1968.

[7] Verhaegen M, Verdult V. *Filtering and System Identification*. Cambridge University Press, Cambridge, UK, 2007.

[8] M. Gašperin M, Juricic D. Application of unscented transformation in nonlinear system identification," *{IFAC} Proceedings Volumes*, vol. 44, no. 1, pp. 4428 – 4433, 2011. 18th {IFAC} World Congress.

AMEBA-Evolutionary Computation Method: Comparison and Toolbox Development

Marko Corn*, Maja Atanasijević-Kunc

Faculty of Electrical Engineering, University of Ljubljana, Tržaška cesta 25, 1000 Ljubljana, Slovenia;
marko.corn@ameba.si*

Simulation Notes Europe SNE 26(4), 2016, 229 – 236
DOI: 10.11128/sne.26.tn10354
Received: November 15, 2016
Accepted: December 5, 2016 (Special Issue Review)

Abstract. Evolution algorithms are optimization methods that mimic a process of the natural evolution. Their stochastic properties result in a huge advantage over other optimization methods, especially regarding solving complex optimization problems. In this paper, several types of evolutionary algorithms are tested regarding a dynamic nonlinear multivariable system modelling and control design. We have defined three problems: the first one is the so-called grey box identification problem where the characteristic of the system's valve is under investigation, the second one is a black box identification where the goal is a dynamic system's model development using system's measurements data, while the third one is a system's controller design. The efficacy of solving presented problems was compared to the usage of the following optimization methods: genetic algorithms, differential evolution, evolutionary strategies, genetic programming, and a developed approach called AMEBA algorithm. All methods have proven to be very useful for grey box identification and design of a system's controller, but AMEBA algorithm has also been successfully used in a black box identification, where it generated a corresponding dynamic mathematical model.

Introduction

In general, the evolutionary algorithms can be divided into two major groups: parametrical and structural algorithms. Parametrical algorithms evolve parameters, while structural algorithms evolve structures or mapping functions. For example, if we would have to design a controller for a dynamic system, parametric algorithm would demand to define parameters of the chosen controller structure (very frequently a PID controller is used).

In contrast to parametrical algorithms, structural algorithms do not require predefined form of the controller, as they can evolve the whole controller through their evolutionary process.

The most popular parametrical algorithms are genetic algorithms (GA) [1][2], evolutionary strategies (ES) [3], differential evolution (DE) [4] and others [5].

Most established structural algorithm is genetic programing (GP) that has multiple implementations from the three-based implementation [6] to the grammatically based implementation [7] and the evolutionary programming that is directed into the evolvement of finite state machines [8].

Evolutionary algorithms can be used also in the complex field of the design of controllers of dynamic systems, e.g. multivariable, non-linear, time-variant [9].

In this paper, the evolution of different models and control strategies are designed and compared with the usage of different evolutionary algorithms. From the parametrical group the efficacy of GA, ES and DE is illustrated, while from the structural group an algorithm of tree based genetic programming and the Agent Modelled Evolutionary Based Algorithm (AMEBA) are used [10],[11]. Relative advantages and disadvantages have been estimated regarding modelling and control design of non-linear multivariable dynamic system of three coupled thanks.

The paper is organized in the following way. In the first section a short description of the three coupled tanks system is given. In the second section a structure of the system's model and the corresponding controller are specified. In the third and fourth sections the modelling and the control design results which were generated using different evolutionary algorithms are presented and compared. The result section is followed by the description of AMEBA system toolbox that was used to generate the results of AMEBA method [12]. At the end, the conclusions and some ideas for the future work are given.

1 Three Coupled Tanks System

System of three coupled thanks is illustrated in Figure **1**. It consists of three identical cylindrical assembled water tanks with cross area S, which are interconnected with the pipes and two valves V_1 and V_2, while the valve at the output pipe is V_3. Actuators of this system are two water pumps that supply the first and the third tank with water flows $\Phi_{vh1}(t)$ and $\Phi_{vh2}(t)$. Water levels in each tank $h_1(t)$, $h_2(t)$, and $h_3(t)$ are measured with the corresponding sensors. Level difference between the first and the second tank generates water flow $\Phi_3(t)$ through the valve V_1 and level difference between the second and the third tank generates flow $\Phi_4(t)$ through the valve V_2. The output flow $\Phi_{izh}(t)$ depends only on the water level $h_3(t)$ and valve V_3 properties.

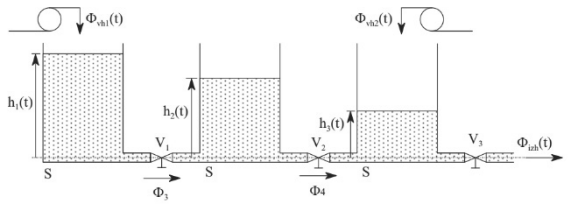


Figure 1: System of three coupled tanks.

System of three coupled thanks represents a laboratory device but for the testing we have used its model [13].

1.1 Model structure

During the phase of designing a model of certain dynamic system it is usually desired to include as much knowledge of the system as possible. In such a way, we have more chances of building a suitable model. Theoretical modelling approach enables model building on the basis of the equilibrium equations which determine system's basic behaviour. For further model improvement, additional nonlinear functions are needed which describe different specific parts of the system. In the first phase the system's model can be presented with three equilibrium equations which are described with equations (1).

$$\begin{aligned} \Phi_{vh\,1}(t) - \Phi_3(t) &= S \cdot \dot{h}_1(t) \\ \Phi_3(t) - \Phi_4(t) &= S \cdot \dot{h}_2(t) \\ \Phi_4(t) + \Phi_{vh\,2}(t) - \Phi_{izh}(t) &= S \cdot \dot{h}_3(t) \end{aligned} \tag{1}$$

Input flow rates are determined by the water pumps which are controlled with the voltage signals u_1 and u_2. Water flows from the first to the second tank and from the second to the third tank are given with the equations (2).

$$\begin{aligned} \Phi_3(t) &= k_1\sqrt{h_1(t) - h_2(t)} \\ \Phi_4(t) &= k_2\sqrt{h_2(t) - h_3(t)} \end{aligned} \tag{2}$$

These water flows depend on the water levels in the tanks and the characteristics of the valves. These characteristics are expected to be of the square root type. From the experimental data it was established that static characteristic of the valve V_3 is not square root function and so we have tried to estimate corresponding description by the so-called indirect identification method or 'grey box identification' [14]. Grey box identification is a process in which we firstly gather measurements of the system's behaviour, secondly we build a mathematical model and include all the data that we have into it. Thirdly we try to estimate the missing parameters or functions to the constructed model. Block diagram of the chosen structure is illustrated in Figure 2.

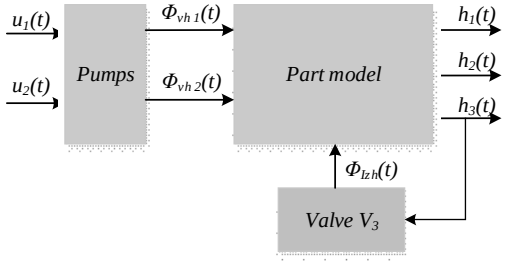


Figure 2: Block diagram of the three coupled tanks system structure.

Estimation of the characteristic of the valve V_3 is defined optimization problem as the rest of the model had been constructed using the equilibrium equations and measured characteristics of the other parts. Optimization process was minimizing the difference between responses of the model and measurements of the system by adapting valve's characteristic. The fitness function used in this optimization process is presented by equation (3).

$$\mathrm{J} = \sum_{i=1}^{3} \int |h_i(t) - h_i^*(t)| dt \tag{3}$$

Fitness function is equal to the absolute sum of difference between responses of the model and corresponding measurements.

Measurements obtained for the identification process consist of eight responses to the different input or excitation signals. Six of them were used in the identification process and two for the validation of the model. One pair of the excitation signals and corresponding responses is illustrated in Figure 3 and Figure 4.

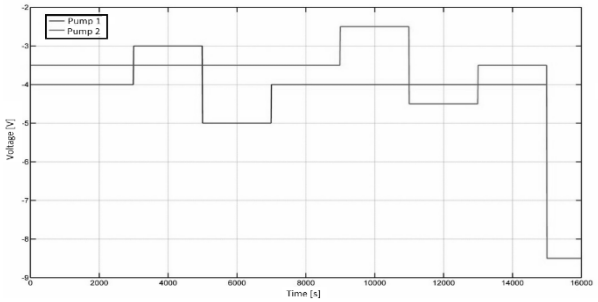


Figure 3: Input signals $u_1(t)$ and $u_2(t)$.

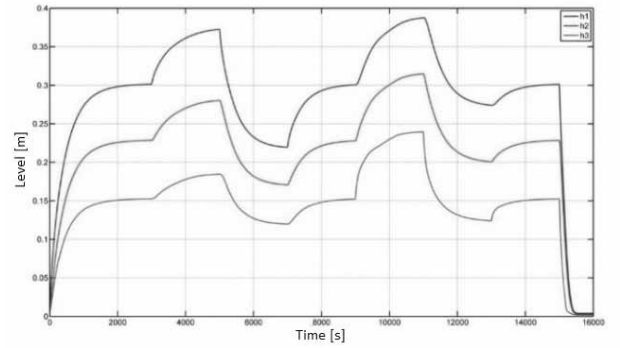


Figure 4: Responses of the system to chosen input signals.

From the presented responses, the cross couplings are visible (each input influences both systems' outputs $h_1(t)$ and $h_2(t)$). These cross couplings also prove that the system is a multivariable one.

1.2 Controller design

Block diagram of system's close loop operation is presented in Figure 5.

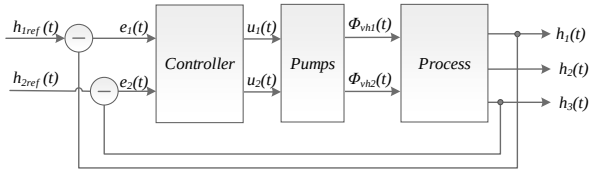


Figure 5: Closed-loop system operation

Close loop system should maintain water levels in the first and in the third tank at the corresponding reference values h_{ref1} and h_{ref2}. Fitness function that is used in the optimization process of the controller design is presented in equation (4).

$$J = w_{opt} \int |e_1(t)| + |e_2(t)| dt + (1 - w_{opt}) \int |u_1(t)| + |u_2(t)| dt \quad (4)$$

Fitness function represents a sum of the integrals of errors e_1 and e_2 (that represents difference between actual water levels h_1 and h_3 and referenced values h_{ref1} and h_{ref2}) and integrals of the pumps activity u_1 and u_2. Both contributions are weighted with the weight w_{opt}. The control system was tested with the usage of the reference signals that are presented in Figure 6.

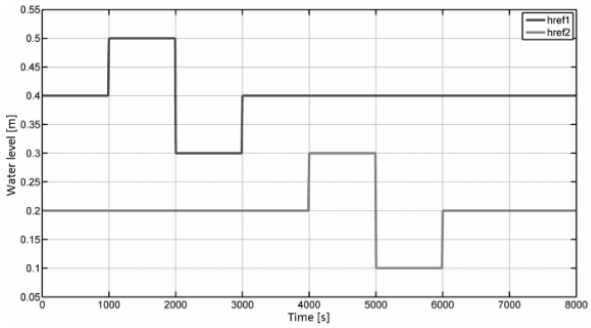


Figure 6: Reference signals.

Controller must be able to control the systems water levels in a way that is demanded by the step shaped changes of the reference signals.

2 Modelling Results

Modelling results are divided into two groups. The first group consists of the results obtained by the parametrical evolutionary algorithms and the second group by the structural evolutionary algorithms.

2.1 Parametrical evolutionary algorithms

Parametrical evolutionary algorithms can optimize only parameters, so we have constructed a polynomial mathematical function with four parameters a_1, a_2, a_3, and a_4 which should describe as good as possible the relation between the water level $h_3(t)$ and output water flow $\Phi_{izh}(t)$.

$$\Phi_{izh}(t) = a_1 h_3^3(t) + a_2 h_3^2(t) + a_3 h_3(t) + a_4 \quad (5)$$

We have tested and compared three parametrical methods GA, ES, DE. Optimization process was defined for all methods identically in order to get comparable results. Solutions have been evolved during 1000 generations and with the generation size of 30 individuals. Results are presented in two ways. The first way is the comparison of the quality of the model that was generated by each method and the second is the comparison of the convergence of the used methods. Quality of generated solutions is presented in Table 1.

Met.	Error identification [%]	Error validation [%]
DE	1.77	3.27
ES	1.79	3.58
GA	1.88	4.57

Table 1: Evaluation of modelling results of parametrical algorithms.

Error column represents a relative average deviation from the identification signals of the system and validation column represents relative average deviation from the validation signals. All results are quite similar, which means that there is high probability that we have found a global minimum of the proposed valve function. Best algorithms are DE and ES that have managed to generate 1% better result. Example of the system's responses of the best model generated by the DE method is presented in Figure 7.

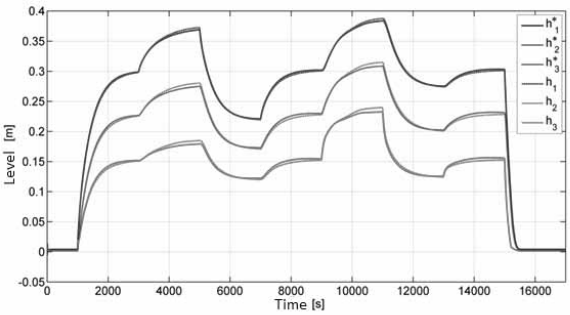


Figure 7: Comparison of measurements with the response of the model generated by the DE method.

We have compared also the convergence of the algorithms and the results which represent the average convergence of 10 optimization runs for each method are presented in Figure 8.

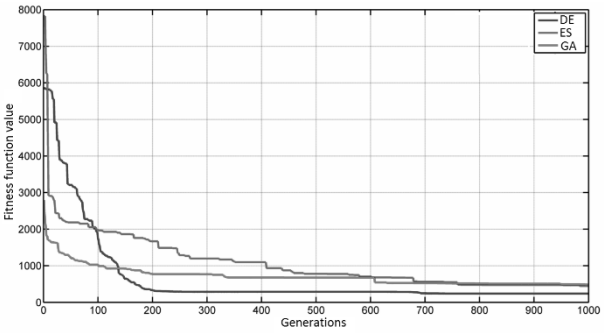


Figure 8: Average convergence of parametrical methods

Statistical analysis of the methods' convergences shows efficiency of each algorithm during the search of optimal solution. DE has the fastest convergence and it generates the best results.

2.2 Structural evolutionary algorithms

In addition to parametrical optimization also two structural algorithms, namely GP method based on trees and AMEBA were tested. For the AMEBA algorithm additional test has been conducted. Test, where the model of the whole system has been built (not just model of the valve V_3) with the black box identification method as the AMEBA algorithm can be used also for multi-input multi-output systems.

Structural algorithms are capable of building system's structure automatically. Settings of the evolution were the same for both methods which enable the comparison of the results. For the GP, we have used addition, subtraction, multiplication, division, power and constant types of nodes and for the AMEBA algorithm we have used the same nodes' types as for the GP with the use of additional dynamic nodes like delay, integral, derivative, low pass filter and high pass filter. Results are evaluated in Table 2.

Algorithm	Error ident. [%]	Error valid. [%]
GP	1.62	3.12
AMEBA valve	3.57	4.65
AMEBA full model	5.63	7.23

Table 2: Evaluation of modelling results when using structural algorithms.

GP algorithm has generated the best solution and its tree representation is presented in Figure 9.

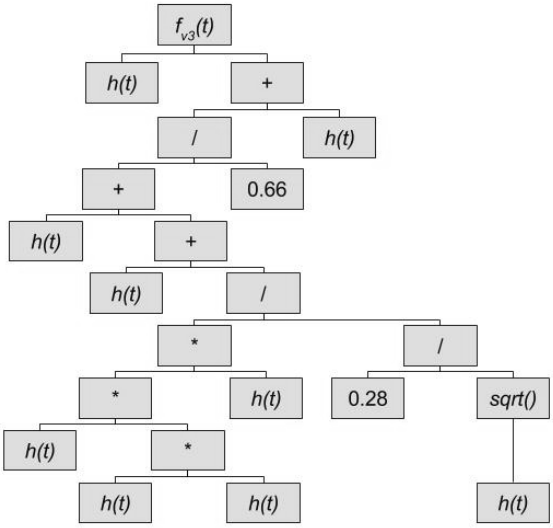


Figure 9: Solution generated with the GP method.

Simplified solution of GP is presented in equation (6). This is a polynomial function with two parts, the first has rational number in the exponent and the other is a linear one.

$$f_{V3}(t) = 2.086\, h(t)^{\frac{9}{2}} + 5.023\, h(t) \qquad (6)$$

Result generated by the AMEBA algorithm is not as good as the result obtained by GP and it is presented in Figure 10.

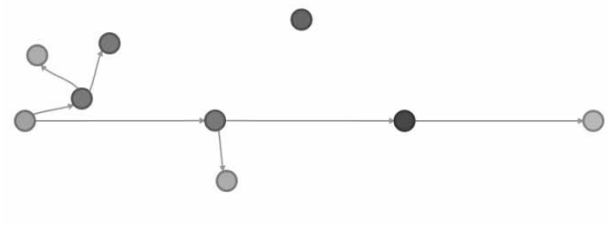


Figure 10: Graph representation of model of the valve generated with AMEBA algorithm.

In Table 3 a legend is presented that shows colours of different types of nodes assembling AMEBA algorithm solutions.

Color	Node	Color	Node
	Input		Amplification
	Output		Exponent
	Low pass filter		Delay
	High pass filter		Derivative
	Multiply		Integral
	Divide		Add

Table 3: Color-legend of different types of nodes.

Valve function that was generated by the AMEBA algorithm is presented in equation (7).

$$\Phi_{izh}(t) = -0.5 \cdot 0.54(-0.8(h_3(t)))^{0.68} \qquad (7)$$

The result of the valve function generated with the AMEBA algorithm is a nonlinear function. AMEBA algorithm has successfully generated also a model of the whole system with the process of black box identification. We have used the same measurements for generating this model that were in use for the identification of the valve. Model is represented in Figure 11. Model generated with AMEBA algorithm is complex, full of nodes of all types and feedback loops that represent system dynamic properties.

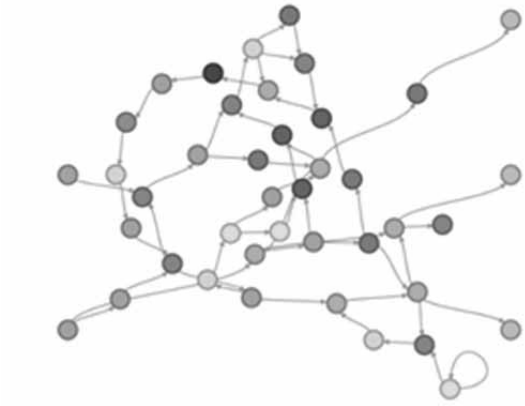


Figure 11: Graph representation of system's model generated with the use of AMEBA algorithm.

3 Results of the Controller Design

Results of designing control algorithm are also divided into two groups: into a parametrical and a structural group.

3.1 Parametrical evolutionary algorithms

Parametric methods usage demands a parametrically defined problem so we constructed a controller that is assembled with four proportional-integral (PI) controllers with 8 parameters to be optimized.

The proposed controller is a multivariable one with two inputs (differences between desired and actual water levels) and two outputs to drive water pumps. Controller's parameters to be optimized are described with equations (8).

$$\vec{u}(t) = \boldsymbol{K_p}\vec{e}(t) + \boldsymbol{K_i}\int \vec{e}(t)dt$$
$$\begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} = \begin{bmatrix} K_{p11} & K_{p12} \\ K_{p21} & K_{p22} \end{bmatrix}\begin{bmatrix} e_1(t) \\ e_2(t) \end{bmatrix} + \begin{bmatrix} K_{i11} & K_{i12} \\ K_{i21} & K_{i22} \end{bmatrix}\int \begin{bmatrix} e_1(t) \\ e_2(t) \end{bmatrix} dt \quad (8)$$
$$\begin{bmatrix} e_1(t) \\ e_2(t) \end{bmatrix} = \begin{bmatrix} h_{1\,ref}(t) - h_1(t) \\ h_{3\,ref}(t) - h_3(t) \end{bmatrix}$$

All 8 parameters are represented in two matrices $\boldsymbol{K_p}$ and $\boldsymbol{K_i}$. Results calculated with the parametrical methods are presented in Table 4.

Algorithm	Error	Energy used
DE	2.04 %	35.9%
GA	2.04 %	36.5%
ES	2.48 %	35.3%

Table 4: Evaluation of controller optimization results calculated with parametrical methods.

Results of all algorithms are very similar but the DE method has ones again proven to be the best as it calculated the controller with the lowest error and minimum estimated usage of energy.

3.2 Structural evolutionary algorithm

Structural evolutionary algorithms don't need the controller's structure to be defined in advance in contrast to parametrical methods. This group is capable to evolve the structure as well as all the parameters automatically. Results of two methods, GP an AMEBA, are presented in Table 5.

Algorithm	Error	Energy used
AMEBA	1.5 %	34.1 %
GP	9.3 %	35.5 %

Table 5: Results of controllers generated by structural evolutionary methods.

The solution which was generated by the GP method is presented by equation (9).

$$u_1(t) = e_1(t)^{2.2\,e_2(t)-1} + e_1(t) \qquad u_2(t) = u_1(t) \quad (9)$$

GP method didn't generate a suitable solution as the controller is not capable to follow corresponding reference signals. The solution generated by the AMEBA algorithm is presented in Figure 12.

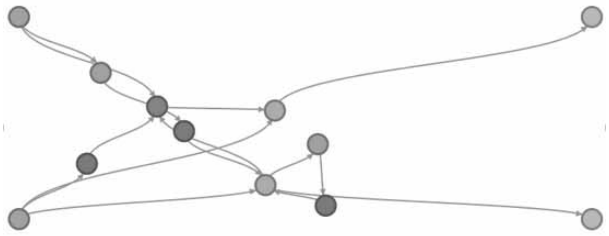


Figure 12: Graph representation of controller generated by the AMEBA algorithm.

Controller that was generated by AMEBA algorithm is illustrated by equation (10). AMEBA algorithm generated a controller with the best performance.

$$\begin{aligned} u_1(k) &= X_2(k) \\ u_2(k) &= 0,74(+e_1 + 0,11(X_0(k) * X_2(k) * 0,95(-0,87)(e_2))) \\ X_0(k) &= e_2 \\ X_1(k) &= 0,34fltLP(X_0(k)) \\ X_2(k) &= -0,2(-e_1 + 0,40fltLP(X_2(k-1)) + 0,08X_1(k-1)) \end{aligned} \quad (10)$$

4 Toolbox development

AMEBA algorithm is being developed also as a software package with user friendly graphical interface. The core development is being built in Java programming environment that can be used also with Matlab, which allows a very efficient support in simulation of dynamic systems via Simulink. Graphical interface is also developed in Matlab due to its good graphical support.

Toolbox enables settings of the simulation environment with the inclusion of Simulink model as it is shown in Figure 13.

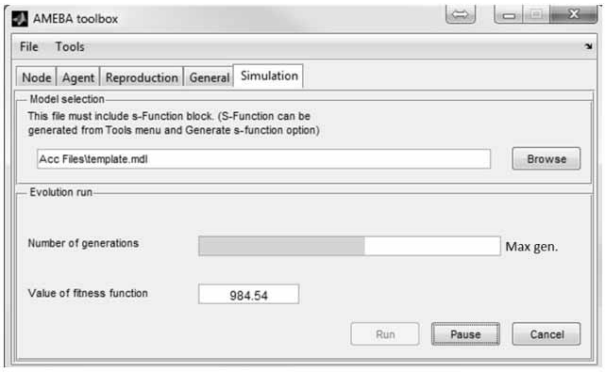


Figure 13: Settings of simulation environment.

The agent of AMEBA algorithm is implemented as S-function so it can be included into the model as a standard block. Toolbox enables control and monitoring of the optimization process where it displays current generation number and the value of the fitness function of the best agent.

Toolbox enables settings of population properties like size of population, size of reproductive population that determines how many best agents will be given opportunity to reproduce, number of elite agents, and other settings that determine the end of optimization process like maximum number of generations and minimum change in fitness function value (Figure 14).

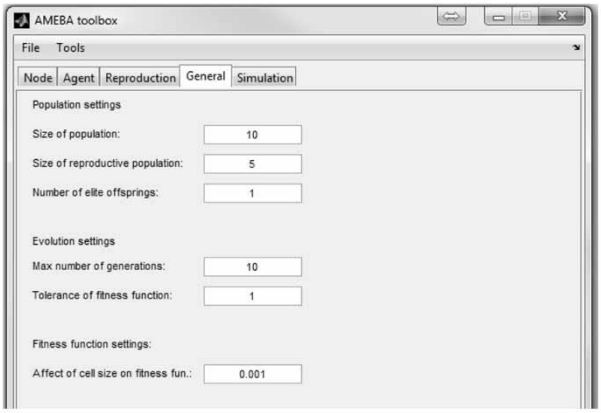


Figure 14: General setting.

The number of inputs and outputs of an agent can be defined together with the maximum number of nodes that can be generated at the agent's creation (Figure 15).

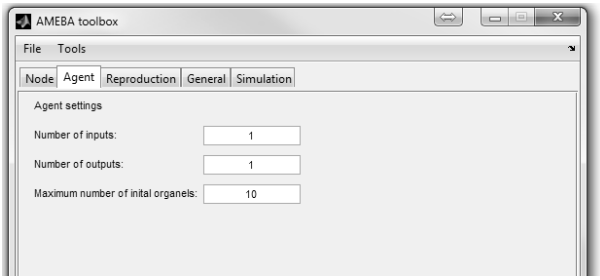


Figure 15: Agent settings.

Different types of nodes can be selected from which the algorithm will chose and build agents. Each node has its own settings that determine initial value of the nodes parameter and steepness of change in case node mutates (Figure 16).

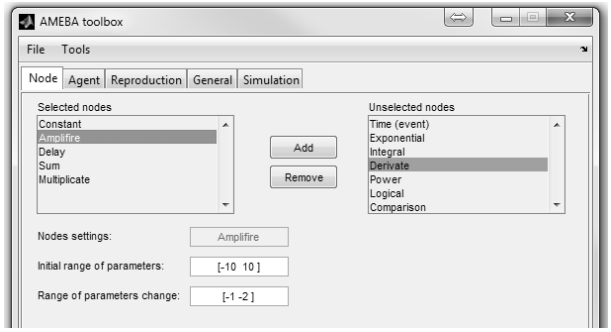


Figure 16: Node settings.

Reproduction mechanisms can be set with their parameter of probability. As agents are evaluated and selected for reproduction the reproduction mechanism is randomly selected and the probability parameter determines their possibility of being selected (Figure 17).

