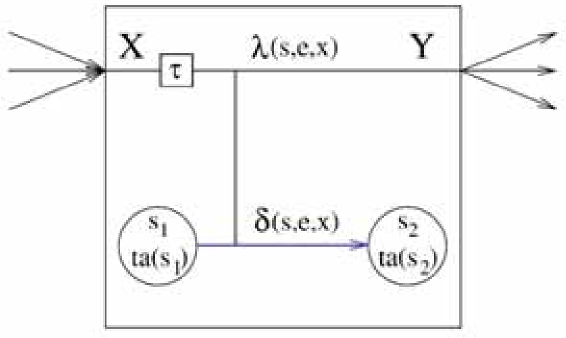
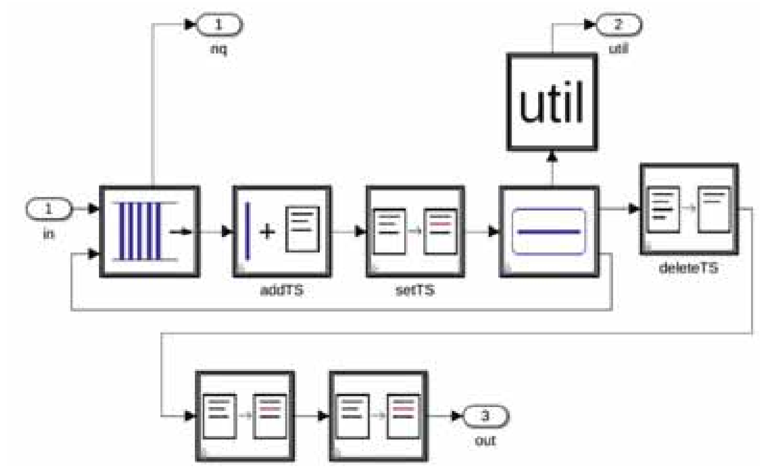
SNE SIMULATION NOTES EUROPE



NSA - DEVS

We invite all modelers interested in discrete-event modeling to try out these tools, ask for enhancements or even provide useful new atomics to the library !



Journal on Developments and Trends in Modelling and Simulation

EUROSIM Scientific Membership Journal

Vol. 36 No.1, March 2026
ISSN Online 2306-0271
ISSN Print 2305-9974
DOI 10.11128/sne.36.1.1076

ARGESIM

Calendar of Events

March 25 - 27, 2026, Nürnberg/Ansbach, Germany

ASIM Workshop Simulation in den Umwelt-
und Geowissenschaften

Strömungssimulation und industrielle Energieeffizienz

www.asim-gi.org/SUGWorkshop2026

June 24 - 26, 2026, TU Wien, Austria

40th ACM SIGSIM International Conference
on Principles of Advanced Discrete Simulation

Cooperation of ASIM with ACM SIGSIM

sigsim.acm.org/conf/pads/2026/

September 16 -17, 2026, Eskilstuna, Sweden

SIMS 2026, 67th International Conference
of the Scandinavian Simulation Society

www.scansims.org

September 21 - 23, 2026, Genova, Italy

12th EUROSIM Congress

M&S & AI: Opportunities and Challenges for Present & Future
with special ASIM Tracks
(reduced fee for ASIM members)

www.msc-les.org/eurosim2026/

together with

I3M 2026

23rd International Multidisciplinary Modelling & Simulation Multi-Conference

www.msc-les.org/i3m2026/

December 6 - 9, 2026, Glasgow, Scotland

2026 Winter Simulation Conference

Simulation for Climate Resilience
(reduced fee for ASIM members)

meetings.informs.org/wordpress/wsc2026/

Editorial

Dear Readers, This fist issue of SNE Volume 36, SNE 36(1), starts with a Project Note: P. Junglas et al. present project development of NSA-DEVS for discrete-event modelling and they 'invite all modelers interested in discrete-event modelling to try out these tools, ask for enhancements or even provide useful new atomics to the library!' we hope for response! Then the issue continues with post-conference publications from ASIM Symposium 2024 (Munich) and from ASIM's Workshop GMMS/STS 2025 at DLR (April 2025). The topics of these contributions show a broad range: generation of simulation models using multi-agent systems, supplier-based market model with rich dynamics, deadlock resolution and collision reduction via deep RL, simulation framework for stochastic scheduling problems, model-based analysis of a diesel generator, forecast-based MPC of multi-energy systems, and a simulator for linear implicit equilibrium dynamics. And last but not least, P. G. and C. A. Ioannou continue ARGESIM Benchmarks solutions with STROBOSCOPE, in this issue for Benchmark C6 'Emergency Department—Follow-Up Treatment'.