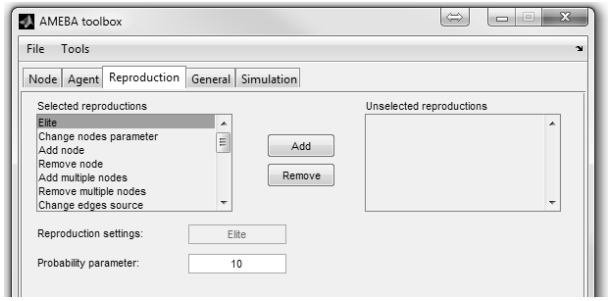


Figure 17: Reproduction settings.

Additional functionalities enable better usability of the method such as saving and importing of all setting into file for later use. With this option, also the initial population can be imported which enables the inclusion of certain knowledge of the solution into the optimization problem. It is also possible to convert agent into mathematical equation to observe its structure. It can also generate Matlab S-function file for the easier implementation in Simulink (Figure 18).

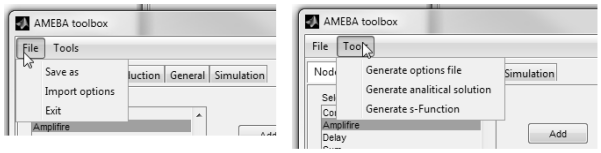


Figure 18: Additions functionalities of Toolbox.

5 Conclusions

The system of three coupled thanks was selected to present the efficacy of three different approaches of the usage of the evolutionary algorithms methods: the grey box identification, the black box identification and the controller design.

Parametrical evolutionary algorithms generated good results for both modelling and control of the system. Also, structural methods manage to generate good solutions for both types of problems. In general, the most important advantage of the structural algorithms in comparison to the parametrical methods is the absence of the need to define a suitable structure. This property is especially important when dealing with more complex systems with multiple inputs and outputs. With the usage of AMEBA algorithm, we have managed to generate also a complete model of the system and we generated a system controller with the best performance.

Future work on AMEBA algorithm development will be focused on optimization process as we are going to explore the impact of various effects on the quality of the solution and on the convergence rate of optimization process like the effect of size of the population size, suppression of the agents with large number of nodes, using multiple environments at once and similar, of course in comparison with other optimization approaches. Special attention will be devoted to the so called smart optimization where additional knowledge from chosen area can be taken into account to improve searching efficacy. The AMEBA method is a work in progress and the method will be available as an open source project.

References

[1] Atanasijević-Kunc M, Belič A, Karba R. *Optimal multivariable control design using genetic algorithms*. Vienna University of Technology; 5th Vienna Symposium on Mathematical Modeling. 2006.

[2] Goldberg D. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley; 1st1989.

[3] Beyer HG. *The Theory of Evolution Strategies (Natural Computing Series)*. Springer; 2010. 400 p.

[4] Storn R, Price K. *Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces*. Journal of Global Optimization. 1997.

[5] Brownlee J. *Clever Algorithms: Nature-Inspired Programming Recipes*. Swinburne University in Melbourne, Australia; Search. 2011.

[6] Koza JR. *Genetic Programming On the Programming of Computers by Means of Natural Selection*. MIT Press; 6th1992.

[7] Whigham PA. *Grammatically-based Genetic Programming*. Departent of Computer Science, University College, University of New South Vales; Workshop on Genetic Programming: From Theory to Real-World Applications. 1992.

[8] Fogel LJ, Owens AJ, Walsh MJ. *Artificial Intelligence through Simulated Evolution*. John Wiley; 1966.

[9] Tomažič S, Logar V, Kristl Ž, Krainer A, Škrjanc I, Košir M. *Indoor-environment simulator for control design purposes*. Building and Environment. 2013.

[10] Corn M, Atanasijević-Kunc M. *Cell based Genetic Programming Toolbox (CGP - Toolbox)*. 20th ERK Conference. 2011.

[11] Corn M, Černe G. *A Graph-Based Evolutionary Algorithm : Cell Based Genetic Programming*. Proceedings of the 5th International Conference on Bioinspired Optimization Methods and their Applications, BIOMA 2012. 2012.

[12] Corn M. *AMEBA – Structural evolutionary optimization: method and toolbox development*. 27th European Modeling and Simulation Symposium. 2015.

[13] Atanasijević-Kunc M. *Multivariabilni sistemi, Zbirke kompleksnejših problemov*. Založba FE in FRI; 42005. 265 p.

[14] Tan KC, Li Y. *Grey-box model identification via evolutionary computing*. Control Engineering Practice. 2002.

Modeling and Simulation of GMA Welding Process and Welding Power Sources

Marjan Golob

Faculty of Electrical Engineering and Computer Science, University of Maribor,
Smetanova ulica 17, 2000 Maribor, Slovenija; *marjan.golob@um.si*

Simulation Notes Europe SNE 26(4), 2016, 237 - 244
DOI: 10.11128/sne.26.tn10355
Received: November 10, 2016
Accepted: December 5, 2016 (Special Issue Review)

Abstract. Simulation techniques are useful tools for study and research of new welding technologies, and for the rapid development of new control algorithms and control units such as power source circuits, and welding current or voltage controllers. The objective in this research is to combine the simulation of Gas Metal Arc Welding (GMAW) process models with the simulation models of inverter based power machines. The GMAW process is considered as an electrical circuit and the mathematical model is based on physical descriptions of several parts of GMAW process, as are the electric circuit of power supply, the arc dynamics, and the electrode melting process. To establish the validity of the proposed GMAW model, a simple welding application was simulated and welding parameters were derived from experimental conditions. Next, the simulation model of full-bridge DC-DC converter is presented and the discrete PI controller for welding current feedback control is proposed. Both models, the GMAW model and the inverter power supply model, are combined and simulated together. Finaly, the simulation study of firing the thyristors, which enables steady and pulsed direct current welding with a single fully controlled bridge converter is shown.

Introduction

Conventional approaches to automation of welding have been reasonably successful, but there are still significant opportunities for additional development. Successful implementation of multivariable weld process control involves sensing, modelling, and control.

Process modelling provides a means of incorporating principal and empirical information into a control strategy. Models may be used off-line to evaluate and tune a controller in a simulation. They may also be used to develop transfer functions of a process for use in formal controller design, or to provide maps between input and output parameters. Process models are important bridge between what is known and what is desired.

Several research studies, for example [1]-[4], categorize the GMAW process as an electrical circuit. A mathematical model of the GMAW process is normally developed first. A description of the electric arc is then presented, and all equations are combined into a general model that describes the GMAW process. Simulation methods are used to illustrate the behaviour of the GMAW process. During these simulations the welding power source's dynamic behaviour is often simplified. For most GMAW applications the desired welding conditions are such that time constant of the self-regulating process is shorter than the oscillation rate [2]. With the aim of maintaining a high quality of welding results, the output welding current and voltage must be controlled during the welding process. Furthermore, a real time control system is an important element of modern GMAW welding machine [5], [6].

Modern GMAW equipment is the combination of a sophisticated power electronic device and high performance microprocessor-based control systems. The development process of inverter-based welding power source with the corresponding control system is a complex and expensive process, that requires extensive human and material resources [7]. When using simulation the quality of design process can be improved and the design cost can be significantly reduced.

Therefore, with the aim of improving the design process, an attempt was made to combine the simulation model of GMAW process with the simulation models of welding power sources [8], [9]. The simulation results are very useful for the rapid development of new control algorithms and the designing of new power sources.

1 GMAW Process Dynamic Model

A fundamental significance of GMAW process is that it incorporates automatic feeding of a consumable electrode that is protected by an externally supplied shielding gas as is presented in Figure 1. A constant voltage power supply is fed to the electrode and the workpiece. To get the the desired weld quality, wire feed speed v_e, torch travel speed v, open circuit voltage u, and contact tip to workpiece distance H can be adjusted. Here, h is the distance of the center of mass of the droplet above the workpiece. The mathematical model development of a GMAW process is performed by taking into consideration the model of the electrical circuit, the welding wire melting rate, the model of the dynamics of the pendent drop, and the phenomena of the drop transfer mode.

The sum of the voltages around a GMAW circuit, as presented in Figure 1, is

$$u = R \cdot i + L\frac{di}{dt} + R_l \cdot i + u_{arc} \tag{1}$$

where u is open circuit voltage of the power source, R is resistance of the power source, L is inductance of the power source, i is welding current, R_l is electrode stick-out resistance, and u_{arc} is arc voltage. Electrode stick-out resistance is dependent on resistivity of the electrode stick ρ, cross-sectional area of the electrode wire A, and electrode stick-out length l. It is assumed that u, R, L, ρ, and A are constant parameters and i, l, u_{arc} are dependent variables.

The electrode resistance R_l depends on total the length of the electrode stick-out length and drop length $l=l_s+l_d$. The dynamics of l_s depend on the feeding speed of electrode v_e, the melting speed v_m, and on the vertical velocity of the contact tip v_c. The contact tip to workpiece distance (CTWD) is indicated by H, and by ignoring the length of the drop (l_d = 0 and l = l_s) the length of the arc h is:

$$h = H - l \tag{2}$$

and, the dynamic of electrode stick-out is given by:

$$\frac{di}{dt} = v_e - v_m + v_c \tag{3}$$

With respect to (2) and (3), the arc length speed is:

$$\frac{dh}{dt} = v_m - v_e - v_c \tag{4}$$

the dynamics of the melting speed v_m and arc voltage U_{arc} need to be described in greater detail.

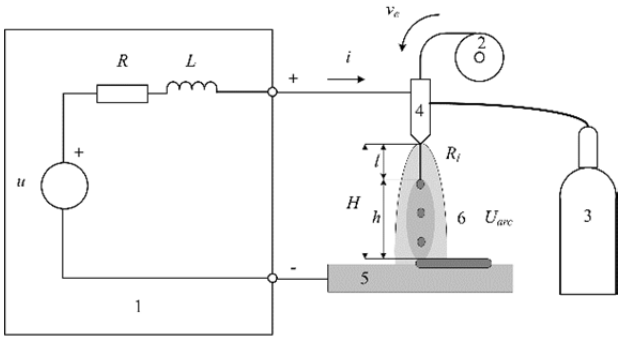


Figure 1: Schematic diagram of GMAW process and electrical circuit of the self-regulating arc process. (1) Power source; (2) wire feed unit; (3) shielding gas; (4) welding gun; (5) workpiece; (6) welding arc and material transfer process.

The total arc voltage u_{arc}, is made up of three separate parts: the anode and cathode drop voltage u_{a+c}, the drop voltage in the arc column, which is a function of the electric field strength E and the arc length h, and the drop voltage, that depends on current i and arc resistance R_{arc}. In our model we suppose that u_{a+c} is constant. Considering this, the simplest model of the electrical arc is a voltage equation:

$$u_{arc} = u_{a+c} + E \cdot h + i \cdot R_{arc} \tag{5}$$

When the current flows through the electrode and the arc the electrode is heated by the current flowing through it. This heat depends on the resistance of the welding wire Several studies have described the physical background of a welding wire melting phenomenon. In [9], the research results from a study of anode and cathode melting rates are presented and in [10], the characteristic of melting rate as a function of current, type of gas, and other parameters is reported. In these and other related works [11], [12] the expression for the total melting velocity v_m is proposed as:

$$v_m = k_1 \cdot i + k_2 \cdot i^2 \cdot l \tag{6}$$

where K_1 and K_2 are empirical constants for given wire materials and sizes. An equivalent state-space representation is presented below.

$$\begin{aligned}\dot{x}_1 &= \frac{1}{L}\cdot\left(u_1 - \frac{\rho}{A}\cdot x_1\cdot(x_3 - x_2)\right) - u_{a+c} - E\cdot x_2 \\ &\quad - (R_{arc} + R)\cdot x_1 \\ \dot{x}_2 &= k_1\cdot x_1 + k_2\cdot x_1^2\cdot(x_3 - x_2) - u_2 - u_3 \\ \dot{x}_3 &= u_3 \\ y &= x_1\end{aligned} \tag{7}$$

where the states are: $x_1 = i$ is the welding current, $x_2 = h$ is the length of the arc, $x_3 = H$ is the CTWD, and imputs are $u_1 = u$ is open circuit voltage of the power source, $u_2 = v_e$ is the feeding speed of electrode, and $u_3 = v_c$ is the vertical velocity of the contact tip.

Further parts of GMAW process dynamics, for example the welding drop dynamics, or drop detachment process are also important, but in this model are neglected. On the other hand, in GMAW not only the spray transfer conditions are widely employed, but also the short-circuiting arc's conditions with a relatively small current. This type of material transfer is within the mainstream of high-speed welding regarding thin sheet or overhead position welding of line pipes. Short-circuiting welding is a complicated process in which short-circuiting and arc generations are repeated intermittently. Transfer of molten droplet of mass m involves many complex parameters (more while using CO2, and other shielding gases). Many researchers have worked on the modeling of the GMAW process. A fifth order nonlinear model of GMAW process has been used by most of the researchers for the control process [1], [3]-[6].

2 Welding Power Sources

A precise control of the arc welding process with its complexity of heat input, material transfer and arc behaviour could be achieved when we use a modern transistor controlled inverter-based power sources. To achieve controlled droplet transfer it is necessary to switch the current level from about 15A to 500A within less than 200 μs for a pulse time of 1 ms and a pulse frequency of 200 Hz. Today the asymmetrical half-bridge-forward-converter became the favourite topology of an inverter power source. IGBTs or MOSFETs are mostly used as power switches dependent on supply voltage and output power range.

2.1 Inverter based power sources

The inverter-based welding power supply consists of a rectifier, an inverter switch circuit, a high-frequency ferrite transformer, high-frequency rectifier, and an inductor as is presented in Figure 2.

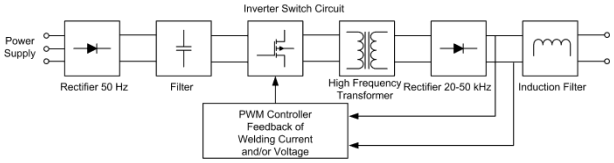


Figure 2: Power supply architecture of modern-inverter based welding machine.

The switch circuits are controlled by microprocessor-based PWM controller units. The schematic of a simulation model of a full-bridge DC-DC converter is shown in Figure 3.

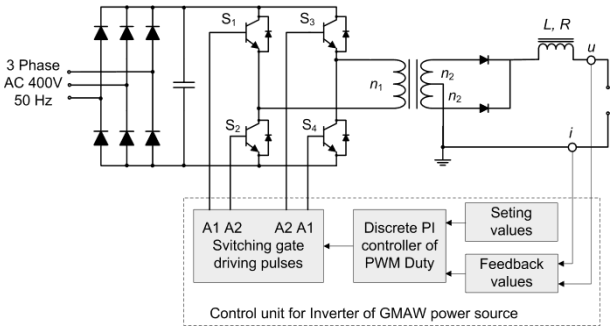


Figure 3: Power supply architecture of modern-inverter based welding machine.

The conventional DC-DC converter operates using a Pulse Width Modulation (PWM) current controller. The DC-DC converter operates at constant switching frequency, which is usually limited to 20 - 50 kHz. The amplitude of the welding current depends on the change of the phase shift between transistors S1, S2, S3, and S4. The PWM signals are generated using a simple circuit and are used for driving four transistors by changing the duty cycle. The duty cycles are usually controlled using feedback controller (voltage, current, or both). In [13] and [14], the implementation studies of proportional integral derivative (PID) are presented.

2.2 Thyristor based power sources

The synergic pulsed MIG/MAG welding with width-controlled sine-wave current pulses is mostly realised with thyristor-based power sources.

Such current-pulse waveform could be obtained if a power source is controlled by thyristors integrated, when the pulse frequency is constant and the current-pulse power is controlled by the delay time of the thyristor ignition [15], [16], [17].

The thyristor-based welding power supply consists of a rectifier, an inverter switch circuit, a high-frequency ferrite transformer, high-frequency rectifier, and an inductor as is presented in Figure 4.

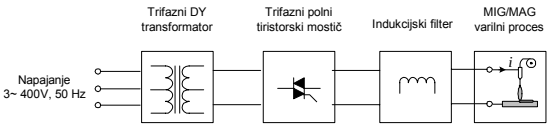


Figure 4: Power supply architecture of modern-inverter based welding machine.

The thyristor-based power sorces produce harmonics in the AC power supply network. In the case of pulsed direct current welding, it is possible, with a proper method of firing the thyristors, to generate both steady direct as well as pulsed direct current with a single fully controlled bridge controller. Using an active filter during pulse welding we could reduce harmonics effectively.

3 Simulation Examples and Results

3.1 Simulation of GMAW Proces

An automatic welding application was assumed. The parameters derived from experimental conditions are shown in Table 1.

A constant welding speed was assumed. The welding torch was positioned 16 mm (H) from the work distance. The selected welding wire feed rate v_e = 50 mm/s and the open circuit voltage u = 24 V were set.

The first simulation was performed to find the welding current response when the CTWD was changed from 16 mm to 12 mm (at time 2.5 s) and back (at time 7.5 s). In addition, the electrode feeding speed v_e was changed from 50 cm/min to 75 cm/min at time t = 5 s.

Figure 5 shows the changes in the welding voltage and current time responses, and the changes of the arc length.

Parameter	Descr. of the parameter	Value
R	Power source resistance	0.07 Ω
L	Power source inductance	0.02 mH
ρ	Specific electrical resistance of the electrode	0.1 Ω/m
A	Cross-sectional area of the electrode wire	$1.02 \cdot 10^{-6}$ m^2
E	Electric field strength	675 V/m
u_{a+c}	Arc voltage constant	11.55 V
R_{arc}	Arc resistance	0.03 Ω
v_e	Feeding speed of electrode	0.5 m/min
k_1	Empirical constant	0.626 m/(As)
k_2	Empirical constant	$7.55 \cdot 10^{-5}$ (A^{2}s)$^{-1}$
H	Contact tip to workpiece distance (CTWD)	0.16 m

Table 1: GMAW process simulation parameters.

The welding current rose and fell with the changes of H and v_e, as expected. It can be seen from the first plot in Figure 2 that the arc length h (dotted curve) decreased after the H changed from 16 to 12 mm and then increased back to the previous length. Accordingly, the electrode length l changed from 11.25 mm to 7.5 mm, which meant that the electrode melted at a higher speed when the current increased. In the second plot of Figure 2 the electrode feeding speed was increased from 50 mm/s do 70 mm/s. This led a reduction of the arc resistance and an increasing of welding current.

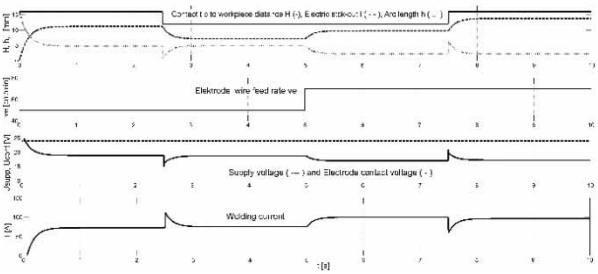


Figure 5: Simulated results of contact to workpiece voltage waveform (third plot) and welding current waveform (fourth plot). Simulation response of the GMAW model when the CTWD was changed from 16 mm to 12 mm (first plot) and the electrode feeding speed ve was changed from 0.5 m/min to 0.7 m/min (second plot).

3.2 Simulation of Dynamic Behaviour of a Full-Bridge DC-DC converter

Simulation of welding source was performed in the program Matlab / Simulink using blocks of the SimPowerSimulation library. A full-bridge circuit is simulated as the topology of the main inverter circuit. The load of the inverter depended on GMAW simulation model and was continuously changing. In Table 2 the design specification of the DC-DC converter and the circuit parameters are described, respectively.

Parameter	Descr. of the parameter	Value
f_s	Switching frequency	40 kHz
C	Capacitance	1 μF
Pn	Transformer nominal power	5 kW
$n_1 : n_2 : n_2$	Transformer turns ratio	3.5 : 1 : 1
S_1 - S_4	Ideal switch, IGBT	
R_{on}	Switch internal resistance	140 mΩ
R_s	Snubber resistance	1MΩ
C_{sd}	Snubber capacitance	4.7 nF
T_s	Control sample time	0.1 ms
T_I	PI controller Integral constant	2 ms
K_P	PI controller proportional gain	0.2 %/A

Table 2: DC-DC converter and other circuit parameters.

The simulation results from the welding using current control feedback and the PWM full-bridge DC-DC converter are shown in Figure 6.

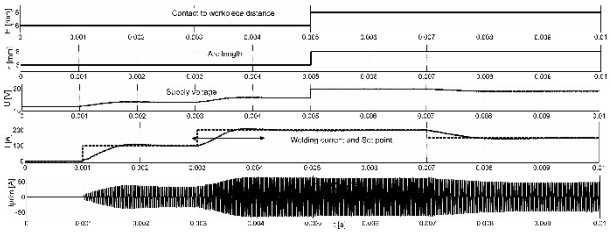


Figure 6: Simulation results of the welding process with current control feedback and PWM full-bridge DC-DC converter-based welding power source. The upper plot shows the change of the CTWD from 16 mm to 18 mm. The forth plot presents the welding current transient response, and on the fifth plot the time response of the primary current is shown.

Constant welding speed was supposed. The welding torch was positioned at 16 mm (H) from work distance (CTWD) and after 5 ms H was increased to 18 mm, as is presented in the first graph of Figure 5. The welding wire feed-rate v_e was set at 70 mm/s.

After 1 ms the welding current's set point was increased to 100 A, after 3 ms to 200 A, and finally after 7 ms to 150 A. The fourth plot in Figure 5 presents the current control system transient response, which was stable with a small overshoot and was sufficiently fast. On the fifth plot, the time response of the primary current is shown.

For a better presentation of the generated PWM signals the same simulation results were plotted within a time window from about 3 to 4 ms, and marked with an arrow in the fourth graph in Figure 7.

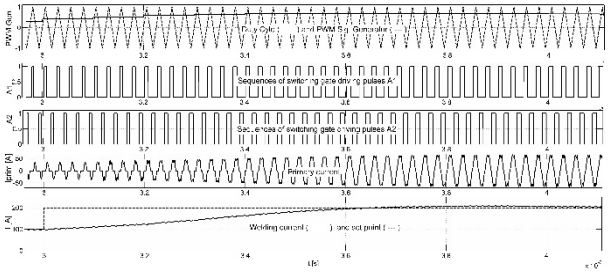


Figure 7: Simulation results of generated PWM signals, which depend on the duty cycle controlled with simple PI controller. The second and third plots show the PWM signals for driving the full-bridge DC-DC converter switches. The fourth and fifth plots show the corresponding changes of primary current and secondary - welding current.

In the first plot of Figure 6, the PWM frequency generator is compared with current controller output (duty cycle). In second and third plots the PWM signals for driving the full-bridge DC-DC converter switches are presented. The periods of pulses A1 and A2 changes depended on the duty cycle determined by the discrete PI controller. The maximum simulation step-size was 0.1 μs and the discrete PI controller sample time was 100 μs.

3.3 Simulation of Dynamic Behaviour of a Thyristor Based Weldin Power Source

The power of the welding power source with a three-phase transformer in a delta-star connection was set at 50 kVA. We selected turns ratio between 4.5 and 6. In transformer block, we can change all essential parameters, such as resistance and inductance of the primary and secondary windings, and the loss resistance in the core. The thyristor bridge converter was chosen as it reduces the reactive power with firing angle α beyond 60 deg (α > 60 deg).

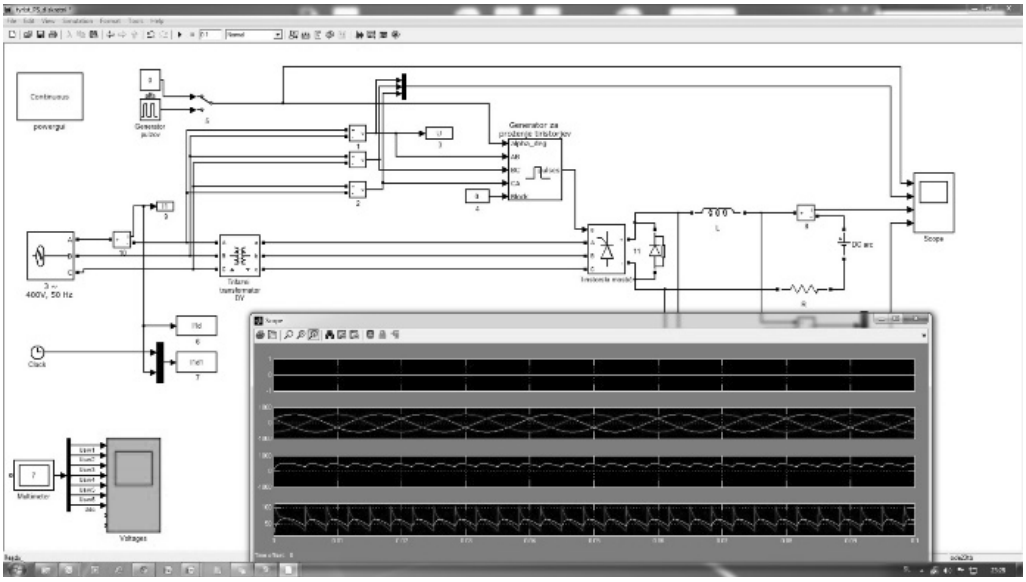


Figure 8: The simulation scheme of a three-phase welding source with thyristor full bridge circuits in Matlab/Simulink. The scope window shows the time graphs of signals within a time window of 100 ms.

For firing angles less than 60 deg, the DC voltage of the converter is always positive, and the freewheel diode does not come into operation. As the firing angle advances beyond this point, the load current starts to freewheel through the diode, thus cutting off the input line current and preventing the DC voltage from swinging into the negative direction. This reduces the amount of reactive power drawn from the mains, thus improving its power factor [15]. Figure 8 shows the simulation scheme in the program Matlab/Simulink and simulated voltage and current waveforms in a time window of 100 ms. The simulation was carried out with a firing angle of 0 deg.

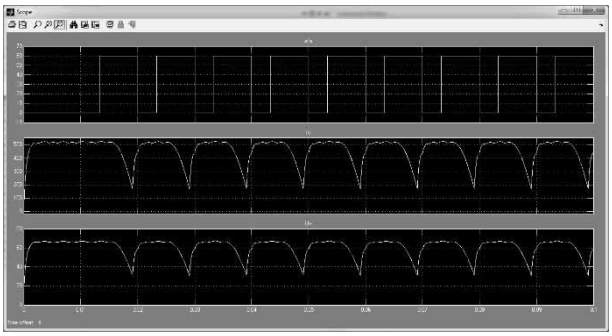


Figure 9: Voltage and current waveforms and variation of the firing angle by pulse frequency f_p = 100 Hz and pulse width t_p = 6.6 ms.

Figure 9 shows the output voltage, current and firing angle α for the pulse frequency f_p of 100 Hz. The firing angle α lies between α = 0 deg and α = 60 deg. The pulse width t_P can be varied as well. Figure 10 shows an example of a wider pulse.

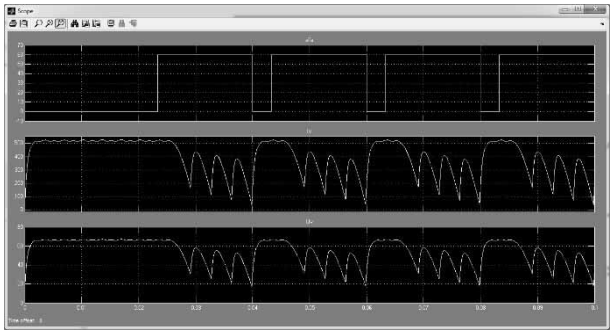


Figure 10: Voltage and current waveforms and variation of the firing angle by pulse frequency f_p = 50 Hz and pulse width t_p = 3.3 ms.

Pulsed welding with different pulse frequencies generates different harmonics on the AC side. The harmonic are accompanied by two adjacent frequencies on the AC side [15]. The Fourier analysis of the AC current can further verify this. For example, welding with a pulse frequency of 150 Hz produces harmonics of 100 and 200 Hz in the alternating current. Figure 11 show the Fourier analysis of the AC current during the simulation of the pulsed current welding process.

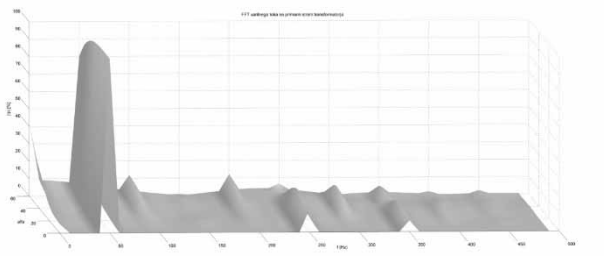


Figure 11: Fourier spectrum of the AC current during pulsed current welding with pulse frequency fp = 50 Hz and pulse width tP = 3.3 ms by changing the firing angle from 0 deg to 60 deg.

4 Conclusion

A simulation application has been presented for simulating the GMAW process, inverter-based welding sources, and thyritsor-based welding sources. The mathematical model is based on physical descriptions of several parts of the GMAW process, such as the electric circuits of the power supply, the arc dynamics, the electrode melting process, etc.

The simulation of inverter power source for welding power supply has been proposed and tested together with the GMAW simulation model. The simulation results showed that the conventional full-bridge DC-DC converter with appropriate current feedback controller makes the output welding current follow the set references.

The proposed models and simulations, which are combined together to simulate the power source circuits using simulations of the GMAW process, are suitable for the development of new power source circuits, i.e. resonant converters. By establishing appropriate models of the GMAW process and the full-bridge DC-DC converter model, simulation is an effective tool for investigating new welding technologies, for example the Pulsed GMAW process, or Surface Tension Transfer welding process (STT). Simulation results could be very useful for the rapid development of new control algorithms and for the designing of new inverter control units.

References

[1] Moore KL, Naidu DS, Yender R, Tyler J. "Arc Welding Control: Part 1 – Modeling and Analysis", *Nonlinear Analysis: Theory, Methods & Applications*, vol. 30, pp. 3101-3111, 1997.

[2] Moore KL, Naidu DS, Ozcelik S. *Modeling, Sensing and Control of Gas Metal Arc Welding.* Oxford, UK: Elsevier Science Ltd., 2003.

[3] Golob M, Koves A, Puklavec A, Torvornik B. "Modelling, simulation and fuzzy control of the GMAW process", in Conf. Proceedings of the 15th International Federation of Automatic Control (IFAC) - Triennial World Congress on Automatic Control, Barcelona, Spain, 2002, vol. 13, pp. 253-258.

[4] Thomsen JS. "Control of Pulsed Gas Metal Arc Welding", *International Journal of Modelling, Identification and Control*, vol. 1, no. 2, pp. 115-125, 2006.

[5] Zhang J, Walcott BL. "Adaptive Interval Model Control of Arc Welding Process", *IEEE Trans. On Control Systems Technology*, vol. 14, pp. 1127 - 1134, Nov. 2006.

[6] Bera MK, Bandyopadhyay B, Paul AK, Robust nonlinear control of GMAW systems-a higher order sliding mode approach. *IEEE International Conference on Industrial Technology* (ICIT), 2013, 175-180.

[7] Ngo MD, Duy VH, Phuong NT, Kim HK, Kim SB. "Development of digital gas metal arc welding system", *Journal of Materials Processing Technology*, vol. 198, no. 1-3, pp. 384-391, 2007.

[8] Golob M, Torvornik B, "Modelling, simulation and control of gas metal arc welding", in *Proceedings of the 7th Congress on Modelling and Simulation EUROSIM*, Prague, Czech Republic, 2010, pp. 347-352.

[9] Golob M. Integrated Models of a Gas Metal ARC Welding Process and Inverter based Power Supply for Process Control Simulation Studies. *ELEKTRONIKA IR ELEKTROTECHNIKA*. 2014; 20(7): 3-6.

[10] Lesnewich A. "Control of the Melting Rate and Metal Transfer in Gas Shielded Metal Arc Welding - Part 1", *Welding Journal*, vol. 37, pp. 343s-354s, 1958.

[11] Tusek J, Suban M. "Dependence of Melting Rate in MIG/MAG Welding on the Type of Shielding Gas Used", *Journal of Materials Processing Technology*, vol. 119, pp. 185-192, 2001.

[12] Halmøy E. "Wire melting rate, droplet temperature and effective anode potential", in *Proceedings of the International Conference on Arc Physics and Weld Pool Behaviou,* London, England, 1979, pp. 49-57.

[13] Krejcar O, Spicka I, Frischer R. “Implementation of Full-Featured PID Regulator in Microcontrollers”, *Electronics and Electrical Engineering*, vol. 113, no. 7, pp. 77–82, 2011.

[14] Petrovas A, Lisauskas S, Rinkeviciene R. “Digital Automatic Control System with PID Controller”, *Electronics and Electrical Engineering*, vol. 110, no. 4, pp. 13–16, 2011.

[15] Thamodharan M, Beck HP. in Wolf A.: Steady and Pulsed Direct Current Welding with a Single Converter. *Supplement to the Weldig Journal*, March 1999. 75s-79s.

[16] Langus D, Kralj V. in Grum J: Optimisation of welding parameters in pulsed MIG/MAG welding width-controlled sine-wave current pulses. Part 1: Determination of a general synergetic equation and a normalised parametric diagram with a defined parametric welding range. Int. j. mater. prod. technol., 2007, letn. 29, št. 1/2/3/4, str. 244-254

[17] Langus D, Kralj V. in Grum J: Optimisation of welding parameters in pulsed MIG/MAG welding width-controlled sine-wave current pulses. Part 2: Determination of an optimum material transfer through the arc and a control method. Int. j. mater. prod. technol., 2007, letn. 29, št. 1/2/3/4, str. 255-271.

Inverse Simulation Methods Applied to Investigations of Actuator Nonlinearities in Ship Steering

David J. Murray-Smith

Emeritus Professor and Honorary Senior Research Fellow, School of Engineering, Rankine Building, University of Glasgow, Glasgow G12 8QQ, Scotland, United Kingdom; *David.Murray-Smith@Glasgow.ac.uk*

Simulation Notes Europe SNE 26(4), 2016, 245 – 256
DOI: 10.11128/sne.26.tn.10356
Received: October 25, 2016
Accepted: December 5, 2016 (Special Issue Review)

Abstract. Actuators associated with control surfaces in aircraft, ships and underwater vehicles often introduce problems in terms of the control characteristics of the vehicle if significant saturation and rate limiting effects are present. Rate limits, in particular, have been linked to a number of well-publicised safety and handling-qualities issues for aircraft. Such limits also present difficulties in ship steering and ship autopilot systems. This paper describes an investigation of the effects of actuator nonlinearities involving a ship steering control application. The method of approach involves the use of inverse simulation to detect the onset of limiting. The paper shows that inverse simulation methods allow direct prediction of situations in which rudder saturation and rate limiting have significant effects in terms of the manoeuvrability of the vessel. It is also shown that a two-stage inverse-simulation method allows direct assessment of the difference between desired and achievable manoeuvres.

Introduction

Inverse dynamic models allow time histories of input variables to be found that permit a given set of output time- history requirements to be achieved. This has relevance for many dynamic problems, especially where actuator performance and limits are important. Inverse models have proved to be particularly useful for investigations involving systems in which a human operator has a central role.

Although analytical approaches to model inversion are of great value, they can present difficulties with many forms of nonlinear model. In recent years, extensive use has been made of simulation techniques for finding inverse solutions rather than depending entirely on analytical methods of inversion. Examples of applications of this kind include aircraft handling qualities investigations and agility studies, both for fixed-wing aircraft and helicopters (see, e.g., [1], [2]). In such cases the inverse solution provides vital information about the relative difficulty of performing different manoeuvres and about control margins available as actuator amplitude or rate limits are approached. In recent years much progress has also been made in using inverse simulation methods in control system design applications (see, e.g., [3], [4]).

1 Models of Actuators and Ship Steering Dynamics

Detailed, physically-based, models of actuators of various kinds are available in the literature and, whether the actuators are hydraulic, electro-hydraulic or electrical in form, the actuator systems have well-defined amplitude and rate limits. Along with the inherent dynamic characteristics of the actuator, these limits are important in determining overall performance of the vehicle or other system within which the actuator is an essential component. For example, actuator performance is of vital importance in aircraft flight control, as discussed in detail by Fielding and Flux [5]. A chronological bibliography of saturating actuators has been prepared by Bernstein and Michel [6] and this includes information from papers and reports involving the use of actuators in many different application areas.

In the case of actuators used for steering in marine vehicles a number of simplified actuator models have been proposed (see, e.g. [7]). Some of these relate directly to earlier work of van Amerongen [8] who, in the context of research on ship steering control systems, proposed the use of a simplified block diagram of the form shown in Figure 1.

This block diagram structure is also used for aeronautical engineering studies of actuator limiting in fixed-wing aircraft and helicopter flight control systems and can be modified quite readily to describe actuators which have second-order characteristics when operating linearly. In this case the block labelled G_r would no longer be a simple gain factor but would have first-order lag characteristics. In principle, accelaration limits as well as rate limits could be incorporated into this type of block diagram structure but this has not been considered in the present investigation. The structure shown in Figure 1, thus represents a general form of model which is capable of describing the linear and nonlinear characteristics of a wide range of actuators in a simple fashion and is appropriate for applications involving marine vehicles or aircraft.

Within the block diagram of Figure 1 the saturation limit block has a simple form and, when the input $G_a\delta_c$ lies in the range between the upper and lower saturation limits (δ_{cU} and δ_{cL}), it behaves as a linear gain element, having unity gain,. However, when the input $G_a\delta_c \geq \delta_{cU}$ the output value is limited at δ_{cU} and, correspondingly, when $G_a\delta_c \leq \delta_{cL}$ the output is limited at δ_{cL}. For many cases of practical importance this limiting behaviour is symmetrical for positive and negative inputs and $\delta_{cU} = -\delta_{cL}$. The rate limit block has an identical form, having unity gain when the output of the block (the rate of change ($\dot{\delta}(t)$) of the actuator ouput position $\delta(t)$) has values that lie within the specified upper and lower actuator rate limits. The rate limit block gives a constant output equal to the positive or negative rate limit when $\dot{\delta}(t)$ has a value beyond the specified upper and lower actuator rate limits.

The type of actuator model outlined above can be used with many different forms of ship model. One nonlinear form of model, which is commonly-used to represent the manoeuvring characteristics of course-stable ships in yaw, is an extended form of Nomoto's first-order model [7], [9] which relates heading variables to the rudder angle.

Although it is based on physical principles, the model involves a number of damping coefficients that must be estimated from data obtained experimentally. It has been shown to be a satisfactory representation for a range of operating conditions [7-9].

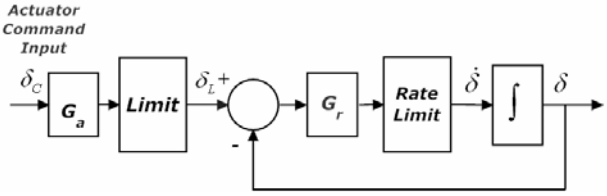


Figure 1: Simplified block diagram of actuator system with amplitude and rate limiting.

The basic model is given by:

$$T\ddot{\psi} + H_N(\dot{\psi}) = K\delta(t) \quad (1)$$

where the variable ψ is the yaw angle (heading) of the vessel, δ is the rudder angle, T is an inertia constant and the function H_N, which is a function of the rate of change of heading ($\dot{\psi}$) is given by:

$$H_N(\dot{\psi}) = n_1\dot{\psi} + n_3(\dot{\psi})^3 \quad (2)$$

where n_1 and n_3 are positive damping constants, known as Norrbin coefficients. For the specific case of the *R.O.V. Zeefakkel*, which is a 45 m long training ship belonging to the Royal Netherland Naval College, the parameters n_1 and n_3 have been estimated for a number of different forward speeds [8]. Combining Eqns (1) and (2) gives:

$$\delta = m\ddot{\psi} + d_1\dot{\psi} + d_3(\dot{\psi})^3 \quad (3)$$

where $m = \frac{T}{K}$, $d_1 = \frac{n_1}{K}$ and $d_3 = \frac{n_3}{K}$. Values of these parameters vary significantly for typical speed values over the range of interest for this vessel, as shown in Table 1.

In this application, the rudder and its associated actuator are modelled using the first-order lag type of description with input saturation and rate limits, as shown in Figure 1. In the linear mode of operation an actuator time constant of 3 s is given by a value of the gain factor G_r of 0.333.

If the required rudder deflection δ_c is the variable subjected to limiting, the gain factor G_a in Figure 1 is unity. For the purposes of this investigation the saturation limit for the rudder is typically of the order of ±35 deg, while the two different rate limit values used in the illustrative examples that follow are ±7 deg/s and ±10 deg/s.

Forward speed (U m/s)	T	K	$m = \frac{T}{K}$	d_1	d_3
2.6	33.0	0.19	173.68	3.3330	3.7037
5	31.0	0.50	62.00	2.0000	0.8000

Table 1: Parameter values used for the model of R.O.V. Zeefakkel [8].

2 Inverse Simulation Methods

Inverse simulation techniques may be divided conventiently into methods that are based on discretised models and are essentially iterative in nature and techniques that are based on continuous system simulation principles. Although the emphasis within this paper is on use of one of the second group of methods, both types of approach are reviewed here since some continuous system simulation approaches have origins in iterative methods involving discretised models.

2.1 Iterative methods of inverse simulation based on discrete models

Several inverse simulation techniques were developed initially for aircraft handling qualities and agility investigations, as mentioned above. The technique that is most widely used was developed first by Hess, Gao and Wang [10] and involves repeated solution of a forward simulation model of the vehicle to allow determination, in an iterative fashion, of inputs that allow the output to follow a specified manoeuvre. This has been termed an 'integration-based' approach. Very similar techniques were developed independently by Thomson and Bradley and their colleagues (see, e.g., [11], [1], [2]). This type of iterative technique is based on the use of gradient methods but search-based optimization methods have also been applied, with success, in a range of applications (see e.g. [12]). Another method, which can be traced back to original work in the aircraft flight mechanics field, involves use of a so-called 'differentiation' method in which a continuous system model of the given system is transformed into a discrete-time description through the use of a finite difference approximation. This approach was developed by Thomson and his colleagues (see, e.g. [13], [14]) in the context of helicopter applications and by Kato and Sugiura [15] for fixed-wing aircraft problems.

Other iterative techniques were also developed for similar applications, including optimization-based approaches by Celi [16] and by Lee and Kim [17]. The paper by Thomson and Bradley [2] provides a useful overview of a number of these iterative techniques, as developed initially for aeronautical applications. Inverse simulation techniques based on discrete forms of model have also been used for the design of model-based output-tracking control systems and a paper by Lu, Murray-Smith and McGookin [3] describes the use of inverse simulation in the design of feed-forward control systems based on a Lynx helicopter model and also for combined steering control and roll stabilisation in a container ship application.

2.2 The continuous system simulation approach

Although the iterative type of approach has been used with considerable success in a number of aeronautical applications, a second (and entirely different) approach to the development of inverse simulation methods has evolved which is based on the use of continuous system simulation principles and avoids the need for iterative solutions.

One approach is based upon the numerical solution of differential algebraic equations (DAEs) (see, e.g. [18], [19]), using DAE solvers. However, it appears that, at present, numerical issues have limited the application of this method to cases involving relatively simple low-order models.

Two other approaches to inverse simulation using continuous system simulation principles are currently available. One of these involves the use of feedback methods (see, e.g., [20], [21]) while the second is based upon an approximate method of differentiation (see, e.g., [22]). From experience gained with other applications, it is known that in the approximate differentiation approach any changes in the structure of the forward model require restructuring of the inverse simulation model and this can be time consuming. In contrast, in the feedback method, changes within the model can be incorporated without changes in the feedback structure (other than possible adjustments of some feedback loop gains). For this reason the feedback approach has been chosen for the work described in this paper.

2.3 Principles of the feedback approach

Some of the earliest developments in inverse simulation involving the use of feedback principles can be found in work carried out at the DLR aeronautical research institute at Braunschweig in Germany, as outlined by Hamel (see e.g. [23]), and discussed in more detail by Gray and von Grünhagen [24] and by Buchholz and von Grünhagen [25]. These methods have more recently been used in a number of applications involving aircraft, process systems and underwater vehicle models (see e.g. [20-21], [26-28]).

A similar type of approach, which is linked specifically to control system design, has been developed by Tagawa and Fukui [29]. Their overall approach is termed 'inverse dynamics compensation via simulation of feedback control systems' (IDCS) and the dervation of an inverse simulation through the use of feedback is a central element of this control design methodology. They have used the IDCS method in control system design applications involving servo-hydraulic actuators and robotics, as described in recent papers [30], [4].

The feedback approach to inverse simulation can best be understood by considering the case of a linear model. The block diagram of Figure 2 involves a single-input single-output linear model $G(s)$ and a feedback loop having a cascaded block with transfer function $K(s)$. The transfer function relating the variable $W(s)$ to a reference input $V(s)$ is given by:

$$\frac{W(s)}{V(s)} = \frac{1}{\frac{1}{K(s)} + G(s)} \tag{4}$$

If the term $1/K(s)$ is very small compared with the magnitude of $G(s)$, over the range of frequencies of interest, the transfer function may be approximated by:

$$\frac{W(s)}{V(s)} \approx \frac{1}{G(s)} \tag{5}$$

Thus, if $K(s)$ is large, the transfer function $W(s)/V(s)$ is a close approximation to the inverse model.

Although a linear single-input single-output system model is used here, the same principles apply to the case of nonlinear models and to multi-input multi-output model structures. While the use of simple high-gain feedback provides acceptable solutions in many cases, it should be noted that the principle of feedback-based model inversion applies also to other feedback structures and the approach is not limited to proportional control methods or to linear models.

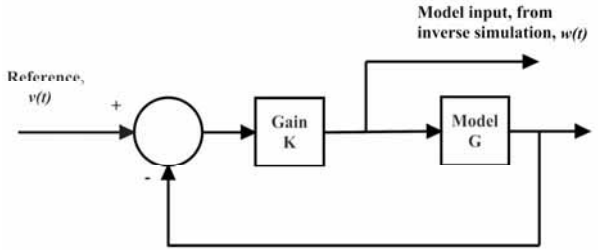


Figure 2: Block diagram for inverse simulation using feedback principles for a given linear or nonlinear model *G*. For a high value of the gain *K*, the variable *w* is a close approximation to the model input required to produce an output that matches a given time history *v*(*t*).

In its origins, the feedback-based approach can be linked back to the use of feedback principles for division and inverse function generation operations in electronic analog computers. Recent work has shown that the approach has very wide applicability [20] and that it allows analysis of the dependence of inverse solutions on parameters of the forward model (without parameter perturbation) through the use of sensitivity models [26]. This can have advantages, especially in the linear case, in terms of the additional physical insight provided when compared with parameter perturbation methods for sensitivity investigation.

One potential problem in applying the feedback-based approach to problems involving actuator saturation and rate limits concerns difficulties arising from possible limit cycle effects. Hard nonlinearities of the type that arise in actuators can give rise to limit cycle phenomena within any feedback loop. For single-input single-output feedback systems, describing function analysis methods (see, e.g. [5], [31]) can be used to predict the existence of limit cycles for feedback systems which involve one dominant nonlinearity and, otherwise, can be described adequately by linear dynamic elements within the feedback loop. The conditions associated with the onset of limit cycle oscillations depend critically on the order of that linearised model and on the form of the nonlinearity. In general, the higher the order of the linear model the more likely it is that limit cycle phenomena will be encountered when saturation or rate limiting effects are present within the feedback loop. Also, nonlinear elements which have describing functions which have a complex form (with imaginary as well as real components) are more likely to give rise to limit cycle oscillations, as discussed by Fielding and Flux [5].

This means that problems of limit cycles are likely to be encountered in attempting to apply the feedback approach to inverse simulation in the case of applications involving significant rate limits. Therefore, in such cases, some modifications to the standard feedback approach may be necessary or entirely different methods of inverse simulation may have to be applied that do not involve the use of feedback.

3 Inverse Simulation Applied to the Ship Model

The first approach considered involves the application of the simple feedback method of inverse simulation, as outlined in Section 2.3 and discussed in greater detail elsewhere (see, e.g. [20], [26], [27]).

3.1 Feedback applied to the ship model with the actuator sub-model included.

Figure 3 is a block diagram which shows the structure of the feedback system which is applied around the ship model, including the actuator sub-model which, in the general case, incorporates saturation and rate limits. The signal used to represent the desired response of the vessel is generated using a reference model. In general terms this must involve a defined output that is consistent with the dynamics of the vessel, with smooth derivatives in order to give realistically smooth actuator control demand movements. In this application the reference input is generated using a third-order reference model which provides appropriate inputs, either in terms of the desired rate of change of heading or the desired heading. In the case involving the desired heading, the structure and parameter values of this reference model are chosen to give a reference signal which rises smoothly from zero to a specified final value of heading over a period of about 30 s. This, together with the corresponding heading-rate reference input, represents appropriate steering dynamics for a vessel of the type being considered. The heading-rate signal from the reference model is used as the reference input in Figure 3.

Feedback was provided by the heading-rate signal which was compared with the heading-rate reference to produce the heading-rate error which was then amplified by the gain *K*.

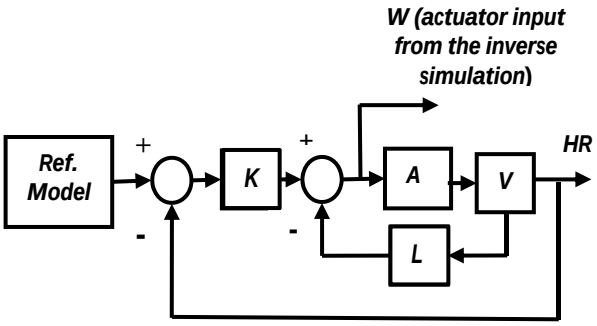


Figure 3: Block diagram of the feedback system used for inverse simulation with the actuator sub-model incorporated within the feedback loop. Here the block *A* represents the actuator and *V* represents the vehicle. The variable *HR* is the vehicle heading rate. The reference model generates the time history of the desired manoeuvre in terms of the required heading-rate time history. The block shown as having a gain factor *L* is a subsidiary feedback loop and, in the case of the application considered here, involves angular acceleration feedback.

As shown in Figure 3, an additional feedback pathway with a gain factor *L* was provided from the heading acceleration signal within ship model as this was found to be beneficial and provided additional damping. Appropriate values for the gain factors *K* and *L* in Figure 3 were determined using basic feedback theory, with some further trial-and-error optimization. A suitable value for the gain factor *K* in the heading-rate feedback loop was found to be 10×10^6 while an appropriate value for the gain factor *L* in the subsidiary loop involving feedback of the angular acceleration was found to be 10×10^4. Results from the inverse simulation studies were found to be relatively insensitive to the precise values used in these feedback loops, provided the two gain factors remained large.

Figures 4-9 show results obtained from the feedback system for a case involving a forward speed of 5 m/s and a demanded heading change of 8 deg. The reference signal is the heading-rate signal obtained from the reference model (Figure 4), corresponding to the heading change shown in Figure 5. The saturation limit in this case is ±35 deg and the rate limits are ±10 deg/s.

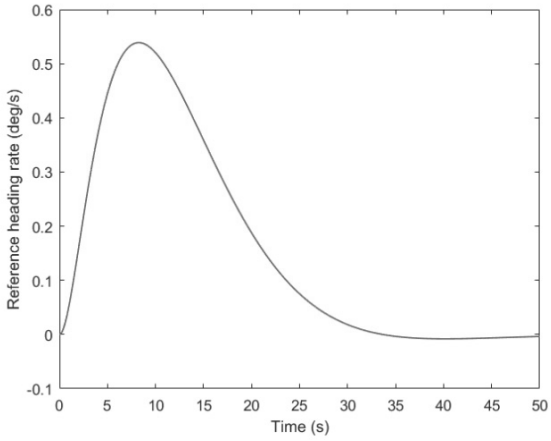
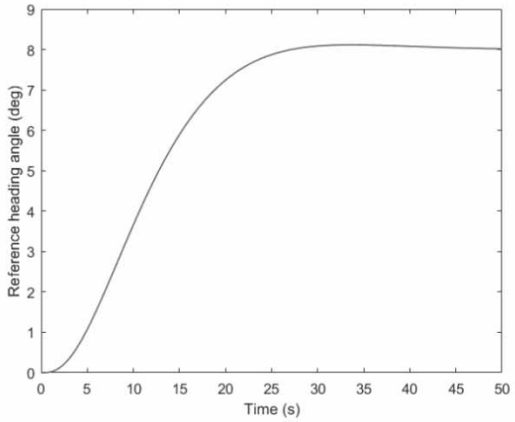


Figure 4: Reference input applied to the feedback system for the case of the ship model with forward speed of 5 m/s and a demanded heading change of 8 deg.

The results in Figures 6 and 7 show that, for the chosen manoeuvre and forward speed condition, the rudder did not approach its angular saturation limit of ±35 deg or its angular rate limit of ±10 deg/s. When applied as input to the forward simulation model, the rudder deflection found from inverse simulation (as shown in Figure 6) produced a heading-rate response which matched almost exactly the required heading rate with heading-rate errors less than $\pm 12\times10^{-4}$ deg/s over the 50 second response time considered (as shown in Figure 8). This corresponds to a maximum heading error (as shown in Figure 9) of approximately 5.5×10^{-3} deg. Errors in heading angle and heading rate would of course be slightly different for other values of gain factors in the feedback pathways and, in particular, would increase if the gain in the heading-rate feedback loop were reduced significantly. This level of agreement is typical of results found using the feedback method outlined in Section 2, for cases where actuators operate within their limits.

However, if the forward speed of the ship is reduced to 2.6 m/s, the situation changes. At this lower forward speed the manoeuvre is more demanding than that considered in the previous example. Figures 10 and 11 show results for the same 8 deg demanded course change and, it can be seen that the required rudder rate goes well beyond the limit of 10 deg/s, although the rudder deflection does not reach the saturation level of 35 deg.



. **Figure 5:** Heading change corresponding to the heading-rate reference signal of Figure 4.