The next issues will continue with post-conference publications from ASIM's Workshop 2025 at DLR and from MATHMOD 2025 Vienna, - and a special issue with contributions of ASIM's Conference Simulation in Production and Logistics (Dresden, 2025).

*I would like to thank all authors for their contributions, and many thanks to the SNE Editorial Office for layout, typesetting, preparations for printing, electronic publishing, and much more. And have a look at the info on EUROSIM-related simulation events of this year: 12th EUROSIM Congress **EUROSIM 2026** in September 2026 in Genova (with ASIM tracks!), WinterSim Conference 2026 in December in Glasgow (this year in Europe!), and further conferences, e.g. SIMS 2026 in September in Eskilstuna, Sweden.*

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

Contents SNE 36(1)

e-SNE 36(1), DOI 10.11128/sne.36.1.1076

www.sne-journal.org/sne-volumes/volume-36

SNE Basic e-Version with Open Access, CC BY 4.0

SNE Full e-Version for Members of publication-active EUROSIM Societies: ASIM, CEA-SMSG, CSSS, DBSS, KA-SIM, LIOPHANT, SIMS, SLOSIM, UKSIM

print-SNE for members on demand (printer INTU TU Wien)

SNE Contact & Info

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

→ *www.sne-journal.org*

office@sne-journal.org, eic@sne-journal.org

SNE Editorial Office
Felix Breitenecker (Organisation, Author Mentoring)
Irmgard Husinsky (Web, Electronic Publishing),
ARGESIM/Math. Modelling & Simulation Group,
Inst. of Analysis and Scientific Computing, TU Wien
Wiedner Hauptstrasse 8-10, 1040 Vienna, Austria

SNE Simulation Notes Europe

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

Scope: Developments and trends in modelling and simulation in various areas and in application and theory; comparative studies and benchmarks; modelling and simulation in and for education; post-conference publication for conferences of EUROSIM societies, society information and membership information for EUROSIM members (Federation of European Simulation Societies and Groups).

Editor-in-Chief: Felix Breitenecker, TU Wien, Math. Modelling Group
Felix.Breitenecker@tuwien.ac.at, *eic@sne-journal.org*

Print SNE: INTU (TU Wien), Wiedner Hauptstrasse 8-10, 1040, Vienna, Austria – *www.intu.at*

ARGESIM Publisher: Arbeitsgemeinschaft Simulation News c/o Math. Modelling and Simulation Group, TU Wien / 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*

SNE is a joint publication of **ARGESIM** and **ASIM.**

SNE - Aims and Scope

Simulation Notes Europe (SNE) provides an international, high-quality forum for presentation of new ideas and approaches in simulation - from modelling to experiment analysis, from implementation to verification, from validation to identification, from numerics to visualisation (www.sne-journal.org).

SNE seeks to serve scientists, researchers, developers and users of the simulation process across a variety of theoretical and applied fields in pursuit of novel ideas in simulation. **SNE** follows the recent developments and trends of modelling and simulation in new and/or joining areas, as complex systems and big data. **SNE** puts special emphasis on the overall view in simulation, and on comparative investigations, as benchmarks and comparisons in methodology and application. For this purpose, **SNE** documents the **ARGESIM Benchmarks** on *Modelling Approaches and Simulation Implementations* with publication of definitions, solutions and discussions. **SNE** welcomes also contributions in education in/for/with simulation.