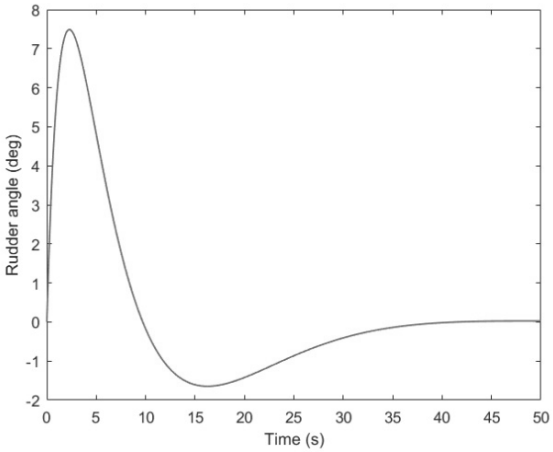


Figure 6: Rudder angle time history found using the inverse simulation process for the ship model with forward speed of 5 m/s.

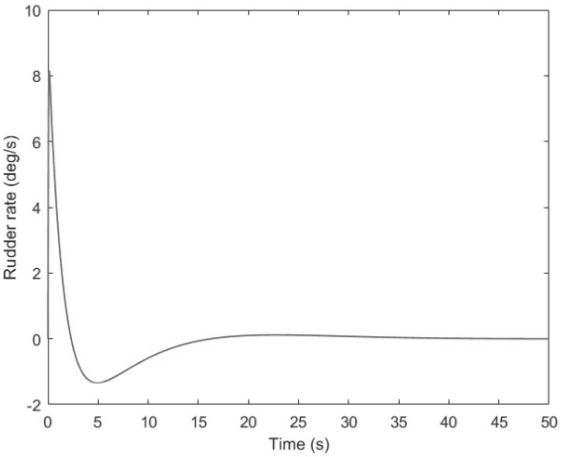


Figure 7: Rudder angular velocity time history found using the inverse simulation process for the ship model with forward speed of 5 m/s.

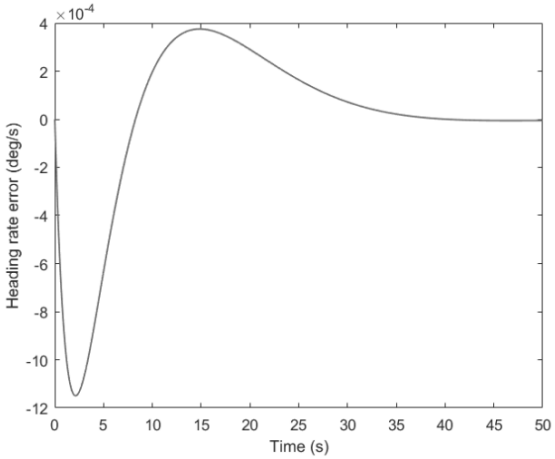


Figure 8: The difference between the heading-rate reference input and the heading-rate found from a forward simulation using the rudder deflection time history of Figure 6.

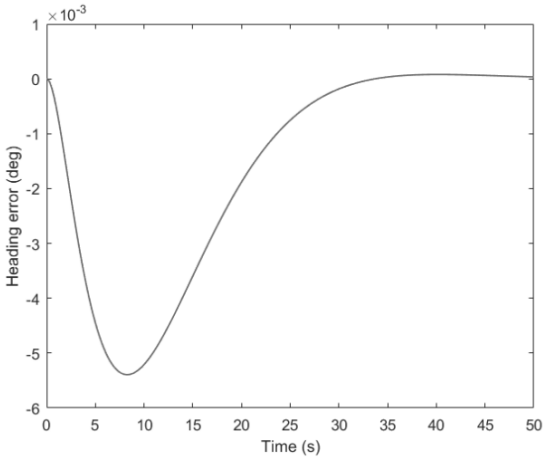


Figure 9: The error in heading corresponding to the results shown in Figure 8.

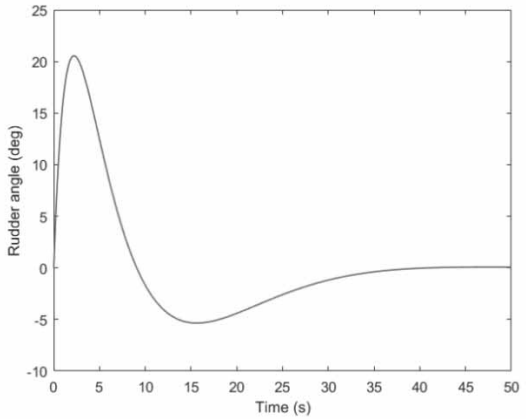


Figure 10: Rudder angle time history found for forward speed of 2.6 m/s. Other conditions for this simulation are the same as for the previous results.

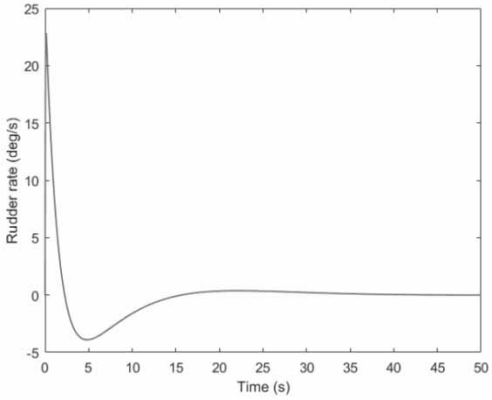


Figure 11: Rudder angular-rate time history found for forward speed of 2.6 m/s. Other conditions for this simulation are the same as for the previous results.

If the demanded heading change is now made larger for the forward speed of 5 m/s, the saturation and rate limits both become important. Figures 12 and 13 show results for a forward speed of 5 m/s and a desired manoeuvre involving a final course change of 40 deg. Clearly the actuator position (rudder angle) now exceeds the 35 deg saturation limit and the angular velocity also exceeeds the 10 deg/s rate limit. This procedure gives a clear indication of situations where demanded manoeuvres exceed the hard limits of the actuator and could cause problems in terms of the control characteristics of the vessel. Thus, if the purpose of the investigation is to establish whether or not a specific manoeuvre gives rise to saturation or rate limiting, the inverse simulation model involving a linear actuator sub-model can provide useful information.

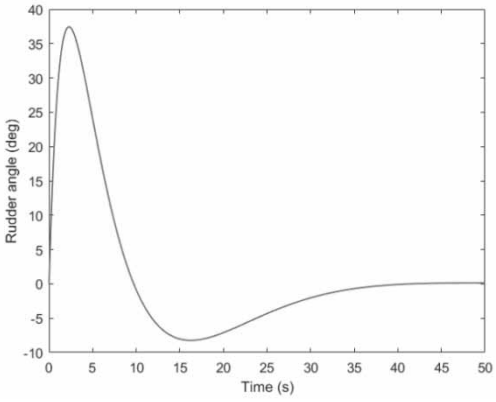


Figure 12: Rudder angle time history found for forward speed of 5 m/s for a demanded manoeuvre corresponding to a 40 deg heading change. Other conditions for this inverse simulation are the same as for the previous results.

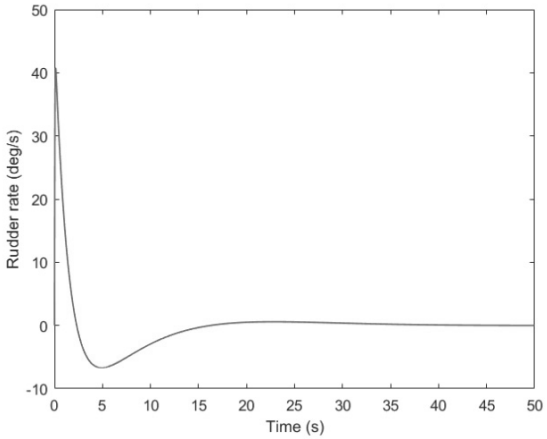


Figure 13: Rudder angular-rate time history found for forward speed of 5 m/s s for a demanded manoeuvre corresponding to a 40 deg heading change.

It can also be concluded that, for less demanding inputs, such as the 8 deg manoeuvre considered in Figures 4-9, inverse simulation methods do provide a direct and clear indication of the margins of control available. In terms of the saturation limit this margin is found from the difference between the maximum rudder deflection and the saturation limit. For the rate limit, the corresponding margin is found by comparing rudder angular velocity values over the complete time history with the rate limit value. In Figure 6 the maximum rudder deflection is about 7.5 deg compared with the saturation limit of 35 deg and there is therefore a large margin of control (27.5 deg of rudder deflection) before the helmsman or autopilot system would encounter problems. Similarly the results of Figure 7 show that the rate of change of rudder angular deflection of about 8 deg/s is below the critical level of 10 deg/s and this suggests that the manoeuvre could be made slightly more demanding before difficulties due to rate limits would be encountered.

The availability of information of this kind is clearly useful in assesing the maneuvrability of a specific vehicle or in considering specific design changes (such as within the actuator and rudder system).

If saturation and rate limits are included within the actuator model, the feedback structure used for inverse simulation changes its behaviour significantly in manoeuvres for which actuator limits are exceeded.

For example, Figure 14 shows the rudder deflection generated from the inverse simulation for a manoeuvre involving a 30 deg change of heading with a forward speed of 2.6 m/s and with a saturation limit of ±35 deg and rate limit of ±7 deg/s. This time history has a very different character from those considered previously and shows a transient which displays limit cycle type oscillations. Although this is not a stable limit cycle phenomenon, investigation based on describing function methods suggests that this transient is an artefact of the feedback methodology and arises as a result of the inclusion of the actuator rate limit. It should be noted that the use of heading feedback rather than heading-rate feedback tends to make this limit cycle behaviour even more pronounced.

In applications where investigation of the effect of actuator limits on the overall dynamic characteristics of the complete vehicle is important, some way must be found of incorporating the nonlinear actuator sub-model within the inverse simulation procedure. In view of the limit cycle problems encountered when the nonlinear actuator sub-model is included within the inverse simulation (as reported above) the simple feedback approach is clearly inappropriate. One possible strategy is outlined in the next section and involves a combination of inverse simulation and conventional forward simulation in a two-stage procedure [28].

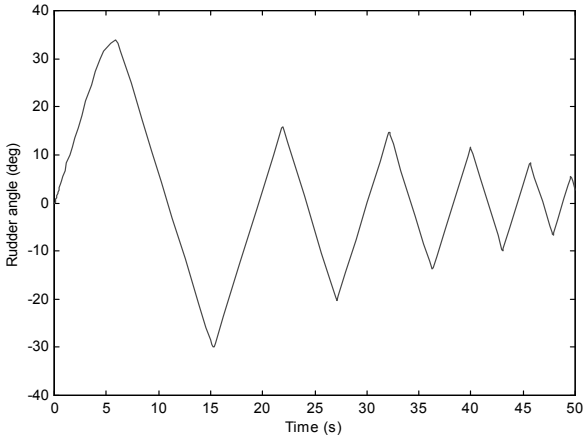


Figure 14: Results in terms of rudder angle obtained by inverse simulation using the feedback approach for the case of the ship model with forward speed of 2.6 m/s and a demanded heading change of 30 deg with a rudder saturation limit of ±35 deg and rudder rate limit of ±7 deg/s.

3.2 A two-stage feedback method

Figure 15 is a block diagram illustrating a two-stage method which allows the feedback approach to be used but which avoids the limit cycle problems encountered with the traditional method in which the nonlinear actuator model is included within the feedback loop. As before, feedback is applied around the ship model to allow a rudder input to be found that produces a heading rate that best matches the reference heading rate. The actuator sub-model is included within the feedback loop, but without the saturation and rate limits. In the first stage of the procedure, inverse simulation based on the feedback structure is used to find an input to this linear actuator model to achieve the desired response if no limits were present. The effect of including the saturation and rate limits is then investigated in the second stage by applying this idealised actuator input found from inverse simulation to a forward simulation of the ship involving the full nonlinear actuator sub-model.

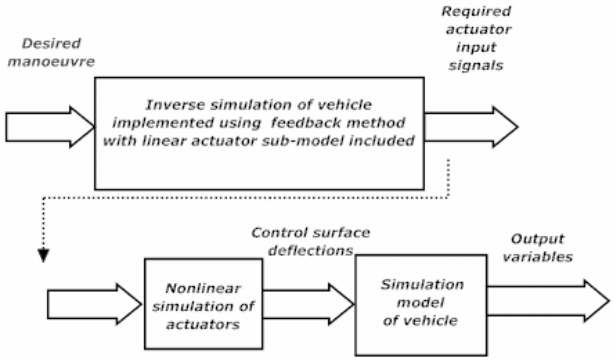


Figure 15: Block diagram of the two-stage procedure for inverse simulation using the feedback method.

Results obtained using this approach are shown in Figure 16 for a case involving a 30 deg heading change for a forward speed of 2.6 m/s with a rudder deflection limit of 35 deg, as before, and a rate limit of ±7 deg/s. The time-history of the rudder response indicates clearly that the rudder moves at the positive rate limit of 7 deg/s for an initial period of about 4 to 5 s, by which time the rudder angle is close to the saturation limit of 35 deg. The rudder then starts to move in the opposite direction and almost immediately reaches the negative rate limit of -7 deg/s. This rate is maintained for a further period of about 6 s, after which the rudder response enters a linear mode of operation, with the rudder angle approaching zero in the final 5 s of the response.

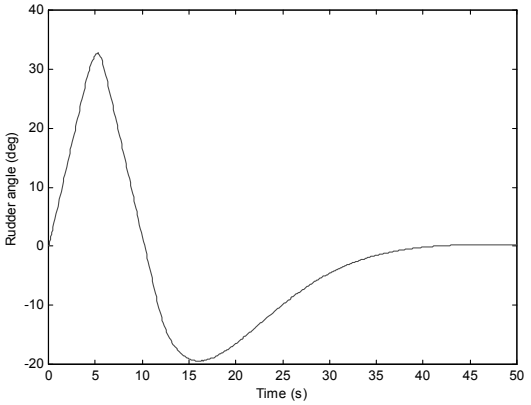


Figure 16: Results in terms of rudder angle obtained by inverse simulation using the two-stage approach for the case of the ship model with forward speed of 2.6 m/s and a demanded heading change of 30 deg with a rudder saturation limit of ±35 deg and rudder rate limit of ±7 deg/s.

Since the primary feedback loop used for the inverse simulation involves comparison of the rate of change of the heading of the vessel with the corresponding quantity from the reference model, it is appropriate to examine the heading-rate error when the rudder deflection time history is used as input to a forward model of the vessel. This is shown in Figure 17 for the 50 s test under consideration. The largest error (about 1.2 deg/s) occurs after about 5s, at the end of the initial period of rudder actuator rate limiting. As would be expected, the heading error found from this forward simulation builds up steadily over the complete time history and reaches almost 35 deg after 50 seconds, as shown in Figure 18.

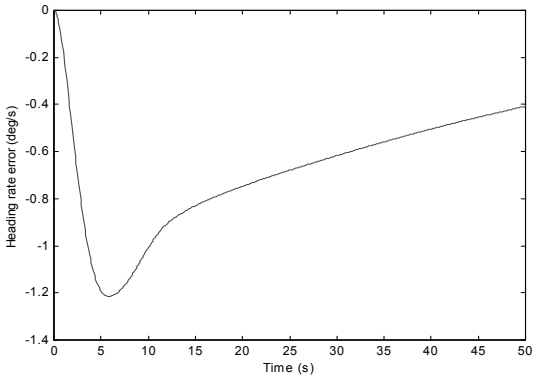


Figure 17: Heading rate error from forward simulation (second stage of the two-stage inverse simulation procedure) using the rudder deflection time history of Figure 16.

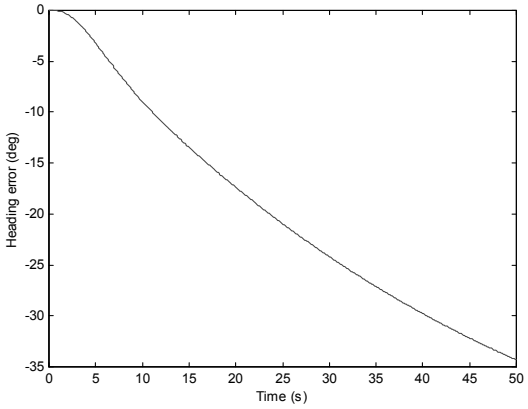


Figure 18: Heading error from forward simulation (second stage of the two-stage inverse simulation procedure) using the rudder deflection time history of Figure 16.

4 Discussion and Conclusions

It can be concluded, from this application, that inverse simulation methods based on feedback principles can provide useful information about the margins of control available before limiting effects in actuators lead to a downgrading of system performance. This is potentially very important in systems involving manual control where actuator input saturation and rate limits can give rise to undesirable oscillatory phenomena such as the pilot-induced oscillations that have been observed in aircraft flight testing. Knowledge of conditions associated with the onset of actuator saturation and rate limiting is also important for the design of automatic control systems, as has been discussed previously in the context of ship control (see, e.g. [7]) and aircraft flight control (see, e.g., [5]).

Inverse simulation techniques are particularly important in all of the above areas because they allow information to be gathered directly about how the input that is needed to perform a specific manoeuvre is affected by the operating condition and parameters of the model.

In order to investigate the effects of actuator saturation and rate limiting on the overall model output, a two-stage inverse-simulation approach has been shown to be useful. This avoids artefacts of the feedback approach which can lead to undesirable limit cycle oscillations. Indeed, it could even be argued that the feedback method of inverse simulation ceases to be valid when hard limiting occurs since the feedback loop then becomes transiently inactive.

However, in most practical situations involving hard limiting of actuators, we are concerned primarily with detecting conditions when limiting occurs and with finding ways of avoiding these, rather than obtaining a complete time-history of the outputs from the inverse simulation model.

It should be noted that the conventional single-stage feedback method of inverse simulation still has practical value for cases in which rate limiting does not occur. This allows inverse simulation to be used as a general-purpose design tool and can assist the designer in investigating the performance of different planned configurations at an early stage in the design process. For example, it can provide answers to questions about the capability of the vehicle under investigation, with known power and control limits, to perform a specified manoeuvre. If it is found that the manoeuvre cannot be carried out inverse simulation may help the designer to make configurational changes, such as a change of actuator characteristics or rudder area that then allow the design requirements to be satisfied.

In general terms, it can be concluded from this application that looking directly at inputs required to perform specific manoeuvres can provide insight that is significantly different from that available using conventional forward simulation tools. The fact that inverse simulation allows the sensitivity of the required input to changes of model parameters to be investigated directly is an important benefit in terms of the design process.

In terms of the specific results obtained in this application, further adjustment of the gain factors in the feedback pathways could be considered and could further improve the accuracy of the inverse simulation results. However, as always, a compromise has to be found between accuracy and computational speed and convenience.

It should be noted that the design of a feedback system for inverse simulation is significantly different from the design of a feedback control system involving a plant model of equivalent complexity. Issues of the robustness of the feedback system in terms of its response to external disturbances, measurement noise and parametric uncertainties, do not have to be considered. This means that less-robust design methods that might be considered inappropriate for control system applications, such as high gain solutions or eigenstructure design methods, can often prove useful in the development of inverse simulations.

References

[1] Thomson DG, Bradley R. The principles and practical application of helicopter inverse simulation. *Simulation Practice and Theory*. 1998; 6 (1): 47-70. doi:10.1016/S0928-4869(97)00012-8.

[2] Thomson D, Bradley R. Inverse simulation as a tool for flight dynamics research - Principles and applications. *Progress in Aerospace Sciences*. 2006; 42 (3): 174-210. doi:10.1016/j.paerosci.2006.07.002.

[3] Lu L, Murray-Smith DJ, McGookin, EW. Investigation of inverse simulation for design of feedforward controllers. *Mathematical and Computer Modelling of Dynamical Systems.* 2007; 13 (5): 437-454. doi:10.1080/ 13873950701344023.

[4] Tagawa Y, Tu JY, Stoten DP. Inverse dynamics compensation via 'simulation of feedback control systems'. *Proc Inst. Mech Eng, Part I: Journal of Systems and Control Engineering*. 2012; 225: 137-153. doi: 10.1243/09596518JSCE1050.

[5] Fielding C, Flux PK. Nonlinearities in flight control systems. *Aeronautical Journal*. 2003; 107(1077): 673-686. doi: 10.1017/S0001924000013543

[6] Bernstein DS, Michel AN., A chronological bibliography of saturating actuators. *International Journal of Robust and Nonlinear Control*. 1995; 5: 375-380. doi: 10.1002/rnc.459005050.

[7] Fossen, TI. *Guidance and Control of Ocean Vehicles*, Chichester, UK: Wiley; 1994. 494 pp.

[8] van Amerongen, J. *Adaptive Steering of Ships: A model-reference approach to improved manoeuvring and economical course keeping.* [PhD Dissertation], Delft University of Technology, The Netherlands; 1982. uuid:2edf96d7-ed7f-40a5-a927-72ff62dea72e.

[9] Kocijan J, Murray-Smith DJ. Robust nonlinear control for ship steering. In Mastorakis NE, editor, *Progress in Simulation, Modeling, Analysis and Synthesis of Modern Electrical and Electronic Systems*, Danvers, Mass. USA, World Scientific and Engineering Society Press, 1999, 235-240.

[10] Hess RA, Gao C, Wang SH. A generalized technique for inverse simulation applied to aircraft maneuvers. *AIAA Journal of Guidance Control and Dynamics*. 1991; 14 (5): 920-926. doi: 10.2514/3.20732.

[11] Rutherford S, Thomson DG. Improved methodology for inverse simulation. *Aeronautical Journal*. 1996; 100 (993): 79-86. doi:10.1017/S0001924000067348 .

[12] Lu L, Murray-Smith DJ, Thomson DG. Issues of numerical accuracy and stability in inverse simulation. *Simulation Modelling Practice and Theory*. 2008; 16(9): 1350-1364. doi: http://dx.doi.org/10.1016/j.simpat.2008.07.003

[13] Thomson DG, Bradley R. An analytical method for quantifying helicopter agility. In: *12th European Rotorcraft Forum, Garmisch-Partenkirchen, Federal Republic of Germany, September 1986*. Paper 45.

[14] Thomson DG, Bradley R. Recent developments in the calculation of inverse solutions of the helicopter equations of motion. In: UK Simulation Council Conference, University College of North Wales, 9-11 September 1987, Ghent, Belgium: UKSC, 1987, 227-234.

[15] Kato O, Sugiura I. An interpretation of airplane motion and control as inverse problems. *AIAA Journal of Guidance, Control and Dynamics.* 1986; 9 (2): 198-204. doi: 10.2514/3.20090.

[16] Celi R. Optimisation based inverse simulation of a helicopter slalom maneuver. *AIAA Journal of Guidance Control and Dynamics.* 2000; 23 (2): 289-297. doi: 10.2514/2.4521

[17] Lee S, Kim Y. Time domain finite element method for inverse problem of aircraft maneuvers. *AIAA Journal of Guidance, Control and Dynamics.* 1997; 20 (1): 97-103. doi: 10.2514/2.4000

[18] Blajer W, Kołodziejczky K. Improved DAE formulation for inverse dynamics simulation of cranes. *Multibody System Dynamics*. 2011; 25 (2): 131-143. doi:10.1007/s11044-010-9227-6.

[19] Thümme M, Looye G, Kurze M, Otter M, Bals J. Nonlinear inverse models for control. In Schmitz G, editor, *Proceedings of the 4th International Modelica Conference, Hamburg, Germany, 7-8 March 2005*. Linköping, Sweden; Modelica Organisation, 2005. 267-279.

[20] Murray-Smith DJ. Feedback methods for inverse simulation of dynamic models for engineering applications. *Mathematical and Computer Modelling of Dynamical Systems.* 2011; 17 (5): 515-541. doi:10.1080/13873954.2011.584323.

[21] Murray-Smith DJ. *Modelling and Simulation of Integrated Systems in Engineering: Issues of Methodology, Quality, Testing and Application.* Cambridge, UK: Woodhead; 2012. 372 pp.

[22] Murray-Smith DJ. An approximate differentiation method of inverse simulation based on a continuous system simulation approach. *Simulation Notes Europe*. 2013; 23(3-4):111-116. doi:10.11128/sne.23.tn.10201.

[23] Hamel PG. Aerospace vehicle modelling requirements for high bandwidth flight control. In Cook MV, Rycroft MJ, editors, *Aerospace Vehicle Dynamics and Control*, Oxford, UK; Clarendon Press, 1994. 1-31

[24] Gray GJ, von Grünhagen W. An investigation of open-loop and inverse simulation as nonlinear model validation tools for helicopter flight mechanics. *Math. and Comp. Modelling of Dynamical Syst.,* 1998; 4: 32-57. doi: 10.1080/13873959808837067.

[25] Buchholz JJ, von Grünhagen, W. *Inversion Impossible?* Technical Report, University of Applied Sciences Bremen, Germany, 2004.

[26] Murray-Smith DJ. The application of parameter sensitivity analysis methods to inverse simulation models, *Mathematical and Computer Modelling of Dynamical Systems*, 2013; 19 (1): 67-90. doi:10.1080/13873954.2012.696271.

[27] Murray-Smith DJ. Inverse simulation and analysis of underwater vehicle dynamics using feedback principle, *Mathematical and Computer Modelling of Dynamical Systems*, 2014; 20 (1): 45-65. doi: 10.1080/13873954.2013.805146.

[28] Murray-Smith DJ, McGookin EW. A case study involving continuous system simulation methods of inverse simulation for an unmanned aerial vehicle application. *Proc. Inst Mech Eng, Part G, Journal of Aerospace Engineering.* 2015; 229(14): 2700-2717. doi: 10.1177/0954410015586842

[29] Tagawa Y, Fukui K. Inverse dynamics calculation of nonlinear model using low sensitivity compensator. In: *Proceedings of Dynamics and Design Conference. 1994*, 185-188.

[30] Venture G, Kojima T, Tagawa Y. Fast motion control of robotic systems using inverse dynamics compensation via 'simulation of feedback control systems' (IDCS). In: *Proc. 1st Joint Conf. on Multibody System Dynamics, May 25-27 2010, Lapeenranta, Finland*.

[31] Atherton DP. *Nonlinear Control Engineering*, New York, USA:Van Nostrand Rienhold; 1982. 470 pp.

Conversion of Iterative Balance Models to Directly Calculating Explicit Models for Real-time Process Optimization and Scheduling

Tomas Björkqvist[1]*, Olli Suominen[1], Matti Vilkko[1], Mikko Korpi[2]

[1] Department of Automation Science and Engineering, Tampere University of Technology, Tampere, Finland; * *tomas.bjorkqvist@tut.fi*

[2] Research Center Pori, Outotec (Finland) Oy, Pori, Finland

Simulation Notes Europe SNE 26(4), 2016, 257 - 266
DOI: 10.11128/sne.26.tn.10357
Received: November 10, 2016
Accepted: December 5, 2016 (Special Issue Review)

Abstract. Optimal utilization of complex processes involves real-time operational optimization and scheduling, especially in cases where the production line consists of both continuous and batch operated unit processes. This kind of real-time optimization requires process models which can be computed significantly faster than real-time. Iterative balance calculation is typically far too slow for these cases. This paper presents a method for converting an iterative balance model to a directly calculating model suitable for on-line process optimization. The approach is demonstrated with the first unit process in the copper smelting line, the flash smelting furnace (FSF). The method consisted of formulating an equation group based on the constrained FSF HSC-Sim model and solving the unknown parameters and static states with use of a symbolic calculation software. The solution was implemented as a function whose calculation time fulfilled the requirements for scheduling use.