SNE is the scientific membership journal of **EUROSIM**, the *Federation of European Simulation Societies and Simulation Groups* (www.eurosim.info), also providing Postconference publication for events of the member societies. **SNE**, primarily an electronic journal **e-SNE** (ISSN 2306-0271), follows an open access strategy, with free download in basic version (B/W, low resolution graphics). Members of most **EUROSIM** societies are entitled to download **e-SNE** in an elaborate full version (colour, high resolution graphics), and to access additional sources of benchmark publications, model sources, etc. (via group login of the society), **print-SNE** (ISSN 2305-9974) is available for specific groups of **EUROSIM** societies.

SNE is published by **ARGESIM** (www.argesim.org) on mandate of **EUROSIM** and **ASIM** (www.asim-gi.org), the German simulation society. **SNE** is DOI indexed with prefix 10.11128.

Author's Info. Individual submissions of scientific papers are welcome, as well as post-conference publications of contributions from conferences of **EUROSIM** societies. **SNE** welcomes special issues, either dedicated to special areas and/or new developments, or on occasion of events as conferences and workshops with special emphasis.

Authors are invited to submit contributions which have not been published and have not being considered for publication elsewhere to the **SNE** Editorial Office.

SNE distinguishes different types of contributions (*Notes*), i.e.

- **TN** Technical Note, 6–10 p.
- **SN** Short Note, max. 5 p.
- **SW** Software Note , 4–6 p.
- **BN** Benchmark Note, 2–10 p.
- **ON** Overview Note – only upon invitation, up to 14 p.
- **EN** Education Note, 6–8 p.
- **PN** Project Note 6–8 p.
- **STN** Student Note, 4-6 p., on supervisor's recommendation
- **EBN** Educational Benchmark Note, 4–10 p.

Further info and templates (doc, tex) at **SNE**'s website, or from the Editor-in-Chief

www.sne-journal.org

office@sne-journal.org, *eic@sne-journal.org*

SNE Editorial Board

SNE - Simulation Notes Europe is advised and supervised by an international scientific editorial board. This board is taking care on peer reviewing of submission to SNE (and extended for special issues and Postconference publication):

Using Component-based Discrete-event Modeling with NSA-DEVS – an Invitation

Peter Junglas[1*], David Jammer[2], Thorsten Pawletta[2], Sven Pawletta[2]

[1]PHWT-Institut, PHWT Vechta/Diepholz, Am Campus 2, 49356 Diepholz, Germany; *peter@peter-junglas.de

[2]Research Group Computational Engineering and Automation, University of Applied Sciences Wismar, Philipp-Müller-Straße 14, 23966 Wismar, Germany

SNE 36(1), 2026, 1-8, DOI: 10.11128/sne.36.pn.10761 Selected ASIM SST 2024 Postconf. Publication: 2025-01-10 Received Revised: 2026-01-15; Accepted: 2026-01-30
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna
ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. The quest for a simulation scheme that combines the preciseness of the PDEVS formalism with the ease of use of standard simulation environments has lead to the definition of NSA-DEVS, which has meanwhile been shown to provide a useful basis for real-world applications. A set of modeling and simulation tools for NSA-DEVS is freely available that uses Matlab as programming language and the graphical editor of Simulink for the construction of complex models from simple atomic components.

To demonstrate that these tools are ready for general discrete-event based applications, the implementation of a textbook example is presented in some detail. It is shown that components that are necessary for a transaction-oriented style can be modeled easily, leading to a comprehensible model with a solid mathematical basis.

Introduction

For modeling and simulation using the discrete event approach, practitioners can choose between a lot of commercial simulation environments, which provide users with a wide range of components and helpful tools [1]. However, the behaviour of complex models can sometimes be different than expected, especially because the documentation often does not provide all necessary details.

In such situations users generally build a toolset of workarounds to make things work. But this often leads to conceptual problems and does not deepen the understanding of the precise behaviour of a model [2].

On the other hand, one could start instead with a precise description of the model and its components, using the well-established PDEVS formalism [3]. There are even a few free tools that provide a user-extensible set of DEVS-based components and a graphical user interface for combining components to build complex models [4]. But Preyser et al. have shown in [5] that the way, how PDEVS uses transitory states (i. e. states with lifetime 0), makes it hard to define some simple reusable components, especially when they show Mealy behaviour. Therefore they proposed a revised version of the PDEVS formalism [6] that allows for the direct description of Mealy components.