Introduction

The general digitalization of society and advances in computational power have brought on a pronounced digitalization wave in process industry. Utilization of the advantages of digitalization can improve efficiency and the ability to stay competitive in increasing global competition in many conventional industrial processes. The design of industrial processes is often based on long term empirical and theoretical knowledge which has been incorporated into thoroughly built mathematical models.

These models often include iterative balance calculations to fulfill empirical and physical process constraints. These models are well suited for steady state process design and often used when offering, planning and constructing new process lines, however, they are often computationally too cumbersome for use in real-time solutions demanding short execution time.

Optimal utilization of processes should ideally include real-time operational optimization and scheduling where results can be presented to operators and/or process control quickly. Due to the time requirements and computational complexity of the optimization schemes, the underlying process models must be capable of producing results significantly faster than real-time. Thus, models requiring iterative calculations are typically too cumbersome to incorporate into the optimization. The high demand on execution time can often be compensated by lowering demands on model precision for the real-time operational optimization. Examples of demanding real-time optimization utilized in process design can be found in [1,2,3,4].

Good examples of thoroughly built steady state models can be found in metallurgy. Most metallurgical processes are old and have large societal impact which has allowed extensive development work to model process behavior over many decades. These processes comprise complex physical and chemical reactions and modelling has been both theoretical and empirical. To fulfill the basic requirement of mass and energy conservation and empirical observations iterative calculation is often employed. Lately also dynamic models purely based on fundamental physical laws have been successfully derived, e.g. [5] for the melting process in electric arc furnace.

These types of models are built on differential equations and needs careful parameter determination. They are normally solved with integration algorithms which, depending on model complexity, can be too slow for real-time operational optimization and scheduling.

The incentive for this study is the need for operational optimization of a copper smelting line. Optimal operation of a copper smelting line is challenging for the operators as the operation is divided into many complex individual sub processes. Plant wide optimization is required to maximize production and resource efficiency. Additionally, more challenging ores have to be used to retain economic competitiveness worldwide which increases the need for process optimization. Improved operation of copper smelting can provide improved utilization of different input materials and recyclants. Copper smelters present a challenging optimization problem where the harsh environment can prevent obtaining mineral and operational information, data is highly uncertain or measurements may be severely delayed. A full scale optimization of the complete process line will include a considerable amount of variables and require the consideration of large time horizons. Further, many of the underlying models are nonlinear. Thus, sub processes and the related models should be relatively lightweight in terms of their computational requirements. In principle, the development of optimization for a copper smelting line operation consists of modelling of unit processes and designing of optimization / scheduling for the combined unit process models.

Static input-output process models can be derived with use of mass and energy balances supplemented with sometimes uncertain process reaction knowledge completed with empirical knowledge. In principle, this empirical knowledge can be written as constraints in equation form. These equations can be completed with mass and energy balances to form a complete equation group determining process reactions. By solving the equation group, the unknown parameters and thereby the static process state can be solved under the given constraints. In practice this approach is challenging as the equations are often complex and manual solutions may be error prone and exceptionally time consuming.

Development of aids for this challenge started in the beginning of the 1970s under the scientific area of symbolic computation. Software programs for manual computation are called computer algebra systems (CAS) and are at present highly developed and even implemented in hand held calculators.

These systems include Mathematica [6] and Maple [7], the latter has been implemented in Matlab [8] as the Symbolic Math Toolbox. In later Matlab versions, the toolbox is based on the MuPAD symbolic engine originally developed at the University of Paderborn. Matlab offers a convenient way of shifting from symbolic calculus to numerical computation.

Utilization of symbolic computation for solving unknown variables of restricted mass balance equations seems to be a rare approach or rarely reported. A similar method was used in [9,10] in the same research group but the authors have not found similar work by others. Symbolic computation is, however, commonly utilized when forming first principle models [11,12,13]. Its use is especially convenient for model design with e.g. Lagrangian mechanics [14].

For optimization of the operation of the copper smelting line computationally lightweight models of all unit processes are required. This paper presents a method for converting an iterative balance model to a directly calculated model suitable for process operation optimization. The method is demonstrated with the first unit process in the copper smelting line, the flash smelting furnace (FSF).

1 Examples of Industrial Process Optiomization and Scheduling

Process optimization in general can be viewed as requiring predictive models capable of evaluating the evolution of the process under different process variables and operational schemes. In many cases linear models or finite response models are used to facilitate the fast calculation of predictions. Optimization determines the variables which minimize or maximize some objective function while fulfilling process constraints. The simplest objective is often the maximization of throughput. More advanced objectives may include considerations of energy use or different quality variables. When more exact predictions are required or linearization is not applicable for some other reason, nonlinear process models are used. Solutions will then require complicated optimization algorithms for the determination of optimal process variables and operation. In general, these algorithms require iterative calculation to find optimal values.

Scheduling problems determine which process units are used when and for which process tasks. Almost all scheduling problems consider batch processes and thus require integer variables introducing additional complexity to the optimization problem. Further complexity is introduced when the processes, such as copper production, include the combination of batch and continuous sub processes. Scheduling problem formulations can be roughly divided into discrete time or continuous time problems. In either implementation the number of variables is often in the hundreds and even multiple thousands of constraints are required. The most common method is to define the scheduling problem as a mixed integer linear program (MILP) which can typically be solved in seconds. In addition to production rates and task timings logistical concerns related to transfer of materials and maintenance are often incorporated into the formulation.

Scheduling has been in common use in many industrial applications, especially related to chemical processes, for decades. Some implementations of industrial scheduling and optimization include the scheduling of a pulp and paper machine reported in [15]. Here an optimal production schedule was defined and energy production and prices were considered in the objective. Steel production has often been considered in scheduling problems. One implementation was reported in [16] where the production of different product recipes was considered and the problem solved with decomposition. The optimization and scheduling of copper production has seldom been reported. One implementation was introduced in [17] where throughput is maximized while also enabling the consideration of different maintenance tasks. More recently, [18] reported an implementation of a greedy algorithm to plan the production of a copper plant. The authors introduced a nonlinear optimization of copper production in [19] where a simulation based approach was used.

Different optimization algorithms are available in many commercial products. MILP solvers are included in most computational software. In [17], the problem included 750 variables, of which 84 were binary integer variables, and 984 constraints with a solution time of under one second. MILP solvers roughly work by relaxing the integer constraints, find an optimal solution and if this does not fulfill the integer constraints perform a branch-and-bounding of the problem and find new optimal solutions for the new problems. Nonlinear solvers also require iterative search methods.

For example, Matlab includes the interior point and sequential quadratic programming algorithms for use with constrained nonlinear problems. In [19], an iterative simulation was used to predict the evolution of the process. Solutions were produced in about 60 seconds. Required iteration amounts are in the hundreds or thousands. This illustrates the need for lightweight models to enable real-time optimization.

2 Copper Production Line

Copper smelting plants convert the input materials, concentrates, which consist of mainly copper and iron sulphides, to almost pure copper through multiple oxidization stages. This begins from the mixing of a suitable concentrate mix with a copper content of 20-30 % which, after drying, is fed to the FSF. The mix reacts with the oxygen-enriched air feed and separates to matte (~60-65 % Cu) and slag. Silica flux is added to the FSF feed during operation to achieve suitable conditions for separation of matte and slag. The oxidization reactions generate heat though in some cases additional heating may be required.

Matte and slag are removed intermittently from the FSF, matte is moved to the converters, and slag is processed further in the slag treatment plant. After treatment, both FSF and converter slag can be recycled back to the FSF. Pierce-Smith converters (PSC) use a submerged feed of oxygen enriched air. Converters are operated in batches where first, in multiple slag-making stages, FSF matte is added between air blows. Here, most of the iron compounds will react and move to slag. Second, after removal of slag, in one longer copper-making stage the remaining sulphur is removed from copper compounds. Temperature is controlled with the addition of recycled material, e.g. scrap metal. The ensuing blister copper (~99 % Cu) is moved to anode furnaces where oxygen is removed from the blister copper and the copper is cast to anodes and finally transported to refinery for electrolytic purification to cathode copper. Figure 1 shows a full copper production line including both smelting and refining. A detailed description of the smelting process can be found for example in [20].

The FSF matte copper content can be viewed as one of the main decision variables in smelting as the higher copper content in matte (matte grade) is, the higher the copper content in the slag both in FSF and PSC and less blowing time in PSC.

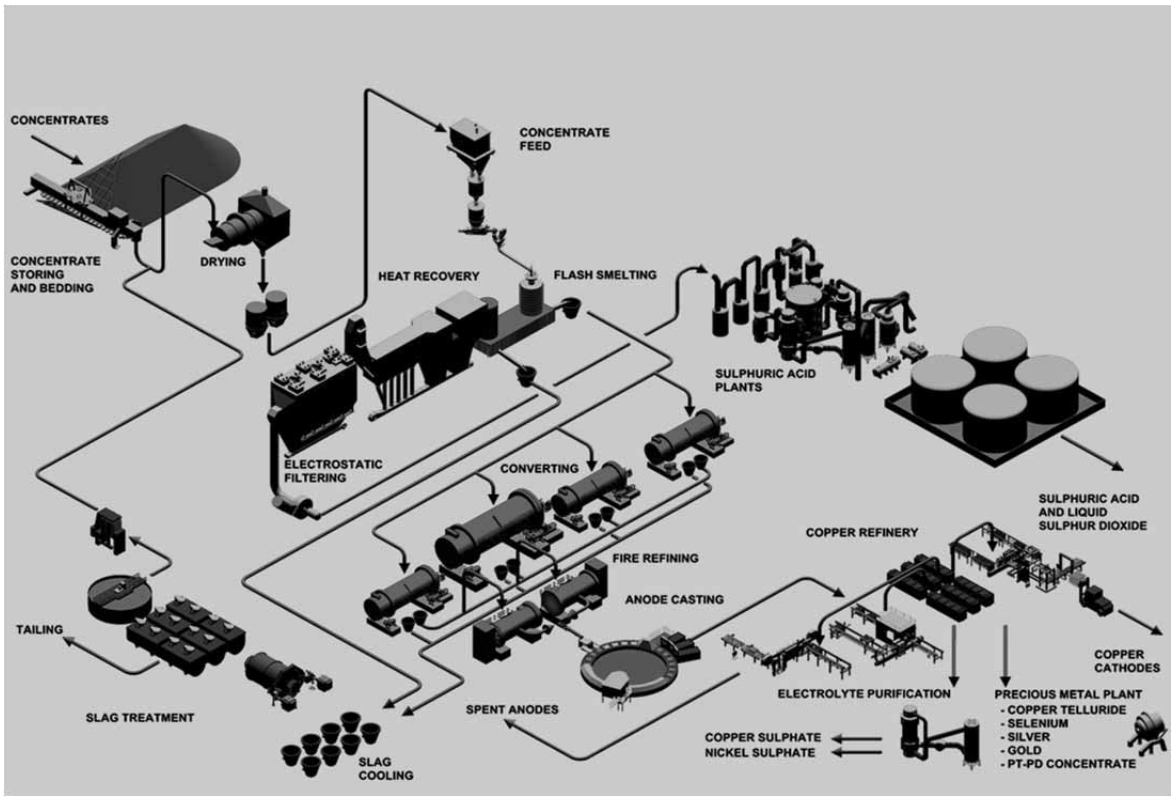


Figure 1: Flow sheet of copper process at Boliden Harjavalta [22].

Additionally, in FSF modelling matte grade is often used as a variable in distribution of other valuable metals, such as silver (Ag), cadmium (Cd), cobalt (Co), nickel (Ni), lead (Pb), tin (Sn), and zinc (Zn), to both matte and slag. Production bottlenecks include the production rate of FSF and the required converter tasks and availability of converters. Transportation of material from the FSF to converters is handled with cranes and may present limits for production rates. SO2 gases are produced in all production stages and the capacity of the gas treatment plant must be considered when determining the production rates and timings of different production tasks.

3 Model Conversion

The method for converting an iterative balance model to a directly calculating model is here demonstrated with a model of the flash smelting furnace, modelled in HSC-Sim [21]. HSC-Sim is a calculation module of HSC Chemistry software developed by Outotec. The name refers to the automatically utilized thermochemical database which contains enthalpy (H), entropy (S) and heat capacity (Cp) data for an extensive amount of chemical compounds.

The HSC-Sim module enables application of HSC Chemistry to a whole process made up of process units and streams. The HSC-Sim module consists of a graphical flowsheet and spreadsheet type process unit models. The custom-made variable list enables creation of different types of process models in chemistry, metallurgy, mineralogy, economics, etc. Each process unit is actually one Excel file. In the Distribution units the compounds are divided into elements and calculation is done with element distribution coefficients. Based on process knowledge some coefficients are defined as fixed. Coefficients for assisting elements in compound formation are calculated based on molar need and supply and called float. Surplus elements are divided with coefficients called rest. Units can be used together or separately and the calculations can be Excel- or DLL-based.

HSC Sim pyro models are mathematical process models based on mass- and energy balances and empirical knowledge controlling the equilibrium state. These models are successfully used in strategic planning of metal processing. The drawback of these models is the iterative calculation needed for reaching the equilibrium state. This iterative calculation is too slow for use in online process optimization of the whole smelter line.

3.1 Legacy model

The flash smelting furnace process has been modelled in HSC-Sim as a static division process with empirical knowledge controlling parts of the division coefficients. The implementation is a spreadsheet-like division calculation with iterative calculation to fulfill constraints derived from empirical and physical knowledge.

The model consists of three main spreadsheets; Input, Distributions and Output, each containing between 146 and 424 rows and 68 columns. The Input sheet is sparsely filled with element mass flows and describes how input compounds in different streams are broken up to elements according to chemical molar consistency. The Distributions sheet is sparsely filled with distribution coefficients dividing element mass flow into compounds for different output streams partly according to chemical reactions. The Output sheet is filled with corresponding element mass flows that build up the output compounds in different output streams. In addition to the three main spread sheets, a Controls sheet includes 27 empirical process observations that must be fulfilled in the stationary state.

In principle, the distribution from input compounds to output compound is built up around how the main elements copper (Cu) and iron (Fe) is distributed between compounds in the output streams. The chemical reactions require assisting element as oxygen (O) and silicon (Si) which are brought in as floating elements. Sulphur (S) is partly handled as a main element and partly as an assisting element. As a result, the model consists of some fixed distribution coefficients, many coefficients which are iteratively adjusted to fulfill the empirical observations and numerous coefficients calculated as float according to corresponding chemical reactions or as rest for surplus elements. The model is thus a system of four spread-sheets with a large number of interconnected cells. An iterative routine is used to solve the distribution coefficients and thereby the element and compound streams in the stationary state.

The calculation is very useful for off-line strategic planning of metal processing. The calculation is, however, too slow for real-time process optimization.

3.2 Method for derivation of fast calculating model

In general, the objective for the study was to find a method for converting iterative output controlled balance models to directly calculating models suitable for process scheduling. The basic idea was to form a symbolic equation group based on the flash smelting furnace HSC-Sim model and to solve this group analytically with symbolic computation to achieve causal outputs as direct functions of inputs. The solution is possible due to empirical knowledge included in the Controls sheet of the FSF HSC-Sim model.

Thus, the task was to write a fully parametrized equation group based on the FSF HSC-Sim model where the equations are based on the equations of empirical knowledge in the Controls sheet. The model is in this analytic approach simplified. The input elements include only the main elements; copper (Cu), iron (Fe), nitrogen (N), oxygen (O), sulphur (S), silicon (Si) and other content (Ot). The distribution of the elements between the output streams, which are settler gas, settler fume, settler dust, slag and matte, is fully in line with the FSF HSC-Sim model. The eight equations determining empirical knowledge regarding the main elements was chosen as base for the equations. To enable an analytic solution with the symbolic software the equation group has to be exactly determined.

The equation group formulation starts with defining all basic variables as symbolic variables. This example included 7 element mass flows, 23 distribution coefficients for element distribution to output streams and 41 distribution coefficients for element distribution into compounds in the different output streams. The main formulation work is to define the relationship between these variables with emphasis on the formulation of the float and rest variables. Here, this part required about 75 definitions. After these definitions, the output compounds can be formulated. Afterwards, the final equations based on the empirical knowledge in the Controls sheets can be written. To ease the derivation of the analytic solution of the software the nonlinearities in the empirical knowledge were linearized. The same variables as the manipulated variables in the iterative solution of HSC-Sim model were chosen as variables for the calculation to solve. They were; distribution coefficient for Fe to matte, distribution coefficient for Fe in slag to FeS, distribution coefficient for Cu to slag, distribution coefficient for Fe in matte to Fe3O4, Ot to matte, Si input stream, O input stream and distribution coefficient for Fe in slag to Fe3O4.

This study utilizes the Symbolic Math Toolbox in the Matlab software. With the relationships concerning use of oxygen still undefined, the solver managed to achieve a fully symbolical solution in around five minutes with a laptop.

When oxygen is taken into account, the solver has been forced to settle for a numeric approximation, which still includes all the variables in an appropriate manner. The length of the analytic solutions is over 25000 characters. The solutions are at this stage provided with the values of the fixed variables. The last task of the program is to produce usable functions of the long analytic solutions.

4 Model Validation and Discussion

Model validation is performed to ensure usability of the model in real-time process optimization and scheduling. As copper content in matte is a good measure of the process state, the validation is performed at varying matte copper percentage.

4.1 Similarity to legacy model

Figure 2 shows a comparison between the analytical direct solution results, with the blue line, and iteratively calculated HSC-Sim results, red line, as function of matte copper percentage.

The cause for the differences is the fact that the analytically solved model is a simplified model of the process including only the main elements. E.g. both silicon and oxygen is consumed by other minor compounds which are not included in the model. The difference is mainly a shift of magnitude which can easily be compensated by a term proportional to the total concentrate flow. With this compensation the analytically solved model is adequate for the on-line utilization.

4.2 Calculation time

As the optimization and scheduling algorithm calls the model hundreds of times per second the calculation time has to be short. A test function call from Matlab showed that the execution time is only some milliseconds for calls of two to eight variables, which is sufficient for the on-line utilization. The calculation time for the iterative solution of the HSC-Sim model is a few seconds.

5 Model Utilization

The directly calculating model of the flash smelting furnace process will be utilized in scheduling of a copper production line to optimize production and costs. When solving the equation group, the solvable variables can be freely chosen. There are two evident ways of model formulation that can be utilized.

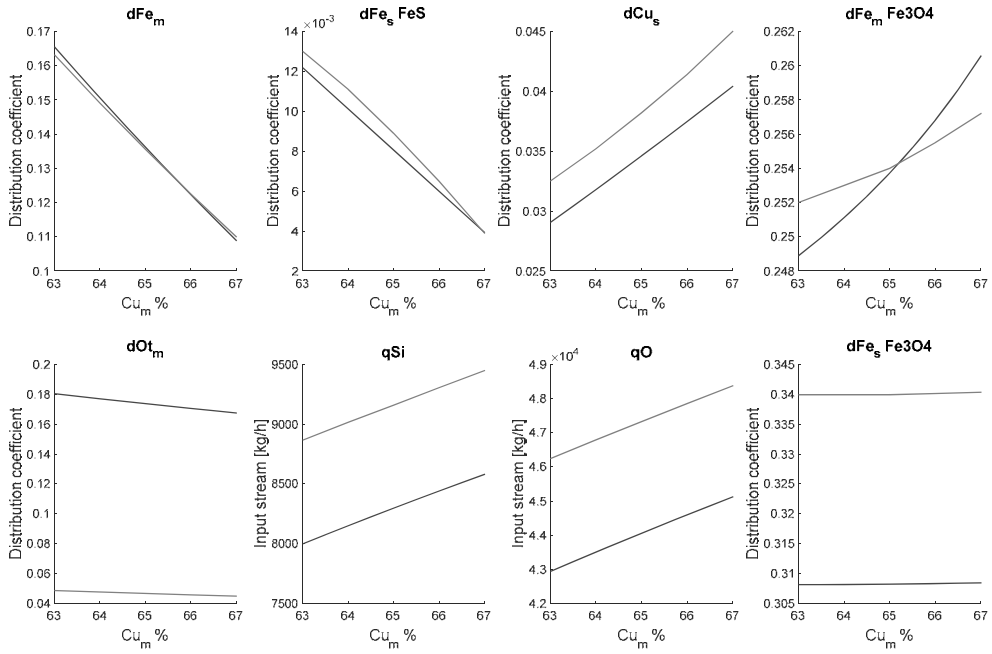


Figure 2: A comparison between analytical solution results with blue line and iteratively calculated HSC-Sim results with red line.

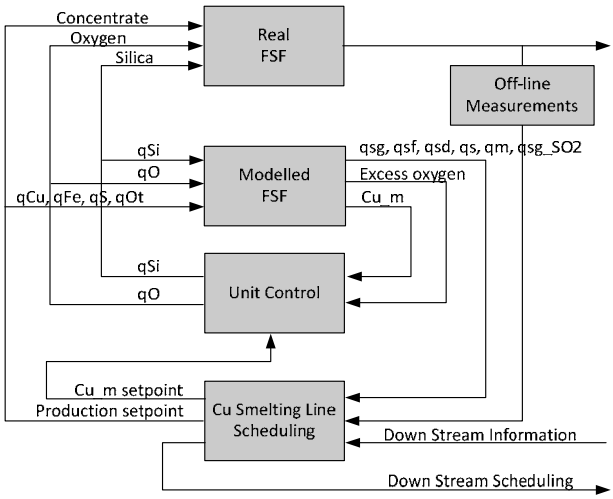


Figure 3: Direct input output model utilized in scheduling.

5.1 Direct input output

A natural solution would be to form a direct input output model to mimic the real smelting process. Figure 3 represents a scheduling structure that utilizes the input output model. As scheduling is a high level task whose interests are in production rate and oxidation level in first stage smelting, a lower level control structure has to deal with the unit control of the flash smelting furnace. This is shown as feedback control of the open loop model. In practice, this could be a sub optimization task for the scheduling routine.

5.2 Closed analytic solution

To enhance the direct scheduling interests, the required control variables can directly be chosen as solvable variables in the equation group. The static model allows us to utilize a closed analytic solution whose scheduling structure is clear and shown in Figure 4. This direct solution will not need the sub optimization. Feedback from the off-line measurements compensates for model inaccuracy.

5.3 Model based schedule calculation

To demonstrate the usage of the directly calculating model an example schedule is derived where special attention is paid to the calculation time.

A similar routine will be called at high frequency when the model is utilized in the real-time operational and scheduling optimization. The routine is called at the moment when the nonlinear optimization algorithm executes a new iterative schedule. The example is in line with the utilization of the closed analytic solution presented in Figure 4.

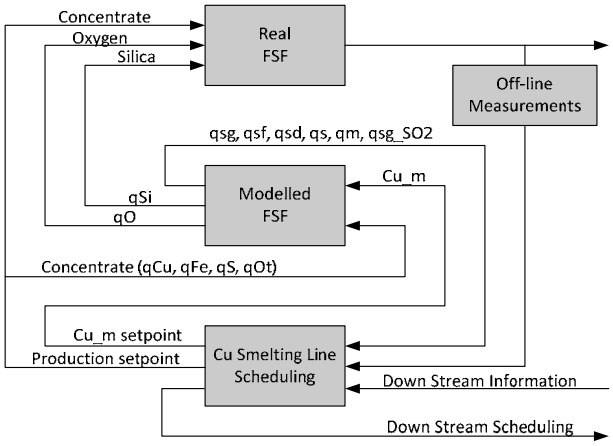


Figure 4: Closed analytic solution utilized in scheduling

The flash smelting furnace is here regarded as a static smelting process feeding parts of the formed compounds to matte. Matte volume in the bottom of the furnace is assumed to be fully mixed i.e. we have a static material distribution process followed by a fully mixed stock. The example comprises variable time moments which can be chosen by the optimization algorithm and where the algorithm can suggest changes in production rate and copper content of feed to matte. Additionally, the algorithm schedules tapping of matte for further delivery to Pierce-Smith converters. The example routine utilizes the closed analytic solution and simply track element flows to matte and respective stock situations starting from an initial state. As the stock is assumed to be fully mixed the copper percentage can be directly calculated as copper mass of total mass in storage. Figure 5 shows a plausible schedule of an optimization algorithm including two steps in concentrate feed rate, one step in copper content of feed to matte and one matte tapping.

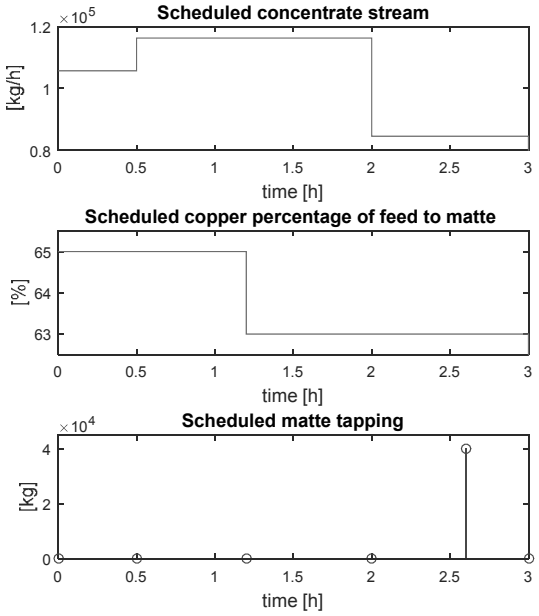


Figure 5: A plausible schedule of an optimization algorithm.

When changing the concentrate feed rate, the internal element mix has been kept constant. Figure 6 shows the outcome of the example routine. First the routine has directly calculated the required silica and oxygen for the different static smelting process steps according to the closed analytic solution. Second the routine has calculated the stepwise changing element streams to matte and kept track of the total mass in matte and the copper amount in matte to be able to track the copper percentage in matte. The copper matte percentage is exact in the figure at the time moments. In reality the change between the time moments in copper matte percentage is similar to a first order step response due to the integrator effect of the matte volume in the bottom of the furnace. Due to the immediate tapping the total matte mass is not exact in the period before the tapping moment. This inexactness in plotted figures is not a problem as the optimization algorithm only need the values at the algorithm chosen moments. The necessity of including the matte volume in the schedule calculation is revealed when comparing the copper content of feed to matte to the actual copper matte percentage in the matte volume i.e. the copper matte percentage of the tapped copper delivered to the next unit process in the smelter line.