However, this formalism still has problems with chains of concurrent events [7]. Therefore it has been extended to NSA-DEVS (*Non-Standard Analysis DEVS*), which solves these difficulties by formally introducing infinitesimal delays. This idea has been analyzed thoroughly in [8, 9] and used to implement a large real-world example [10]. A corresponding simulation environment has been built, which contains graphical tools and a growing library of components. It is based on Matlab and the graphical editor of Simulink and is freely available from [11].

Since this article is primarily an invitation to use these new tools for modeling and simulation in discrete-event based studies, it concentrates on practical aspects, not on the underlying mathematical formalism. After a short definition of NSA-DEVS and a recapitulation of previous results, the structure of the tools and the basic workflow for concrete studies will be presented in some detail. A basic example from Law's textbook [12] will illustrate how to implement and apply components for standard entity-based applications.

1 Definition of NSA-DEVS

Like the PDEVS specification, the NSA-DEVS formalism describes two types of models: *atomic models*, which are the basic components, and *coupled models*, which combine atomic and coupled models in a hierarchical structure.

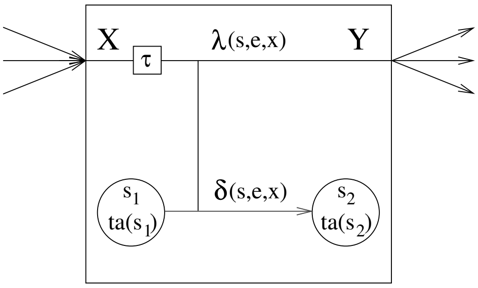


Figure 1: Dynamics of an atomic component.

An atomic model (cf. Fig. 1) is given by a set X of input ports, each with a name, a similar set Y of output ports, a set S of internal states and an input delay time τ. Each state s has a lifetime, given by the time advance function $ta(s)$. The behaviour is given by the output function λ, which defines output values y, and the transition function δ, which computes the next internal state. Both functions depend on three values: the current state s, the elapsed time e since the last transition and the input values x. When an external event, i. e. a set x of input values, occurs at time t, λ is called at time $t+\tau$, followed by an immediate call of δ. An internal event, i.e. a state change after a waiting time $ta(s)$, leads to a direct (undelayed) call of λ and δ.

The essential modification of NSA-DEVS is the introduction of the input delay together with extended time values: Introducing an infinitesimal value $\varepsilon > 0$, times are defined as values of the form $a+b\varepsilon$. The delay time is usually defined as $\tau = \varepsilon$ and only changed at rare occasions to guarantee a given order of events. Furthermore, the lifetime of states can never be 0, but an “immediate” (transitory) state change needs at least an infinitesimal time τ_D.

A coupled model is basically a set of several lists that describe the submodels used (atomic or coupled), the input and output ports of the coupled model and all connections between the submodels and from or to the ports of the coupled model. Inputs of a coupled model are immediate, i. e. they have no additional input delays.

2 Current Status of NSA-DEVS

The fundamental insight of [5] was that while the basic PDEVS formalism is sufficient to model any discrete-event based system, this is generally not possible with every reusable component.

After the definition of NSA-DEVS one had therefore to show that it was up to this task.

The first step was the definition of a corresponding abstract simulator [8]. According to the DEVS philosophy, the simulator is actually part of the definition of the formalism. It adds a semantic layer to the static definition of models by defining their exact behaviour. In a next step [9], a set of standard examples was defined formally and implemented using simple reusable atomic components.

A special point of interest was, how one can define the infinitesimal parameters introduced by NSA-DEVS in a simple and systematic way. This was studied further with a large real-world example in [10] consisting of 391 atomic and 88 coupled models in 5 hierarchical levels. It contained 391 input delay times τ and 12 additional delays τ_D for transitory states. Its construction has been simplified by the introduction of a graphical model builder, which uses Simulink for the definition of coupled models.

In the course of these investigations, a basic set of atomic models has been defined and implemented:

- sources (constant, several generators),
- math operations (add, gain, multiply, divide, compare),
- logic operations and flipflops based on IEEE 1164,
- routing components (combine, distribute),
- a QSS-based integrator,
- a sink (toworkspace) for logging simulation results,
- common logistics components (queue, server, batch, unbatch, terminator),
- statistical computations (getmax, utilization).

For all atomic models, corresponding NSA-DEVS blocks are provided for the Simulink editor and assembled in libraries. They make it possible to construct coupled models using Simulink's graphical capabilities for positioning and connecting blocks and ports.

In addition, block parameters can be set, among them the values for the input delay τ and, where necessary, the transition delay τ_D.

All delay values are predefined and usually set to the default value $\tau_{def} = \varepsilon$. An exception are components that emit trains of output values with infinitesimal time distances, such as queue and combine atomics. They need larger values for τ_D, which are predefined in the library to $\tau_D = 2\varepsilon$. This often works, but has to be enlarged in special cases. If a special ordering is requested for loops that contain several sequences of components, one has to increase a few input delays to slow down some paths [10].

3 Implementation in Matlab

For the implementation of the example models, a set of tools and a component library have been constructed that are based on Matlab and the Simulink editor. Using a simple example, we will describe the basic workflow necessary to implement own models and running simulations.

The atomic models are the basic building blocks. They are implemented in Matlab as classes that contain a constructor and the methods `delta`, `lambda` and `ta`. A simple example is the class `am_add2` for the addition of two input values (cf. Listing 1) and shows how to implement a Mealy component. On first sight, it looks similar to its counterpart in a continuous environment, but the discrete-event nature leads to a very typical change: Since input values are defined only, when an input event arrives, these values have to be stored internally using properties `in1` and `in2`. The property `s` is used throughout the whole library to denote "macroscopic" states, which is useful in more complex atomics. Here it is simply set to a constant value. The remaining properties store the values of external parameters, in this case the name of the component, the input delay and a debug flag.

The constructor provides initial values for all properties, its parameter list defines the set of external parameters of the component. The `delta` method just stores incoming values. The `lambda` method computes the output value, which is given here by the sum of the incoming or stored values. Finally the `ta` method returns the lifetime of the state, which is always given as a two-dimensional vector `[a, b]`, denoting the time $t = a + b\varepsilon$. In this case it is always infinite.

As a graphical representation of `am_add2` a Simulink subsystem with the name `am_add2` is stored in an NSA-DEVS library using the Simulink editor.

Internally it just consists of unconnected input and output ports, which have the names of the ports that are used inside the atomic model.

Additionally, it has a mask that defines the order and values of all parameters – except `name`, which is set to the name of the actual component – and short description and help texts (cf. Fig. 2).

Listing 1: Atomic model am_add2.

```
classdef am_add2 < handle
  properties
    s
    in1
    in2
    name
    tau
    debug
  end
  methods
    function obj = am_add2(name, tau,
        debug)
      obj.s = "running";
      obj.in1 = 0;
      obj.in2 = 0;
      obj.name = name;
      obj.debug = debug;
      obj.tau = tau;
    end
    function delta(obj,e,x)
      if isfield(x, "in1")
        obj.in1 = x.in1;
      end
      if isfield(x, "in2")
        obj.in2 = x.in2;
      end
    end
    function y = lambda(obj,e,x)
      s1 = obj.in1;
      s2 = obj.in2;
      if isfield(x, "in1")
        s1 = x.in1;
      end
      if isfield(x, "in2")
        s2 = x.in2;
      end
      y.out = s1 + s2;
    end
    function t = ta(obj)
      t = [inf, 0];
    end
  end
end
```

A coupled model, such as the simple example model `demo1` shown in Fig. 3, is defined by a Matlab function that creates all its atomic and – using recursion – its coupled components and all connections.

To simplify this tedious and error prone programming task, the user instead builds the model from the Simulink representation by copying the components from the library and connecting them in the standard way. The `toworkspace` models are used here to collect the simulation results. In the example, their parameter `varname` is set to `"input"` and `"result"` respectively.

Figure 2: Mask of the `am_add2` atomic.

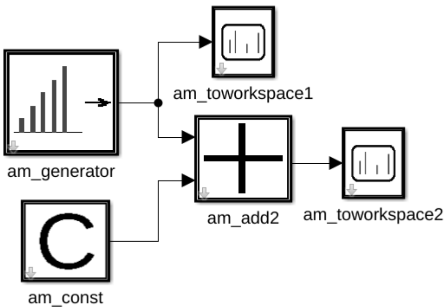


Figure 3: Coupled model `demo1`.