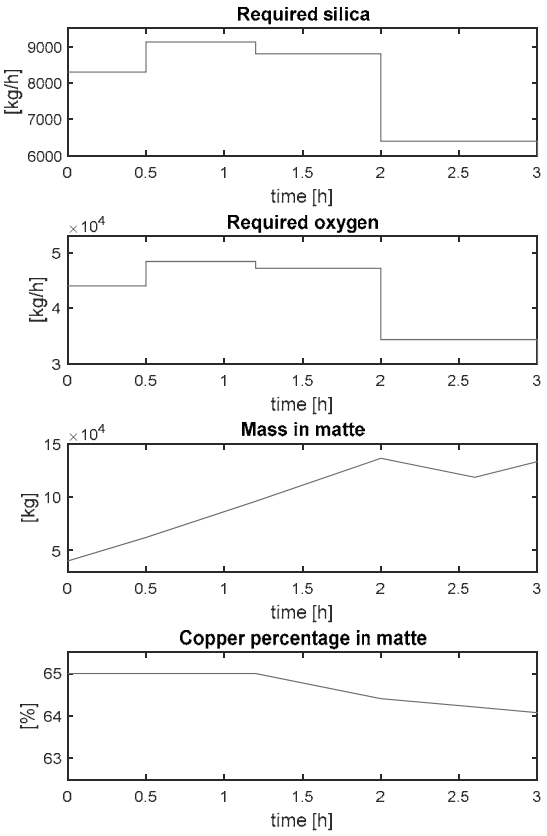


Figure 6: Outcome of example routine including both needed process feeds and matte state.

The calculation time for the example routine executed in Matlab is about 15 ms in a laptop computer. The routine included two function calls which prolonged the execution time but these function calls can be integrated to one when optimizing speed. The execution time is estimated to be short enough for real-time operational and scheduling optimization. FSF models with execution time in few seconds are earlier implemented successfully in controlling the FSF unit process [23], but the execution time demand changes significantly when the whole smelter line operation is to be optimized.

6 Conclusions

The objective of this study was to develop a method for converting iterative output controlled balance models to directly calculating models for process optimization and scheduling.

This method was used in the case of a flash smelting furnace, previously modelled in HSC-Sim. The fast calculating model is to be used in optimization of the total production line operation.

The method consisted of formulating an equation group based on the constrained FSF HSC-Sim model and solving the unknown parameters and static states with use of a symbolic calculation software. The study was successful even if it requires careful formulation work and the solution matched the solution of the original model. The equation group should be fully determined to enable a solution. The solution was implemented as a direct calculation function whose calculation time fulfilled the requirements for scheduling use.

The advantage with the approach is that even though the length of the generated functions disables model maintenance in function form, functions can easily be recalculated after updates in the HSC-Sim model are done. The modelling method has shown to be a powerful general way of converting complex iteratively solvable models to fast directly calculating models for utilization in real-time operational and scheduling optimization.

The presented demonstration model did not include an energy balance and thereby the amount of nitrogen (N) feed is kept constant even if the nitrogen feed is in practice the means to affect process temperature. The legacy model is built on the assumption that temperature is on normal level which enables a mass balance without temperature dependency. The energy balance will be included in future work.

This paper is an extended version of a paper presented at the 9th EUROSIM Congress on Modelling and Simulation [24].

Acknowledgment

This work was carried out in the SIMP research program coordinated by the Finnish Metals and Engineering Competence Cluster (FIMECC) Ltd. The support is gratefully acknowledged. The authors are also grateful for the process expert knowledge and good working possibilities provided by Petri Latostenmaa and Ville Naakka at Boliden Harjavalta.

References

[1] Harjunkoski I, Bauer R. Configurable Scheduling Solution using Flexible Heuristics, Proceedings of the 26th European Symposium on Computer Aided Process Engineering – ESCAPE 26 2016, pp. 2362-2366.

[2] Touretzky C, Harjunkoski I, Baldea M. A Framework for Integrated Scheduling and Control using Discrete-Time Dynamic Process Models, Proceedings of the 26th European Symposium on Computer Aided Process Engineering – ESCAPE 26 2016, pp. 601-606.

[3] Pelusi D. PID and intelligent controllers for optimal timing performances of industrial actuators, (2012) International Journal of Simulation: Systems, Science and Technology, 13 (2), pp. 65-71.

[4] Pelusi D. Improving settling and rise times of controllers via intelligent algorithms (2012) Proceedings - 2012 14th International Conference on Modelling and Simulation, UKSim 2012, art. no. 6205447, pp. 187-192.

[5] Logar V. Modelling and Simulation of the Electric Arc Furnace Processes. Simulation Notes Europe SNE 26(2), 2016, 91-98. DOI: 10.11128/sne.26.tn.10335.

[6] https://www.wolfram.com/mathematica/, 23.6.2016

[7] http://www.maplesoft.com/solutions/education/, 23.6.2016

[8] www.mathworks.com/, 23.6.2016

[9] Korpela T, Björkqvist T, Majanne Y, Lautala P. (2014). Online Monitoring of Flue Gas Emissions in Power Plants having Multiple Fuels. IFAC-PapersOnLine, 19(1), 1355-1360. DOI: 10.3182/20140824-6-ZA-1003.01913

[10] Korpela T, Suominen O, Majanne Y, Laukkanen V, Lautala P. 2016. Robust Data Reconciliation of Combustion Variables in Multi-Fuel Fired Industrial Boilers. Control Engineering Practice. Vol. 55. P. 101–115. DOI: 10.1016/j.conengprac.2016.07.002.

[11] Belkhir W, Ratier N, Nguyen DD, Yang B, Lenczner M, Zamkotsian F, Cirstea H. Towards an automatic tool for multi-scale model derivation illustrated with a micro-mirror array, 17th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing 2015, pp. 47-54. DOI 10.1109/SYNASC.2015.17

[12] Lin J, Chen C. "Computer-aided-symbolic dynamic modeling for Stewart-platform manipulator," Robotica, vol. 27, pp. 331-341, 2009.

[13] Yakhno V, Altunkaynak M. "A polynomial approach to determine the time-dependent electric and magnetic fields in anisotropic materials by symbolic computations," COMPEL - the International Journal for Computation and Mathematics in Electrical and Electronic Engineering, vol. 35, pp. 1179-1202, 2016.

[14] Moosavian SAA, Papadopoulos E, "Explicit dynamics of space free-flyers with multiple manipulators via SPACEMAPLE," Adv. Rob., vol. 18, pp. 223-244, 2004.

[15] Säynevirta S, Luotojärvi M. Integrated paper production and energy planning. PulPaper 2004 conferences – Energy and carbon management (2004), Helsinki, Finland

[16] Harjunkoski I, Grossmann IE. (2001). A decomposition approach for the scheduling of a steel plant production. Computers & Chemical Engineering, 25(11), 1647-1660.

[17] Harjunkoski I, Fahl M, Borchers HW. Scheduling and Optimization of a Copper Production Process. Logistic optimization of chemical production processes (2008).

[18] Navarra A. Automated scheduling and scientific management of copper smelters. Mineral Processing and Extractive Metallurgy Vol. 125 , Iss. 1,2016.

[19] Suominen O, Mörsky V, Ritala R, Vilkko M. Framework for optimization and scheduling of a copper production plant. Proceedings of the 26th European Symposium on Computer Aided Chemical Engineering ESCAPE 26. Volume 38, 2016, Pages 1243–1248, DOI 10.1016/B978-0-444-63428-3.50212-5.

[20] Schlesinger ME, King MJ, Sole KC, Davenport WG. (2011). Extractive metallurgy of copper. Elsevier.

[21] http://www.outotec.com/en/Products--services/HSC-Chemistry/Calculation-modules/Sim--process-simulation/, 23.6.2016

[22] http://www.boliden.com/Documents/Press/Publications/Place%20broschures/boliden-harjavalta-en.pdf, 23.6.2016

[23] Björklund P, Ranki T, Miettinen E. Recent experiences with implementing dynamic process control and monitoring in the flash smelting process, Copper 2013 Conference, Chile. Volume III, Book 1: Nickolas Themelis symposium on pyrometallurgy and process engineering, 2014, Pages 153-164.

[24] Björkqvist T, Suominen O, Vilkko M, Korpi M. From Iterative Balance Models to Directly Calculating Explicit Models for Real-time Process Optimization and Scheduling. 2016 9th EUROSIM Congress on Modelling and Simulation, 184-189. DOI 10.1109/EUROSIM.2016.9

Modelling of Indoor Lighting Conditions in Buildings for Control Design Purposes

Vito Logar[1*]

[1]Faculty of Electrical Engineering, University of Ljubljana, Tržaška 25, 1000 Ljubljana, Slovenia;
*vito.logar@fe.uni-lj.si

Simulation Notes Europe SNE 26(4), 2016, 267 - 276
DOI: 10.11128/sne.26.tn.10358
Received: November 25, 2016
Accepted: December 5, 2016 (Special Issue Review)

Abstract. Lighting conditions in buildings and efficient use of solar energy are a subject of considerate attention in order to provide sufficient living comfort and to reduce the energy use. For this reason numerous methods and techniques, practical and theoretical, have been developed. In this paper a theoretical approach to modelling of the indoor lighting conditions is proposed, based on fuzzy black-box modelling. The presented model is able to estimate indoor illuminance levels as its outputs, by using measured external conditions as its inputs. The model can be used to study the influence of both controllable and uncontrollable variables to the indoor lighting conditions, such as weather, time of the year, blinds position, electric lighting and others. Furthermore, using the above model studies on control design can be performed in order to obtain algorithms for maximal use of the solar energy and to minimize the energy consumption. The study has shown that a fuzzy illuminance model can estimate the indoor illuminance levels comparable to the measured data. Small error measures show that similar modelling approach can be used in order to integrate the proposed model into other environments and can further be used for simulations on indoor lighting comfort, control design or model-based control.

Introduction

Indoor lighting conditions and the efficient use of solar energy have become very important in recent decades, both in terms of overall living comfort [1, 2] and energy-efficient buildings [3, 4, 5, 6].

A summary of the research carried out to date in the area of buildings' energy efficiency, buildings' energy performance and buildings' processes modelling can be found in the excellent review paper by Foucquier et al. [7]. Furthermore, sufficient daylight conditions have been proven to have a beneficial effect on human health [8]. Numerous approaches to controlling indoor-illuminance conditions have been proposed, most of which attempt to either achieve constant indoor-illuminance levels, so as to provide sufficient living comfort, or to maximize the use of solar energy, while still providing acceptable lighting conditions [9, 10, 11, 12, 13]. Together with the modelling of light flux, indoor light intensities and surface illuminances, which usually represent the basis for control design techniques, have also been the subject of much attention. Furthermore, many methods exist that are able to provide approximate illuminance-level prediction in a certain position in a room, given its geometry, global orientation, the position of the sun, the surface characteristics and/or the weather conditions/measurements [14, 15, 16]. Moreover, a study performed by Lindelöf [17] proposes a fast daylight model, able to obtain indoor illuminances as a linear combination of the external global and diffuse radiations, validated using the Radiance model, which can be used as a replacement for the real system of embedded controllers. Similarly, available software tools, i.e., Radiance, Daysim, Skyvision ([18, 19, 20]) and many others, are also able to calculate more-or-less accurate illuminance levels for a given position in a room; however, significant knowledge of the modelled system (complete geometric and photometric characteristics of the room, inventory, windows, blinds, lights, etc.) and the software itself are needed, in order to ensure accurate results. A lot of the existing approaches rely on known mathematical daylighting concepts and thus try to describe the physical relations between the input and output variables.

If the measurements of the real environmental conditions are available, a black-box approach to the calculation of the indoor illuminance can be introduced as one of the modelling possibilities. Black-box models have proved to be a useful tool for the modelling of processes whose characteristics, relations and dynamics are not exactly known or are harder to model with conventional approaches.

The objective of this study is to propose a black-box approach to indoor-illuminance estimation by using a fuzzy inference model. The proposed methodology results in the development of a model that describes the relations between its inputs: horizontal unobstructed illuminance (external illuminance), global and diffuse solar radiations, the positions of the blinds and the status of the lights; and its output: the estimated indoor illuminance. The method is, from the input/output point of view, similar to some existing methods, i.e., Lindelöf [17]; however, the methodological approach between the proposed and the existing methods is entirely different.

One of the main advantages of the proposed method is the simple design and parameterisation of the model, which does not require any knowledge about the modelled system, since the model's parameters, which define the input/output relations, inherit the room's characteristics, implicitly defined in the obtained measurements. Meaning that the room's characteristics, such as: geometry, indoor surfaces' reflectances, blinds' reflectances, quantity, sizes and positions of the windows, lights and furniture; and also the position of the indoor illuminance sensor, reflect in the measured indoor illuminance. Moreover, a change in either the room characteristics or the position of the sensor, if sufficiently large, also affects the measured value. After the model is parameterized (trained) and validated using the particular input/output measurements of interest, simulated or otherwise acquired input data can be used, replacing the actual measurements. The inputs defining the blinds' positions and the lights' status either need to be predetermined, adjusted manually or by means of the controller. Since the method uses measurements instead of physical characteristics in order to define the relations of the model, programming skills and the effort to manually design the room interior are not needed.

The proposed structure of the fuzzy models is very simple (5 inputs, 1 output, 3 Gaussian membership functions per input and 3 fuzzy rules) and the fuzzification/defuzzification procedures are simple vector multiplications.

The simplicity of the model is reflected in the fact that it is a fast model, with short evaluation times, which facilitates its inclusion in other applications or control algorithms. Finally, even though the model's structure is simple, the validation results have very accurate estimations in comparison with the measurements.

Although the proposed methodology has advantages, the fuzzy approach also has some drawbacks, which need to be considered when adopting the concept. In contrast to methods based on the physical modelling of the daylighting processes and pre-programmed algorithms, which normally require only the input part of the data in order to obtain the output, the fuzzy approach requires both the input and the output part of the data in order to parameterize the model using an automated training procedure. After the model is parameterized, only the input data is required. Moreover, since the model is based on measured data, which defines the room's characteristics, the calculation of the indoor illuminance under different conditions (e.g., different geometry, reflectances, sensor positions, etc.) needs a re-training of the model with new measurements. From this point of view, other tools like Radiance outperform the proposed method, since they are able to calculate a more-or-less accurate indoor illuminance for an arbitrarily positioned surface [23, 24].

The purpose of the study is not to propose a specific model that would represent a general solution for all possible situations (like Radiance, for instance), but to propose a simple methodology for how to obtain a model for a particular environment, where the model is characterized as fast, accurate and easy to obtain, without excessive knowledge of the particular problem. Furthermore, since the in-depth studies on, e.g., building automation, control design, energy conservation, living comfort, etc., are practically impossible to perform on real systems, due to varying weather conditions and poor repeatability, the use of a relatively simple illuminance model of sufficient accuracy in combination with the simulation procedures allows fast and repeatable testing of the designed algorithms or the model-based control of real processes.

1 Indoor Environment

The following section gives a description of the indoor environment, whose measurements are used as a basis for the fuzzy black-box model's development and the parameterisation.

The indoor environment consists of a room with dimensions of 7.49 m x 4.93 m x 3.88 m (l x w x h), a floor area of 38.80 m^2 and a volume of 163.40 m^3, with one outside wall that has a window, facing south-west (rotated approximately 30° counter clockwise from the east-west direction), where the outside wall is the longest wall. The room is located on the 4th of 5 floors in a building with no external obstacles that would obscure the light flow through the window (lat: 46.045737, lon: 14.494851). The area of the window is 11.4 m^2, with installed venetian blinds. The transmission of visible light through the window is 80 %. The room characteristics in terms of the photometry are the following: grey floor (35 % reflectance), white ceiling (80 % reflectance), white walls and beige furniture (average 65 % reflectance). Figure 1 shows the floor plan of the particular room, with the marked positions of the sensors (indoor and external illuminance, global and diffuse solar radiation - placed on the roof of the building, blinds' position), the window and the blinds.

The studied indoor environment is equipped with an automation, supervisory control and data-acquisition system (SCADA), which is composed of three distinct parts: the sensor array, the process and supervision level, and the data-acquisition level. The system measures the necessary values, such as the global and diffuse solar radiation, the external illuminance, the position of the blinds (and other values not relevant to this study) and controls the indoor-illuminance levels (and other values not relevant to this study) using the motorized venetian blinds and the electric lighting. The sensor for external illuminance (Thermokon LI65 outdoor light sensor) is mounted vertically on the facade beside the window and is capable of measuring the illuminance in the range from 0 to 20,000 lux. The sensors for the global and diffuse solar radiation (Kipp & Zonen CM7B pyranometer and albedometer) are mounted 2 floors higher, horizontally on the roof of the building and are measuring the solar radiation in the range 305 to 1,800 nm from 0 to 1,400 W/m^2. The indoor illuminance sensor (Thermokon LI04) is placed horizontally on the workbench/desk (at a height of approximately 0.9 m) and is capable of measuring the illuminance in the range from 0 to 2,000 lux. A complete description of the automation system and the applied control algorithms can be obtained from the paper by Košir et al. [13].

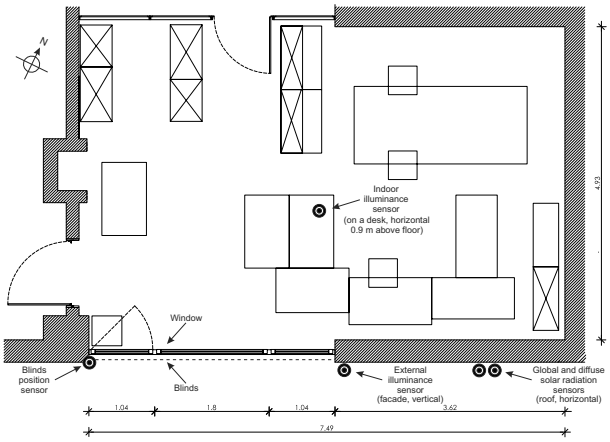


Figure 1: Floor plan of the modelled room, with the marked positions of the sensors (indoor and external illuminance, global and diffuse solar radiation (placed on the roof of the building), blinds' position), the window area and the blinds.

The global and diffuse solar radiation are measured in W/m^2, the external illuminance in lux, while the blinds' position can take values between 1 and 5 (1 - 0° slat angle (vertical), 2 - 30° slat angle, 3 - 60° slat angle, 4 - 90° slat angle (horizontal) or 5 - blinds completely retracted) and the lights' status can be either 0 (OFF) or 1 (ON).

2 Fuzzy Model

The concept of the black-box theory relies on understanding something entirely in terms of its function, without knowing the background or the mechanisms that enable this functionality. From this point of view, machine-learning techniques, among which are also fuzzy-inference systems, as one of the black-box approaches, can be considered as a mechanism of this black box for the input-to-output mapping of the data space. Meaning, if an appropriate structure of the system is designed, an arbitrary nonlinear function between the system's inputs and outputs can be described by the fuzzy mechanism. Since the black-box approach has no physical background to the particular process, the function describing the input-to-output space can be as close to the real physical relation as the learning data can describe it. Such a system can, therefore, be definitely valid only in the vicinity of the mapped data space.

This could be considered as one of the drawbacks of the black-box approach, as if the learning-data input-output relation is missing, the later incidence of such an input could lead, but not necessarily, to incorrect output estimations and should be experimentally validated. However, if sufficient data is available, the black-box approach has certain advantages over conventional modelling approaches. As mentioned before, when the structure of the black box is specified, its relations are determined only according to the input and output data, which eases the modelling for those who are, or are not, familiar with the process. Physical relations, process characteristics and other properties that could be hard to describe mathematically are replaced by the proper selection of the fuzzy structure (which can be obtained experimentally) and only limited knowledge (usually the order of the process) of the physical process is needed to build the process model. In this manner, the proper structure and parameterisation (pre-programmed automated learning procedures) of the fuzzy mechanism ensures that the model's parameters, which define the input/output relations, inherit the room's characteristics, implicitly defined in the obtained measurements.

2.1 Structure of the model

The proposed black-box illuminance model is based on a Takagi-Sugeno (TS) fuzzy-inference system (FIS) [25] that was trained using real environment measurements as the inputs and outputs. The model, in TS form, approximates a nonlinear system by smoothly interpolating affine local models [25]. Each local model contributes to the global model in a fuzzy subset of the space characterized by a membership function. The affine TS model can be used to approximate any arbitrary function with any desired degree of accuracy [26, 27, 28]. The generality can be proven with the Stone-Weierstrass theorem [29], which suggests that any continuous function can be approximated by a fuzzy-basis function expansion [30].

Since the light-flow dynamics can be, from the room-illuminance point of view, considered as infinitely fast or instantaneous, the black-box model is regarded as a static model from the modelling perspective. This means that a change in either the global or the diffuse light, the blinds' position or the lights' status has an immediate effect on the change in the observed illuminance, with no transitional dynamics, meaning that the value of the current illuminance is completely independent of the previous illuminance sample.

The fuzzy model uses the external illuminance multiplied by the position of the blinds, the global and diffuse solar radiation, the lights' status and the blinds' position as the inputs and the indoor illuminance as the output. The reason for using the multiplied external illuminance and blinds position as the input is that better estimation results can be achieved, in comparison to the results with only external illuminance as the input, which is a consequence of a prior virtual increase in the external illumination (multiplied by a factor from 1 to 5), dependent on the position of the blinds. The multiplication, therefore, implies that more light is entering the office when the position of the blinds is higher (higher slat angle) or if the blinds are completely retracted.

This implication allows better estimations of the indoor illuminance by the fuzzy model, since the particular input already partially describes the relation between the external illuminance and the blinds' position. However, the relation between the external illuminance, the blinds position and the indoor illuminance is not linear, as implied by the multiplication, but nonlinear, which is later defined by the corresponding fuzzy membership functions and the antecedent rules. The use of the global and diffuse radiation as two of the inputs (besides the external illuminance, the blinds' position and the lights' status) is optional and leads to better estimation results, in comparison with the results that have the global and diffuse radiation measurements omitted.

The reason for using the blinds' position as one of the inputs, despite the fact that it has already been used in the multiplication with external illuminance, is that even better estimation results can be achieved in comparison with the results that have this input omitted, which could be a consequence of the nonlinearity of the process caused by the blinds' reflectance, which is thus represented more accurately. Figure 2 shows a schematic representation of the model.

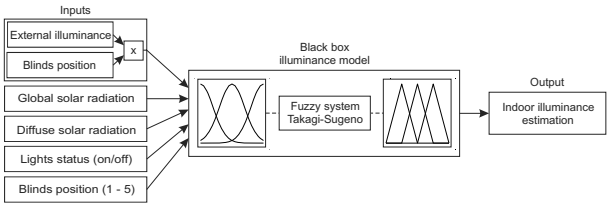


Figure 2: Schematic representation of the fuzzy black-box illuminance model.

2.2 Illuminance reconstruction

This section explains an optional method for simple external illuminance reconstruction, which is necessary when the illuminance sensor has a limit (in this case 20,000 lux). If sensors with higher limits (100,000 lux or higher) are used to obtain the measurements, the external illuminance reconstruction is not needed and the measured external illuminance can be used as an input to the fuzzy model directly.

Knowing that during sunny summer days the external illuminance can go above 100,000 lux, the missing (saturated) data needs to be reconstructed in order to achieve better estimations of the calculated value. As the fuzzy model also uses the global and diffuse solar radiations (among others) as its inputs, it is able to sufficiently well predict the indoor illuminance levels, even when the external illuminance goes above the sensor's limit (20,000 lux) and its value is not reconstructed; however, estimations during peak daylight are less accurate. The reconstructed signal is far from being equal to the actual illuminance levels; however, according to the validation results presented, an approximate estimation of the external illuminance suffices. Since the daily illuminance trend follows approximately the same shape, it can be approximately reconstructed according to the total saturation time of the external illuminance sensor, as given in the equation 1:

$$t_{sat} = t_{end} - t_{start} \tag{1}$$

$$q_{reconstruct}(t_{start}{:}\,t_{end}) = 1.5 \times 10^4 \sqrt{\frac{(-(linspace(-t_{sat}, t_{sat}))^2 + {t_{sat}}^2)(linspace(-t_{sat}, t_{sat}) + t_{sat})}{2t_{sat}}}$$

$$q_{ill} = q_{ill} + q_{reconstructed}$$

where t_{start} and t_{end} represent the beginning and the end of the sensor saturation in hours, t_{sat} represents the overall length of the saturation in hours, $q_{reconstruct}$ represents the reconstructed illuminance signal to be added to the measured illuminance signal in lux, q_{ill} represents the external illuminance signal in lux and the command *linspace(a,b)* represents a vector of linearly spaced values between *a* and *b*.

Figure 3 shows the comparison between the measured and the reconstructed external illuminance, as obtained by equation 1 for a part of the training data (5 days in early August), where the sensor reaches its upper limit.

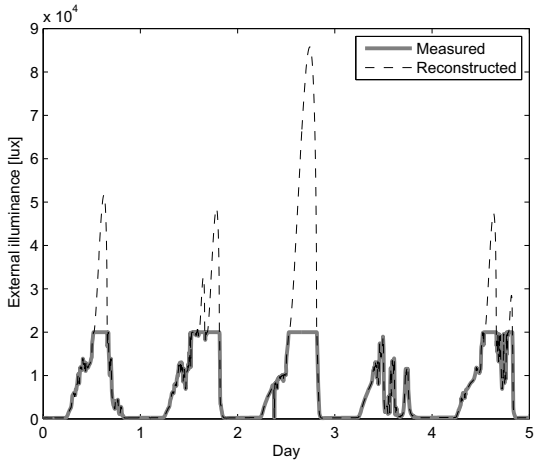


Figure 3: Comparison between the measured and the reconstructed external illuminance for 5 days in early August.

As can be seen in figure 3 the measured external illuminance is saturated at 20,000 lux for several hours around midday, while the reconstructed illuminance reaches up to almost 90,000 lux, which seems reasonable for sunny summer days. Equation 1 and the corresponding results are obtained experimentally, without any physical background and are not validated nor taken from any other source, but merely a rough estimation of what the external illuminance should be like. If the illuminance sensor has a limit that is high enough, such an estimation is not needed.

2.3 Parameterisation of the model

The parameterisation of the fuzzy model, also known as the training, was carried out for 1 month of different measurements with a sample time of 15 s. The data was chosen in a manner that covered all four seasons and as many real-world situations as possible, i.e., sunny, cloudy, foggy weather, different sun azimuths and elevations, quick illuminance changes due to partial cloudiness or incoming thunderstorms, operation of lights, shading, etc. Shorter or longer periods of measurement data can be used to train the model; however, using less training data results in a less versatile and robust model, which is sufficiently accurate only for similar conditions. On the other hand, using more training data results in a more versatile and robust model; however, extended data only increases the model's performance by a smaller amount. The fuzzy model uses three data clusters for each input and the Gaussian membership functions.

An adaptive, neuro-fuzzy, hybrid learning algorithm (ANFIS), as implemented in the Matlab environment, was used for the model training.

Figure 4 shows a part (5 days in early August) of the input data (otherwise 30 days) that was used for the model training, including the external illuminance, the global and diffuse solar radiation, the blinds' position and the lights' status.

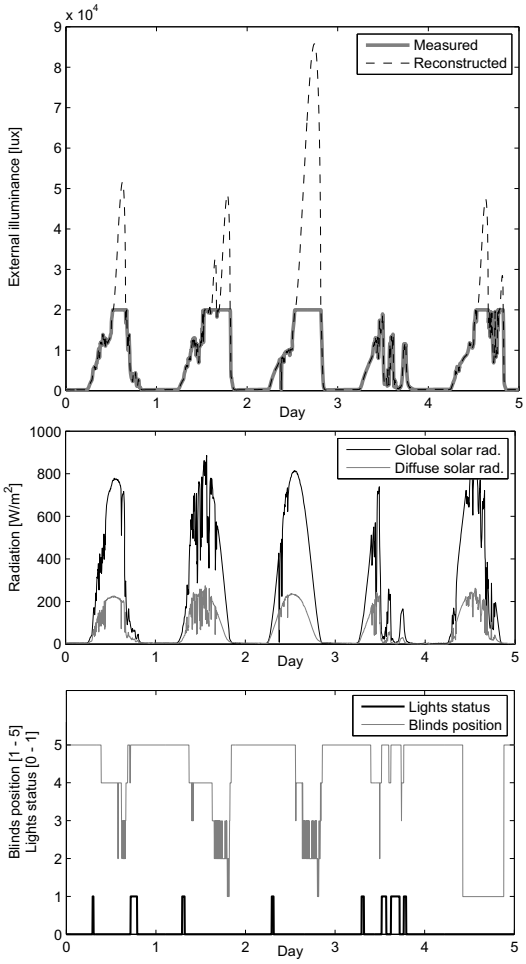


Figure 4: A part (5 days in early August) of the input data for the model training, including the reconstructed external illuminance (upper panel), the global and diffuse solar radiation (middle panel), the lights' status and the blinds' position (lower panel).

As is clear from figure 4, for the selected days, the global and diffuse radiation take values between 0 and 900 W/m^2, the external illuminance between 0 and more than 20,000 lux (reconstructed above 20,000 lux as shown in figure 3), the blinds' position can be in the range from 1 to 5 and the lights' status can be either 0 or 1, as already described.

3 Results

Using real measurements, the model was trained using an ANFIS training algorithm. Figure 5 shows the shape and distribution of the trained-model membership functions for each input. The membership functions, which characterize the input-output space, define the effect of a given input value on the computed output value. Since the presented model has five different inputs and three data clusters, each input has three membership functions with different centres and kurtosis.

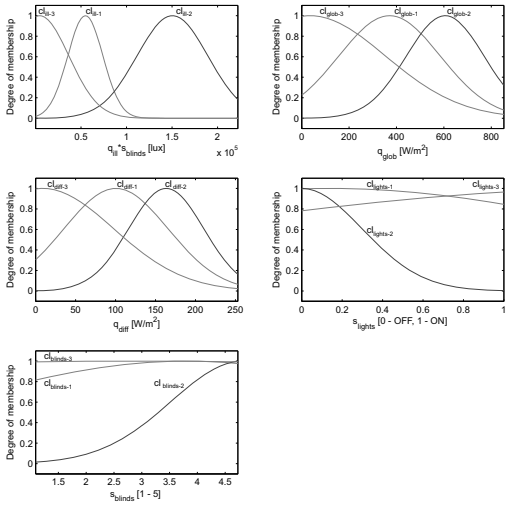


Figure 5: Shape and distribution of the fuzzy membership functions on all five inputs for all three data clusters as obtained by the ANFIS training algorithm.

Figure 5 shows the shape and the distribution of the membership functions, which contribute to the value of the output, i.e., the indoor illuminance level, which is obtained using the following equations:

$$out_1 = [q_{ill}s_{blinds}cl_{ill-1}, q_{glob}cl_{glob-1}, q_{dif}cl_{dif-1}, s_{lights}cl_{lights-1}, s_{blinds}cl_{blinds-1}] \times [0.0002238, -0.2234, 2.091, 453.3, 182.4]^T - 373.1 \quad (2)$$

$$out_2 = [q_{ill}s_{blinds}cl_{ill-2}, q_{glob}cl_{glob-2}, q_{dif}cl_{dif-2}, s_{lights}cl_{lights-2}, s_{blinds}cl_{blinds-2}] \times [0.002637, 0.3512, 1.361, -99.01, -16.13]^T - 168.1 \quad (3)$$

$$out_3 = [q_{ill}s_{blinds}cl_{ill-3}, q_{glob}cl_{glob-3}, q_{dif}cl_{dif-3}, s_{lights}cl_{lights-3}, s_{blinds}cl_{blind-3}] \times [0.01307, -0.8077, 1.146, 128.5, -1.442]^T + 19.17 \quad (4)$$

where q_{ill} represents the external illuminance, s_{blinds} represents the blinds' position, q_{glob} and q_{dif} represents the global and diffuse solar radiation, s_{lights} represents the lights' status and $cl_{x\text{-}n}$ (x - correspondent input, n - cluster number) represents the degree of membership (from 0 to 1) of the given input to the corresponding cluster. Finally, the indoor illuminance q_{in} is obtained using the following equation:

$$q_{in} = q_{out1} + q_{out2} + q_{out3} \quad (5)$$

The membership functions shown in figure 5 and the fuzzy rules described by equations 2 to 5 define the so-called mapping of the input-output data space. Generally, each input contributes to the combined output according to the corresponding membership functions, its centre position and kurtosis and the equations 2 to 5.

Figure 6 shows the validation of the model by comparing the measured and model-estimated indoor illuminances for 12 different days (days 1-3: June, days 4-6: September, days 7-9: December, days 10-12: March), covering different solar positions and climatic conditions. The model was validated using the data from different days than were used for the model training.

As can be seen in the first graph of figure 6 the indoor illuminance levels obtained with the fuzzy model in general correspond to the measured illuminance levels. The indoor illuminance measurements during the periods of daylight are approximately limited to a range from 400 to 1,600 lux and are a consequence of the external illuminance conditions, the position of the blinds and the status of the lights. Lower indoor-illuminance values occur during the morning and early afternoon, while these values are higher later in the day, which is a direct consequence of the room's orientation (direct sunlight to the window area appears in the late afternoon, around 6pm). The illuminances at night, both measured and estimated, are around 10 lux, and not close to 0 lux as expected, which could be a consequence of the skyglow (light pollution) and the neighbouring street lighting.

The second graph of figure 6 shows that external illumination measurements reach over 20,000 lux (near 80,000 lux estimated) in the first 6 validation days' data (June and September), which is reasonable for this period of the year. However, shorter intervals of sensor saturation appear in days 4 to 6 (September), due to the shorter intervals of daylight, which is also reflected in lower estimated peak illuminances (up to 40,000 lux).

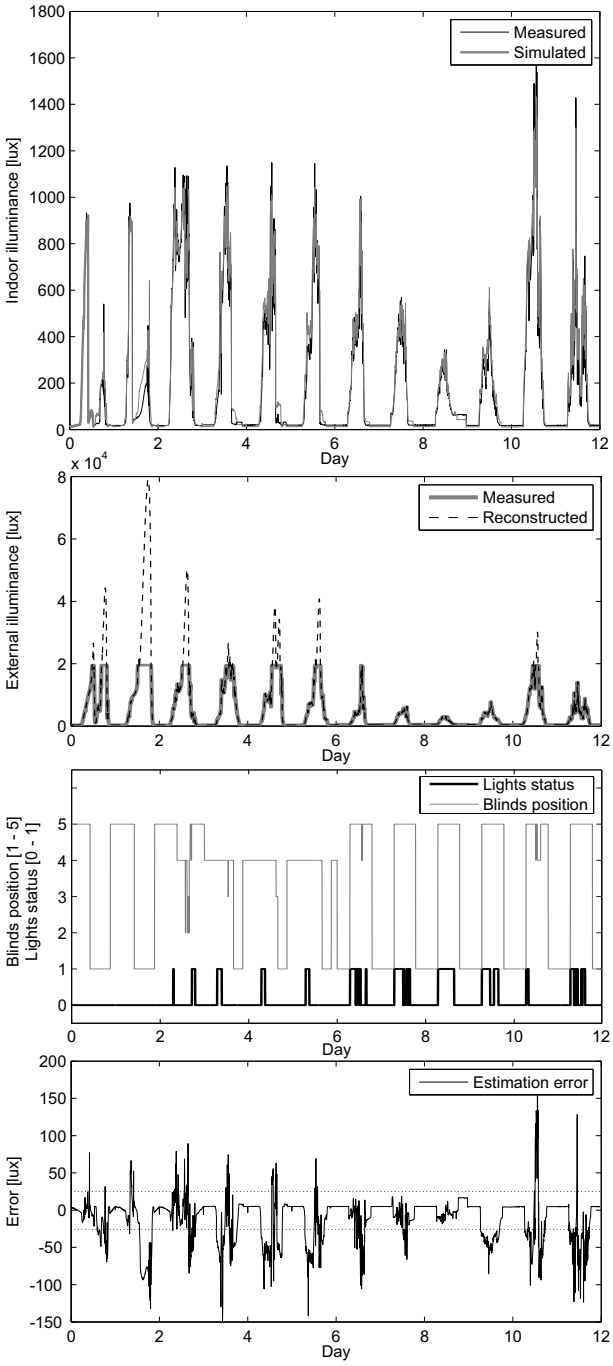


Figure 6: Illuminance model validation; first graph shows the comparison of the measured (black line) and model simulated (grey line) indoor illuminances for 12 different days (days 1-3: June, days 4-6: September, days 7-9: December, days 10-12: March); second graph shows the measured external illuminances with reconstruction; third graph shows the position of the blinds and the lights' status, respectively; fourth graph shows the illuminance estimation error (dashed vertical lines show the mean absolute error interval).

Furthermore, individual occurrences of illuminance below 20,000 lux suggest partial cloudiness or incoming thunderstorms (i.e., day 1). The last 6 validation days' data (December and March) mainly exhibit illuminances below 20,000 lux, typical for the winter and spring seasons. Days 6, 11 and 12 suggest sunny weather with partial cloudiness (day 12), while days 8 to 10 suggest a full overcast or all-day fog, which is typical for this time of the year.

The fourth graph of figure 6 shows the error between the measured and the model-predicted illuminances. Individual error spikes between the measured and the simulated data, visible at days 1-6, 11 and 12, reaching up to approximately 150 lux can be observed. The spikes mostly occur at peak daylight, when the external illuminance is the highest, which implies that significant deviations between the actual and the reconstructed illuminances occur. Namely, the reconstructed illuminance is a smooth curve, neglecting the occurrence of individual clouds, which rapidly decrease the measured illuminances and cause fast illuminance fluctuations.

The problem is not a flaw of the presented method, but occurs solely due to the external illuminance sensor limit and its reconstruction and can easily be avoided using an illuminance sensor with a limit above 100,000 lux, which allows proper measurements of the illuminances above 20,000 lux. On average, the error between the measured and the estimated illuminances (mean absolute error - MAE) is approximately 25 lux (dashed vertical lines), which can be considered as a good model approximation of the real data. Furthermore, the root-mean-square error (RMSE) and the mean-bias error (MBE), as two of the measures of estimation reliability, are 12.60 % and 7.76 %, respectively, and suggest the satisfactory reliability of the fuzzy estimator (RMSE lower than 35 % and MBE lower than 15 %) [23].

Based on the presented comparison between the measured and the estimated indoor illuminances and the error measures (MAE, RMSE and MBE), the validation of the model can be considered as successful. The study has shown that the presented fuzzy black-box approach can be used as a satisfactory approximator for the indoor illuminance conditions in buildings.

4 Conclusion

The paper presents a fuzzy black-box modelling approach to estimating the illuminance of an indoor environment. The model is used to estimate the illuminance level at a certain point on a surface, given the required input values, and is easy to design and implement. The method uses real environment measurements, such as external illuminance, global and diffuse solar radiations, blinds' position, lights' status and indoor illuminance, in order to define the relations and obtain the parameters of the model.

After the model is parameterized using the particular input/output measurements of interest, simulated or otherwise acquired input data can be used, replacing the actual measurements if desired. The study has shown that a relatively simple fuzzy structure can be used in order to estimate the indoor illuminance at a particular spot. Since the fuzzy approach is based on simple mathematics and the model's structure only incorporates three membership functions per input and three fuzzy rules, the estimation of the indoor illuminance is very fast, allowing the use of the model in broader online or offline environments. Furthermore, low estimation errors (MAE = 25 lux, RMSE = 12.60 %, MBE = 7.76 %) suggest a satisfactory reliability for the proposed estimator.

The fuzzy black-box approach has its advantages and flaws, when compared to other similar methods, which need to be considered when deciding for the appropriate illuminance-estimation method. Briefly, the advantages of such an approach are the ease of model design and parameterisation, an accurate estimation, the mathematical simplicity, a quick calculation and the fact that only limited knowledge of the process is needed. Since the fuzzy model is parameterized from the measured data, the model's structure can also be used at different latitudes, different building orientations, window sizes, indoor photometric and geometric characteristics, etc., but only if the data is available to properly parameterize the model.

This claim has not been experimentally validated; however, following the essence of the black-box approach, i.e., to define a relation between the model's inputs and outputs without extensive knowledge about the background to the problem, but only possessing the required input/output data, the claim is reasonable.

While the proposed approach has certain advantages over other methods, it also has some general flaws, which mainly originate from the need for measurements. A fair number of measurements, in as many real situations as possible, are needed in order to obtain a robust and versatile model that covers a broad range of climatic conditions (external illuminance, solar radiation) and other influential variables (blinds, lights). Moreover, the proposed approach is able to estimate the illuminances for only one position; thus, if the illuminance levels of different positions in a room are required, measurements from multiple illuminance sensors should be obtained and used to train a multiple-output fuzzy model.

During the study it was discovered that, depending on the desired model versatility for different conditions, measurements for at least a few days (3-5) are needed in order to parameterize the model to satisfactorily estimate the illuminance for similar inputs. If estimations for a broader range of inputs are needed, at least 15 days of data are required. In order to obtain the results as shown in this paper, 30 days of data for different climatic conditions should be used. The study revealed, that the model performs surprisingly well even if the training data is obtained only during the summer (due to higher external illuminances and solar radiations) and the model is used for estimations in the other three seasons, i.e., training data: 15 consecutive days in July, validation data: 10 consecutive days in January, yields: MAE = 35 lux, RMSE = 18.1 %, MBE = 12.2 %. Using a method in this manner, a relatively small amount of model-training data is needed, which can be obtained quickly in exchange for a slightly lower accuracy of the model.

The black-box fuzzy model for indoor illuminance estimation, as presented here, is not primarily intended to be used as a stand-alone application, although it can be, but to be included in broader test environments, useful for a variety of studies, for instance: building automation, living comfort, energy conservation, control design, etc. Since such studies are practically impossible to perform on real systems, due to varying weather conditions and almost no repeatability, a combination of different dynamic models (thermal, visual and air quality) and simulation procedures is the most frequently used approach.

Therefore, the use of a simple illuminance model of sufficient accuracy allows quick and repeatable testing of the particular methods. Since the model is reliable and allows for a quick estimation of the illuminance, it can also be used in a similar way to the model presented by Lindelöf [17], i.e., as a model-based control approach using embedded controllers.

References

[1] Kim JT, Kim G. Overview and new developments in optical daylighting systems for building a healthy indoor environment, Building and Environment 45 (2010) 256-269.

[2] Huang L, Zhu Y, Ouyang Q, Cao B. A study on the effects of thermal, luminous, and acoustic environments on indoor environmental comfort in offices, Building and Environment 49 (2012) 304-309.

[3] da Fonseca RW, Didon\'{e} EL, Pereira FOR, Using artificial neural networks to predict the impact of daylighting on building final electric energy requirements, Energy and Buildings 61 (2013) 31-38.

[4] da Silva PC, Leal V, Andersen M. Influence of shading control patterns on the energy assessment of office spaces, Energy and Buildings 50 (2012) 35-48.

[5] Li DHW. A review of daylight illuminance determinations and energy implications, Applied Energy 87 (2010) 2109-2118.

[6] Tagliabue LC, Buzzetti M, Arosio B. Energy saving through the sun: Analysis of visual comfort and energy consumption in office space, Energy Procedia 30 (2012) 693-703.

[7] Foucquier A, Robert S, Suard F, Stephan L, Jay A. State of the art in building modelling and energy performances prediction: A review, Renewable and Sustainable Energy Reviews 23 (2013) 272-288.

[8] Košir M, Krainer A, Dovjak M, Kristl Ž. Automatically controlled daylighting for visual and nonvisual effects, Lighting Research and Technology 43 (2001) 439-455.

[9] Trobec Lah M, Zupančič B, Peternelj J, Krainer A. Daylight illuminance control with fuzzy logic, Solar Energy 80 (2006) 307-321.

[10] Kristl Ž, Košir M, Trobec Lah M, Krainer A. Fuzzy control system for thermal and visual comfort in building, Renewable Energy 33 (2008) 694-702.

[11] Oh MH, Lee KH, Yoon JH. Automated control strategies of inside slat-type blind considering visual comfort and building energy performance, Energy and Buildings 55 (2012) 728-737.

[12] Yun GY, Kim H, Kim JT. Effects of occupancy and lighting use patterns on lighting energy consumption, Energy and Buildings 46 (2012) 152-158.

[13] Košir M, Krainer A, Kristl Ž. Integral control system of indoor environment in continuously occupied spaces, Automation in Construction 21 (2012) 199-209.

[14] Li DHW, Lau CCS, Lam JC. Predicting daylight illuminance by computer simulation techniques, Lighting Research and Technology 36 (2004) 113-129.

[15] Li DHW, Cheung GHW, Lau CCS. A simplified procedure for determining indoor daylight illuminance using daylight coefficient concept, Building and Environment 41 (2006) 578-589.

[16] Kazanasmaz T, Günaydin M, Binol S. Artificial neural networks to predict daylight illuminance in office buildings, Building and Environment 44 (2009) 1751-1757.

[17] Lindelöf D. A fast daylight model suitable for embedded controllers, Solar Energy 83 (2009) 57-68.

[18] Larson GW, Shakespeare RA. Rendering with Radiance, Morgan Kaufmann Publishers, San Francisco, 2004.

[19] Reinhart C, Breton PF. Experimental validation of 3ds Max Design 2009 and Daysim 3.0, Eleventh International IBPSA Conference (2009), Glasgow, Scotland, 1514-1521.

[20] Laouadi A, Arsenault C. Validation of Skyvision, Institute of Research in Construction, Ottawa, 2004.

[21] Reinhart CF, Andersen M. Development and validation of a Radiance model for a translucent panel, Energy and Buildings 38 (2006) 890-904.

[22] Mardaljevic J. Validation of a lighting simulation program under real sky conditions, Lighting Research and Technology, 27 (1995) 181-188.

[23] Takagi T, Sugeno M. Fuzzy identification of systems and its applications to modelling and control, IEEE Transactions on Systems Man and Cybernetics 15 (1985) 116-132.

[24] Wang LX, Mendel JM. Fuzzy basis functions, universal approximation, and orthogonal least-squares learning, IEEE Transactions on Neural Networks 3 (1992) 807-814.

[25] Kosko B. Fuzzy systems as universal approximators, IEEE Transactions on Computers 43 (1994) 1329-1333.

[26] Ying GH. Necessary conditions for some typical fuzzy systems as universal approximators, Automatica 33 (1997) 1333-1338.

[27] Golberg RR. Methods of real analysis, John Wiley and Sons, New York, 1976.

[28] Lin CH. Siso nonlinear system identification using a fuzzy-neural hybrid system, International Journal on Neural Systems 8 (1997) 325-337.

SNE Simulation News

EUROSIM Data and Quick Info

EUROSIM 2019

10th EUROSIM Congress on Modelling and Simulation

La Rioja, Logroño, Spain, July 2019

Contents

Simulation Notes Europe SNE is the official membership journal of EUROSIM and distributed / available to members of the EUROSIM Societies as part of the membership benefits.
If you have any information, announcement, etc. you want to see published, please contact a member of the editorial board in your country or the editorial office. For scientific publications, please contact the EiC.

This *EUROSIM Data & Quick Info* compiles data from **EUROSIM** societies and groups: addresses, weblinks, and officers of societies with function and email, to be published regularly in **SNE** issues. This information is also published at EUROSIM's website *www.eurosim.info*.

SNE Editorial Office /ARGESIM

→ *www.sne-journal.org, www.eurosim.info*
✉ *office@sne-journal.org,* Andreas Körner, (info, news)
✉ *eic@sne-journal.org,* Felix Breitenecker (publications)
✉ SNE Editorial Office, Andreas Körner c/o ARGESIM / Mathematical Modelling & Simulation Group, TU Wien /101,
Wiedner Haupstrasse 8-10, 1040 Vienna , Austria

EUROSIM
Federation of European Simulation Societies

General Information. **EUROSIM**, the Federation of European Simulation Societies, was set up in 1989. The purpose of **EUROSIM** is to provide a European forum for simulation societies and groups to promote advancement of modelling and simulation in industry, research, and development. → *www.eurosim.info*

Member Societies. **EUROSIM** members may be national simulation societies and regional or international societies and groups dealing with modelling and simulation. At present **EUROSIM** has 16 *Full Members* and 2 *Observer Members*:

ASIM	Arbeitsgemeinschaft Simulation *Austria, Germany, Switzerland*
CEA-SMSG	Spanish Modelling and Simulation Group *Spain*
CROSSIM	Croatian Society for Simulation Modeling *Croatia*
CSSS	Czech and Slovak Simulation Society *Czech Republic, Slovak Republic*
DBSS	Dutch Benelux Simulation Society *Belgium, Netherlands*
FRANCOSIM	Société Francophone de Simulation *Belgium, France*
HSS	Hungarian Simulation Society; *Hungary*
ISCS	Italian Society for Computer Simulation *Italy*
KA-SIM	Kosovo Simulation Society, *Kosovo*
LIOPHANT	LIOPHANT Simulation Club *Italy & International*
LSS	Latvian Simulation Society; *Latvia*
PSCS	Polish Society for Computer Simulation *Poland*
MIMOS	Italian Modelling and Simulation Association, *Italy, Observer Member*
RNSS	Russian National Simulation Society *Russian Federation*
ROMSIM	Romanian Society for Modelling and Simulation, *Romania, Observer Member*
SIMS	Simulation Society of Scandinavia *Denmark, Finland, Norway, Sweden*
SLOSIM	Slovenian Simulation Society *Slovenia*
UKSIM	United Kingdom Simulation Society *UK, Ireland*

EUROSIM Board / Officers. EUROSIM is governed by a board consisting of one representative of each member society, president and past president, and representatives for **SNE** Simulation Notes Europe. The President is nominated by the society organising the next **EUROSIM** Congress. Secretary, Secretary to the Board, and Treasurer are elected out of members of the board.

President	Emilio Jiménez (CAE-SMSG), *emilio.jimenez@unirioja.es*
Past President	Esko Juuso (SIMS) *esko.juuso@oulu.fi*
Secretary	M. Mujica Mota (DBSS), *m.mujica.mota@hva.nl*
Treasurer	Felix Breitenecker (ASIM) *felix.breitenecker@tuwien.ac.at*
Secretary to the Board	Andreas Körner *Andreas.koerner@tuwien.ac.at*
SNE Repres.	Felix Breitenecker *felix.breitenecker@tuwien.ac.at*

SNE – Simulation Notes Europe. SNE is a scientific journal with reviewed contributions as well as a membership newsletter for **EUROSIM** with information from the societies in the *News Section*. **EUROSIM** societies are offered to distribute to their members the journal **SNE** as official membership journal. **SNE** Publishers are **EUROSIM**, ARGESIM and ASIM.

Editor-in-Chief	Felix Breitenecker *felix.breitenecker@tuwien.ac.at*

→ *www.sne-journal.org*,

✉ *office@sne-journal.org*

EUROSIM Congress. EUROSIM is running the triennial conference series **EUROSIM** Congress. The congress is organised by one of the **EUROSIM** societies.

EUROSIM 2019, the 10th EUROSIM Congress, will be organised by CAE-SMSG, the Spanish simulation society, in La Rioja, Logroño, Spain, in July 2019.

Chairs / Team EUROSIM 2019

Emilio Jiménez, EUROSIM President,
emilio.jimenez@unirioja.es
Juan Ignacio Latorre, *juanignacio.latorre@unavarra.es*

→ *www.eurosim.info*

EUROSIM Member Societies

ASIM
German Simulation Society
Arbeitsgemeinschaft Simulation

ASIM (Arbeitsgemeinschaft Simulation) is the association for simulation in the German speaking area, servicing mainly Germany, Switzerland and Austria. ASIM was founded in 1981 and has now about 600 individual members, and 90 institutional or industrial members.

→ *www.asim-gi.org* with members' area

info@asim-gi.org, admin@asim-gi.org

ASIM – Inst. f. Analysis and Scientific Computing
Vienna University of Technology
Wiedner Hauptstraße 8-10, 1040 Vienna, Austria

ASIM Officers	
President	Felix Breitenecker *felix.breitenecker@tuwien.ac.at*
Vice presidents	Sigrid Wenzel, *s.wenzel@uni-kassel.de*
	T. Pawletta, *pawel@mb.hs-wismar.de*
Secretary	Ch. Deatcu, *christina.deatcu@hs-wismar.de*
	A. Körner, *andreas.koerner@tuwien.ac.at*
Treasurer	Anna Mathe, *anna.mathe@tuwien.ac.at*
Membership Affairs	S. Wenzel, *s.wenzel@uni-kassel.de*
	W. Maurer, *werner.maurer@zhwin.ch*
	Ch. Deatcu, *christina.deatcu@hs-wismar.de*
	F. Breitenecker, *felix.breitenecker@tuwien.ac.at*
Repr. EUROSIM	F. Breitenecker, *felix.breitenecker@tuwien.ac.at*
	A. Körner, *andreas.koerner@tuwien.ac.at*
Int. Affairs – GI Contact	N. Popper, *niki.popper@drahtwarenhandlung.at*
	O. Rose, *Oliver.Rose@tu-dresden.de*
Editorial Board SNE	T. Pawletta, *pawel@mb.hs-wismar.de*
	Ch. Deatcu, *christina.deatcu@hs-wismar.de*
Web EUROSIM	A. Körner, *andreas.koerner@tuwien.ac.at*

Last data update *December 2016*

ASIM is organising / co-organising the following international conferences:

- ASIM Int. Conference 'Simulation in Production and Logistics' – bi-annual
- ASIM 'Symposium Simulation Technique' – bi-annual
- MATHMOD Int. Vienna Conference on Mathmatical Modelling – tri-annual

Furthermore, ASIM is co-sponsor of WSC – Winter Simulation Conference

ASIM Working Committee	
GMMS	Methods in Modelling and Simulation Th. Pawletta, *pawel@mb.hs-wismar.de*
SUG	Simulation in Environmental Systems J.Wittmann, *wittmann@informatik.uni-hamburg.de*
STS	Simulation of Technical Systems Walter Comerell, *Commerell@hs-ulm.de*
SPL	Simulation in Production and Logistics Sigrid Wenzel, *s.wenzel@uni-kassel.de*
EDU	Simulation in Education/Education in Simulation A. Körner, *andreas.koerner@tuwien.ac.at*
BIG DATA	Working Group Data-driven Simulation in Life Sciences; *niki.popper@drahtwarenhandlung.at*
	Working Groups for Simulation in Business Administration, in Traffic Systems, for Standardisation, etc.

CEA-SMSG – Spanish Modelling and Simulation Group

CEA is the Spanish Society on Automation and Control and it is the national member of IFAC (International Federation of Automatic Control) in Spain. Since 1968 CEA-IFAC looks after the development of the Automation in Spain, in its different issues: automatic control, robotics, SIMULATION, etc. In order to improve the efficiency and to deep into the different fields of Automation. The association is divided into national thematic groups, one of which is centered on Modeling, Simulation and Optimization, constituting the CEA Spanish Modeling and Simulation Group (CEA-SMSG). It looks after the development of the Modelling and Simulation (M&S) in Spain, working basically on all the issues concerning the use of M&S techniques as essential engineering tools for decision-making and optimization.

→ *http://www.ceautomatica.es/grupos/*

→ *emilio.jimenez@unirioja.es*
simulacion@cea-ifac.es

CEA-SMSG / Emilio Jiménez, Department of Electrical Engineering, University of La Rioja, San José de Calasanz 31, 26004 Logroño (La Rioja), SPAIN

CEA - SMSG Officers	
President	Emilio Jiménez, *emilio.jimenez@unirioja.es*
Vice president	Juan Ignacio Latorre *juanignacio.latorre@unavarra.es*
Repr. EUROSIM	Emilio Jiminez, *emilio.jimenez@unirioja.es*
Edit. Board SNE	Emilio Jiminez, *emilio.jimenez@unirioja.es*
Web EUROSIM	Mercedes Perez *mercedes.perez@unirioja.es*

Last data update June 2016

CROSSIM – Croatian Society for Simulation Modelling

CROSSIM-*Croatian Society for Simulation Modelling* was founded in 1992 as a non-profit society with the goal to promote knowledge and use of simulation methods and techniques and development of education. CROSSIM is a full member of **EUROSIM** since 1997.

→ *www.eurosim.info*

vdusak@foi.hr

✉ CROSSIM / Vesna Dušak
Faculty of Organization and
Informatics Varaždin, University of Zagreb
Pavlinska 2, HR-42000 Varaždin, Croatia

CROSSIM Officers	
President	Vesna Dušak, *vdusak@foi.hr*
Vice president	Jadranka Božikov, *jbozikov@snz.hr*
Secretary	Vesna Bosilj-Vukšić, *vbosilj@efzg.hr*
Executive board members	Vlatko Čerić, *vceric@efzg.hr*
	Tarzan Legović, *legovic@irb.hr*
Repr. EUROSIM	Jadranka Božikov, *jbozikov@snz.hr*
Edit. Board SNE	Vesna Dušak, *vdusak@foi.hr*
Web EUROSIM	Jadranka Bozikov, *jbozikov@snz.hr*

Last data update December2012

CSSS – Czech and Slovak Simulation Society

CSSS -The *Czech and Slovak Simulation Society* has about 150 members working in Czech and Slovak national scientific and technical societies (*Czech Society for Applied Cybernetics and Informatics*, *Slovak Society for Applied Cybernetics and Informatics*). The main objectives of the society are: development of education and training in the field of modelling and simulation, organising professional workshops and conferences, disseminating information about modelling and simulation activities in Europe. Since 1992, CSSS is full member of **EUROSIM**.

→ *www.fit.vutbr.cz/CSSS*

snorek@fel.cvut.cz

✉ CSSS / Miroslav Šnorek, CTU Prague
FEE, Dept. Computer Science and Engineering,
Karlovo nam. 13, 121 35 Praha 2, Czech Republic

CSSS Officers	
President	Miroslav Šnorek, *snorek@fel.cvut.cz*
Vice president	Mikuláš Alexík, *alexik@frtk.fri.utc.sk*
Scientific Secr.	A. Kavička, *Antonin.Kavicka@upce.cz*
Repr. EUROSIM	Miroslav Šnorek, *snorek@fel.cvut.cz*
Edit. Board SNE	Mikuláš Alexík, *alexik@frtk.fri.utc.sk*
Web EUROSIM	Petr Peringer, peringer@fit.vutbr.cz

Last data update December2012

DBSS – Dutch Benelux Simulation Society

The Dutch Benelux Simulation Society (DBSS) was founded in July 1986 in order to create an organisation of simulation professionals within the Dutch language area. DBSS has actively promoted creation of similar organisations in other language areas. DBSS is a member of **EUROSIM** and works in close cooperation with its members and with affiliated societies.

→ *www.DutchBSS.org*

a.w.heemink@its.tudelft.nl

✉ DBSS / A. W. Heemink
Delft University of Technology, ITS - twi,
Mekelweg 4, 2628 CD Delft, The Netherlands

DBSS Officers	
President	A. Heemink, *a.w.heemink@its.tudelft.nl*
Vice president	M. Mujica Mota, *m.mujica.mota@hva.nl*
Treasurer	M. Mujica Mota, *m.mujica.mota@hva.nl*
Secretary	P. M. Scala, p.m.scala@hva.nl
Repr. EUROSIM	M. Mujica Mota, *m.mujica.mota@hva.nl*
Edit. SNE/Web	M. Mujica Mota, *m.mujica.mota@hva.nl*

Last data update June 2016

FRANCOSIM - Société Francophone de Simulation

FRANCOSIM was founded in 1991 and aims to the promotion of simulation and research, in industry and academic fields.

djouani@u-pec.fr

✉ FRANCOSIM / Yskandar Hamam
Groupe ESIEE, Cité Descartes,
BP 99, 2 Bd. Blaise Pascal,
93162 Noisy le Grand CEDEX, France

FRANCOSIM Officers	
President	Karim Djouani, *djouani@u-pec.fr*
Treasurer	François Rocaries, *f.rocaries@esiee.fr*
Repr. EUROSIM	Karim Djouani, *djouani@u-pec.fr*
Edit. Board SNE	Karim Djouani, *djouani@u-pec.fr*

Last data update December2012

HSS – Hungarian Simulation Society

The Hungarian Member Society of EUROSIM was established in 1981 as an association promoting the exchange of information within the community of people involved in research, development, application and education of simulation in Hungary and also contributing to the enhancement of exchanging information between the Hungarian simulation community and the simulation communities abroad. HSS deals with the organization of lectures, exhibitions, demonstrations, and conferences.

→ *www.eurosim.info*

javor@eik.bme.hu

✉ HSS / András Jávor,
Budapest Univ. of Technology and Economics,
Sztoczek u. 4, 1111 Budapest, Hungary

HSS Officers	
President	András Jávor, *javor@eik.bme.hu*
Vice president	Gábor Szűcs, *szucs@itm.bme.hu*
Secretary	Ágnes Vigh, *vigh@itm.bme.hu*
Repr. EUROSIM	András Jávor, *javor@eik.bme.hu*
Deputy	Gábor Szűcs, *szucs@itm.bme.hu*
Edit. Board SNE	András Jávor, *javor@eik.bme.hu*
Web EUROSIM	Gábor Szűcs, *szucs@itm.bme.hu*

Last data update March 2008

ISCS – Italian Society for Computer Simulation

The Italian Society for Computer Simulation (ISCS) is a scientific non-profit association of members from industry, university, education and several public and research institutions with common interest in all fields of computer simulation.

→ *www.eurosim.info*

Mario.savastano@uniina.it

✉ ISCS / Mario Savastano,
c/o CNR - IRSIP,
Via Claudio 21, 80125 Napoli, Italy

ISCS Officers	
President	M. Savastano, *mario.savastano@unina.it*
Vice president	F. Maceri, *Franco.Maceri@uniroma2.it*
Repr. EUROSIM	F. Maceri, *Franco.Maceri@uniroma2.it*
Secretary	Paola Provenzano, *paola.provenzano@uniroma2.it*
Edit. Board SNE	M. Savastano, *mario.savastano@unina.it*

Last data update December2010

LIOPHANT Simulation

Liophant Simulation is a non-profit association born in order to be a trait-d'union among simulation developers and users; Liophant is devoted to promote and diffuse the simulation techniques and methodologies; the Association promotes exchange of students, sabbatical years, organization of International Conferences, courses and internships focused on M&S applications.

→ *www.liophant.org*

info@liophant.org

✉ LIOPHANT Simulation, c/o Agostino G. Bruzzone,
DIME, University of Genoa, Savona Campus
via Molinero 1, 17100 Savona (SV), Italy

LIOPHANT Officers	
President	A.G. Bruzzone, *agostino@itim.unige.it*
Director	E. Bocca, *enrico.bocca@liophant.org*
Secretary	A. Devoti, *devoti.a@iveco.com*
Treasurer	Marina Masseimassei@itim.unige.it
Repr. EUROSIM	A.G. Bruzzone, *agostino@itim.unige.it*
Deputy	F. Longo, *f.longo@unical.it*
Edit. Board SNE	F. Longo, *f.longo@unical.it*
Web EUROSIM	F. Longo, *f.longo@unical.it*

Last data update June 2016

LSS – Latvian Simulation Society

The Latvian Simulation Society (LSS) has been founded in 1990 as the first professional simulation organisation in the field of Modelling and simulation in the post-Soviet area. Its members represent the main simulation centres in Latvia, including both academic and industrial sectors.

→ *briedis.itl.rtu.lv/imb/*

merkur@itl.rtu.lv

✉ LSS / Yuri Merkuryev, Dept. of Modelling
and Simulation Riga Technical University
Kalku street 1, Riga, LV-1658, LATVIA

LSS Officers	
President	Yuri Merkuryev, *merkur@itl.rtu.lv*
Secretary	Artis Teilans, *Artis.Teilans@exigenservices.com*
Repr. EUROSIM	Yuri Merkuryev, *merkur@itl.rtu.lv*
Deputy	Artis Teilans, *Artis.Teilans@exigenservices.com*
Edit. Board SNE	Yuri Merkuryev, *merkur@itl.rtu.lv*
Web EUROSIM	Vitaly Bolshakov, *vitalijs.bolsakovs@rtu.lv*

Last data update June 2016

KA-SIM Kosovo Simulation Society

Kosova Association for Modeling and Simulation (KA – SIM, founded in 2009), is part of Kosova Association of Control, Automation and Systems Engineering (KA – CASE). KA–CASE was registered in 2006 as non Profit Organization and since 2009 is National Member of IFAC – International Federation of Automatic Control. KA-SIM joined EUROSIM as Observer Member in 2011. In 2016, KA-SIM became full member.

KA-SIM has about 50 members, and is organizing the international conference series International Conference in Business, Technology and Innovation, in November, in Durrhes, Albania, and IFAC Simulation Workshops in Pristina.

→ *www.ubt-uni.net/ka-case*

ehajrizi@ubt-uni.net

✉ MOD&SIM KA-CASE; Att. Dr. Edmond Hajrizi
Univ. for Business and Technology (UBT)
Lagjja Kalabria p.n., 10000 Prishtina, Kosovo

KA-SIM Officers	
President	Edmond Hajrizi, *ehajrizi@ubt-uni.net*
Vice president	Muzafer Shala, *info@ka-sim.com*
Secretary	Lulzim Beqiri, *info@ka-sim.com*
Treasurer	Selman Berisha, *info@ka-sim.com*
Repr. EUROSIM	Edmond Hajrizi, *ehajrizi@ubt-uni.net*
Deputy	Muzafer Shala, *info@ka-sim.com*
Edit. Board SNE	Edmond Hajrizi, *ehajrizi@ubt-uni.net*
Web EUROSIM	Betim Gashi, *info@ka-sim.com*

Last data update December 2016

PSCS – Polish Society for Computer Simulation

PSCS was founded in 1993 in Warsaw. PSCS is a scientific, non-profit association of members from universities, research institutes and industry in Poland with common interests in variety of methods of computer simulations and its applications. At present PSCS counts 257 members.

→ *www.eurosim.info* (*www.ptsk.man.bialystok.pl)*

leon@ibib.waw.pl

✉ PSCS / Leon Bobrowski, c/o IBIB PAN,
ul. Trojdena 4 (p.416), 02-109 Warszawa, Poland

PSCS Officers	
President	Leon Bobrowski, *leon@ibib.waw.pl*
Vice president	Tadeusz Nowicki, *Tadeusz.Nowicki@wat.edu.pl*
Treasurer	Z. Sosnowski, *zenon@ii.pb.bialystok.pl*
Secretary	Zdzislaw Galkowski, *Zdzislaw.Galkowski@simr.pw.edu.pl*
Repr. EUROSIM	Leon Bobrowski, *leon@ibib.waw.pl*
Deputy	Tadeusz Nowicki, *tadeusz.nowicki@wat.edu.pl*
Edit. Board SNE	Zenon Sosnowski, z.sosnowski@pb.ed.pl
Web EUROSIM	Magdalena Topczewska *m.topczewska@pb.edu.pl*

Last data update December2013

SIMS – Scandinavian Simulation Society

SIMS is the *Scandinavian Simulation Society* with members from the four Nordic countries Denmark, Finland, Norway and Sweden. The SIMS history goes back to 1959. SIMS practical matters are taken care of by the SIMS board consisting of two representatives from each Nordic country (Iceland one board member).

SIMS Structure. SIMS is organised as federation of regional societies. There are FinSim (Finnish Simulation Forum), DKSIM (Dansk Simuleringsforening) and NFA (Norsk Forening for Automatisering).

→ *www.scansims.org*

esko.juuso@oulu.fi

✉ SIMS / Erik Dahlquist, School of Business, Society and Engineering, Department of Energy, Building and Environment, Mälardalen University, P.O.Box 883, 72123 Västerås, Sweden

SIMS Officers	
President	Erik Dahlquist, *erik.dahlquist@mdh.se*
Vice president	Bernd Lie, *lie@hit.noe*
Treasurer	Vadim Engelson, *vadim.engelson@mathcore.com*
Repr. EUROSIM	Erik Dahlquist, *erik.dahlquist@mdh.se*
Edit. Board SNE	Esko Juuso, *esko.juuso@oulu.fi*
Web EUROSIM	Vadim Engelson, *vadim.engelson@mathcore.com*

Last data update June 2016

SLOSIM – Slovenian Society for Simulation and Modelling

SLOSIM - Slovenian Society for Simulation and Modelling was established in 1994 and became the full member of **EUROSIM** in 1996. Currently it has 90 members from both Slovenian universities, institutes, and industry. It promotes modelling and simulation approaches to problem solving in industrial as well as in academic environments by establishing communication and cooperation among corresponding teams.

→ www.slosim.si

slosim@fe.uni-lj.si

✉ SLOSIM / Vito Logar, Faculty of Electrical Engineering, University of Ljubljana, Tržaška 25, 1000 Ljubljana, Slovenia

SLOSIM Officers	
President	Vito Logar, *vito.logar@fe.uni-lj.si*
Vice president	Božidar Šarler, *bozidar.sarler@ung.si*
Secretary	Aleš Belič, *ales.belic@sandoz.com*
Treasurer	Milan Simčič, *milan.simcic@fe.uni-lj.si*
Repr. EUROSIM	B. Zupančič, *borut.zupancic@fe.uni-lj.si*
Deputy	Vito Logar, *vito.logar@fe.uni-lj.si*
Edit. Board SNE	B. Zupančič, *borut.zupancic@fe.uni-lj.si* Vito Logar, *vito.logar@fe.uni-lj.si* Blaž Rodič, *blaz.rodic@fis.unm.si*
Web EUROSIM	Vito Logar, *vito.logar@fe.uni-lj.si*

Last data update December 2016

UKSIM - United Kingdom Simulation Society

The UK Simulation Society is very active in organizing conferences, meetings and workshops. UKSim holds its annual conference in the March-April period. In recent years the conference has always been held at Emmanuel College, Cambridge. The Asia Modelling and Simulation Section (AMSS) of UKSim holds 4-5 conferences per year including the EMS (European Modelling Symposium), an event mainly aimed at young researchers, organized each year by UKSim in different European cities.

Membership of the UK Simulation Society is free to participants of any of our conferences and their co-authors.

→ *www.uksim.org.uk*

david.al-dabass@ntu.ac.uk

✉ UKSIM / Prof. David Al-Dabass
Computing & Informatics,
Nottingham Trent University
Clifton lane, Nottingham, NG11 8NS
United Kingdom

UKSIM Officers	
President	David Al-Dabass, *david.al-dabass@ntu.ac.uk*
Secretary	A. Orsoni, *A.Orsoni@kingston.ac.uk*
Treasurer	A. Orsoni, *A.Orsoni@kingston.ac.uk*
Membership chair	G. Jenkins, *glenn.l.jenkins@smu.ac.uk*
Local/Venue chair	Richard Cant, *richard.cant@ntu.ac.uk*
Repr. EUROSIM	A. Orsoni, *A.Orsoni@kingston.ac.uk*
Deputy	G. Jenkins, *glenn.l.jenkins@smu.ac.uk*
Edit. Board SNE	A. Orsoni, *A.Orsoni@kingston.ac.uk*

Last data update March 2016

RNSS – Russian Simulation Society

NSS - The Russian National Simulation Society (Национальное Общество Имитационного Моделирования – НОИМ) was officially registered in Russian Federation on February 11, 2011. In February 2012 NSS has been accepted as an observer member of **EUROSIM**, and in 2015 RNSS has become full member.

→ *www.simulation.su*

yusupov@iias.spb.su

✉ RNSS / R. M. Yusupov,
St. Petersburg Institute of Informatics and Automation RAS, 199178, St. Petersburg, 14th lin. V.O, 39

RNSS Officers	
President	R. M. Yusupov, *yusupov@iias.spb.su*
Chair Man. Board	A. Plotnikov, *plotnikov@sstc.spb.ru*
Secretary	M. Dolmatov, *dolmatov@simulation.su*
Repr. EUROSIM	R.M. Yusupov, *yusupov@iias.spb.su* Y. Senichenkov, *senyb@dcn.icc.spbstu.ru*
Deputy	B. Sokolov, *sokol@iias.spb.su*
Edit. Board SNE	Y. Senichenkov, *senyb@dcn.icc.spbstu.ru*

Last data update June 2016

EUROSIM OBSERVER MEMBERS

ROMSIM – Romanian Modelling and Simulation Society

ROMSIM has been founded in 1990 as a non-profit society, devoted to theoretical and applied aspects of modelling and simulation of systems. ROMSIM currently has about 100 members from Romania and Moldavia.

→ *www.eurosim.info* (*www.ici.ro/romsim*)

sflorin@ici.ro

ROMSIM / Florin Hartescu,
National Institute for Research in Informatics, Averescu Av. 8 – 10, 71316 Bucharest, Romania

ROMSIM Officers	
President	
Vice president	Florin Hartescu, *flory@ici.ro* Marius Radulescu, *mradulescu@ici.ro*
Repr. EUROSIM	Florin Stanciulescu, *sflorin@ici.ro*
Deputy	Marius Radulescu, *mradulescu@ici.ro*
Edit. Board SNE	
Web EUROSIM	Zoe Radulescu, *radulescu@ici.ro*

Last data update partly June 2016

MIMOS – Italian Modelling and Simulation Association

MIMOS (Movimento Italiano Modellazione e Simulazione – Italian Modelling and Simulation Association) is the Italian association grouping companies, professionals, universities, and research institutions working in the field of modelling, simulation, virtual reality and 3D, with the aim of enhancing the culture of 'virtuality' in Italy, in every application area.

MIMOS became **EUROSIM** Observer Member in 2016 and is preparing application for full membership.

→ *www.mimos.it*

roma@mimos.it – info@mimos.it

MIMOS – Movimento Italiano Modellazione e Simulazione; via Ugo Foscolo 4, 10126 Torino – via Laurentina 760, 00143 Roma

MIMOS Officers	
President	Paolo Proietti, *roma@mimos.it*
Secretary	Davide Borra, *segreteria@mimos.it*
Treasurer	Davide Borra, *segreteria@mimos.it*
Repr. EUROSIM	Paolo Proietti, *roma@mimos.it*
Deputy	Agostino Bruzzone, *agostino@itim.unige.it*
Edit. Board SNE	Paolo Proietti, *roma@mimos.it*

Last data update December 2016

CANDIDATES

Albanian Simulation Society

At the Department of Statistics and Applied Informatics, Faculty of Economy, University of Tirana, Prof. Dr. Kozeta Sevrani at present is setting up an Albanian Simulation Society. Kozeta Sevrani, professor of Computer Science and Management Information Systems, and head of the Department of Mathematics, Statistics and Applied Informatic, has attended a EUROSIM board meeting in Vienna and has presented simulation activities in Albania and the new simulation society.

The society – constitution and bylaws are being worked out - will be involved in different international and local simulation projects, and will be engaged in the organisation of the conference series ISTI – Information Systems and Technology. The society intends to become a EUROSIM Observer Member.

kozeta.sevrani@unitir.edu.al

Albanian Simulation Goup, attn. Kozeta Sevrani
University of Tirana, Faculty of Economy
rr. Elbasanit, Tirana 355 Albania

Albanian Simulation Society- Officers (Planned)	
President	Kozeta Sevrani, *kozeta.sevrani@unitir.edu.al*
Secretary	
Treasurer	
Repr. EUROSIM	Kozeta Sevrani, *kozeta.sevrani@unitir.edu.al*
Edit. Board SNE	Albana Gorishti, *albana.gorishti@unitir.edu.al* Majlinda Godolja, *majlinda.godolja@fshn.edu.al*

Last data update December 2016

EUROSIM 2019

9th EUROSIM Congress on Modelling and Simulation

La Rioja, Logroño, Spain, July 2019

EUROSIM Congresses are the most important modelling and simulation events in Europe. For EUROSIM 2019, we are soliciting original submissions describing novel research and developments in the following (and related) areas of interest: Continuous, discrete (event) and hybrid modelling, simulation, identification and optimization approaches. Two basic contribution motivations are expected: M&S Methods and Technologies and M&S Applications. Contributions from both technical and non-technical areas are welcome.

Congress Topics The EUROSIM 2019 Congress will include invited talks, parallel, special and poster sessions, exhibition and versatile technical and social tours. The Congress topics of interest include, but are not limited to:

- Intelligent Systems and Applications
- Hybrid and Soft Computing
- Data & Semantic Mining
- Neural Networks, Fuzzy Systems & Evolutionary Computation
- Image, Speech & Signal Processing
- Systems Intelligence and Intelligence Systems
- Autonomous Systems
- Energy and Power Systems
- Mining and Metal Industry
- Forest Industry
- Buildings and Construction
- Communication Systems
- Circuits, Sensors and Devices
- Security Modelling and Simulation
- Bioinformatics, Medicine, Pharmacy and Bioengineering
- Water and Wastewater Treatment, Sludge Management and Biogas Production
- Condition monitoring, Mechatronics and maintenance
- Automotive applications
- e-Science and e-Systems
- Industry, Business, Management, Human Factors and Social Issues
- Virtual Reality, Visualization, Computer Art and Games
- Internet Modelling, Semantic Web and Ontologies
- Computational Finance & Economics
- Simulation Methodologies and Tools
- Parallel and Distributed Architectures and Systems
- Operations Research
- Discrete Event Systems
- Manufacturing and Workflows
- Adaptive Dynamic Programming and Reinforcement Learning
- Mobile/Ad hoc wireless networks, mobicast, sensor placement, target tracking
- Control of Intelligent Systems
- Robotics, Cybernetics, Control Engineering, & Manufacturing
- Transport, Logistics, Harbour, Shipping and Marine Simulation

Congress Venue / Social Events The Congress will be held in the City of Logroño, Capital of La Rioja, Northern Spain. The main venue and the exhibition site is the University of La Rioja (UR), located on a modern campus in Logroño, capital of La Rioja, where 7500 students are registered. The UR is the only University in this small, quiet region in Northern Spain. La Rioja is where the Monasteries of San Millán de la Cogolla, cradle of the first words written in the Spanish language, are situated, sites included in UNESCO's World Heritage List in 1996. Of course, social events will reflect this heritage – and the famous wines in la Rioja.

Congress Team: The Congress is organised by CAE CAE-SMSG, the Spanish simulation society, and Universidad de la Rioja.

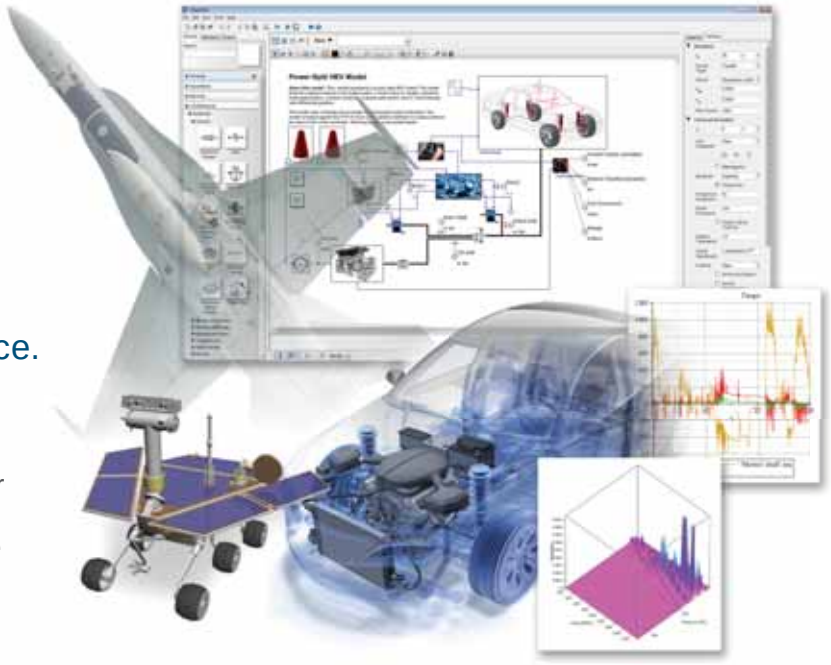
Info: Emilio Jiménez, EUROSIM President, emilio.jimenez@unirioja.es

Juan Ignacio Latorre, juanignacio.latorre@unavarra.es

www.eurosim.info

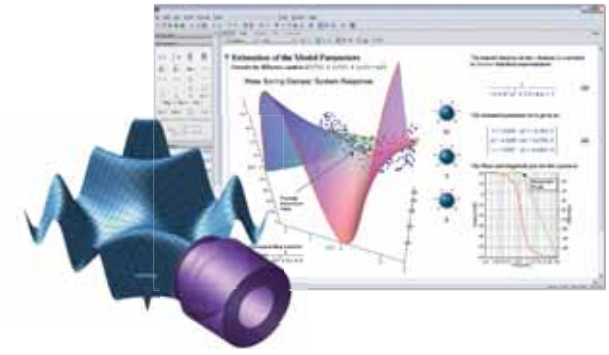
A modern approach to modeling and simulation

MapleSim™



With MapleSim, educators have an industry-proven tool to help bridge the gap between theory and practice.

- MapleSim illustrates concepts, and helps students learn the connection between theory and physical behavior
- A wide variety of models are available to help get started right away



Maple™

MapleSim is built on Maple, which combines the world's most powerful mathematical computation engine with an intuitive, "clickable" user interface.

To learn more about how you can reinforce engineering concepts using a combination of theory, simulation, and hardware, view this webinar.

www.maplesoft.com/SNEWebinar

Contact us: +49 (0)241/980919-30

Maplesoft
Mathematics • Modeling • Simulation
A Cybernet Group Company

www.maplesoft.com | vertrieb@maplesoft.com