The very simple script shown in Listing 2 can now be used to create and run the model and plot the simulation results.

Listing 2: Run script for model `demo1`.

```
function testDemo()
  tEnd = 6;

  model_generator("demo1");
  out = model_simulator("demo1", tEnd);
  plot_results(out, tEnd);
end
```

From the Simulink representation, the `model_generator` creates the Matlab scripts for all coupled models. Next the `model_simulator` runs the model and collects all results in the struct variable `out`.

In the example model it contains the two fields `out.input` and `out.results`, which each have subfields `t` and `y` for the time and result values. They can now be plotted easily with Matlab's standard functions.

A typical result is shown in Fig. 4. The `am_generator` creates increasing numbers, starting at $t = 1$, which are shifted by the constant value 3. The output event at $t = 0$ probably comes unexpected. It is due to the `am_const` atomic that sends its value only once at the beginning.

It is then stored in `am_add2` and added to the initial value 0 stored for the other input port. One should bear in mind that the coupled models look like Simulink models, but the inner workings of discrete-event models are still very different.

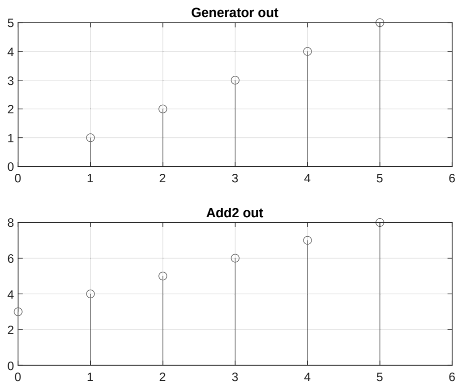


Figure 4: Plot of simulation results for the model `demo1`.

Discrete-event models can easily contain very hard to find errors. To support the debugging process, the toolset supports three different levels, from simple time stamps over debug outputs from individually chosen atomic components to a complete output of internal simulator messages. The last level creates a graphical representation of all messages with a huge amount of information and is usually only useful for simple test models.

4 Example Model

In order to show that the methods and tools presented can be used directly for transaction-based modeling, we will implement a standard textbook example [12]. The model describes a time-shared computer with N terminals, which submit jobs of varying computing time demands.

These jobs are processed on a single CPU in time-slices of length q using a round-robin scheduler with a switching time t_{swap}. When a job completes, a new job is created after a waiting time. The waiting and processing times are exponentially distributed random variables with mean values t_W and t_S. After the completion of N_J jobs, the average response time and queue length and the CPU utilization are computed.

A common method to implement such a model uses entities describing the jobs, which contain attributes such as the service time, i. e. the remaining processing time, or the start time. Entity attributes are implemented using struct variables.

Four atomics have been created to handle such entity attributes (cf. Fig. 5): `am_adddata` adds a set of fields denoting new attributes to each incoming entity. If the input is not already an entity (i. e. of type struct), an entity is created with an additional attribute that stores the input value. `am_writedata` changes the value of an entity attribute using values from other attributes. The changing function is defined as string parameter describing an arbitrary Matlab command. `am_readdata` outputs the value of an attribute from the input entity and `am_deletedata` deletes a set of attributes.

Figure 5: Atomic models for entity handling.

A few existing atomics have been modified to optionally read an attribute from an incoming entity instead of using a parameter or an input, among them the `server` and `distribute` components.

With these atomics, one can create a coupled model for a server with exponentially distributed service times t_S (cf. Fig. 6): The `addTS` component adds an attribute to store the value of t_S, `setTS` sets the value of this attribute using the Matlab command string

```
"out = -" + tS + "*log(rand());"
```

where the variable `tS` is the mean service time, given by a mask parameter. The `am_server` component uses the attribute of incoming entities to set the current service time. Finally, `deleteTS` deletes the attribute for the sake of better encapsulation.

Using a different formula for the computation of t_S, which uses several attributes, one can implement complex strategies for calculating the service time.

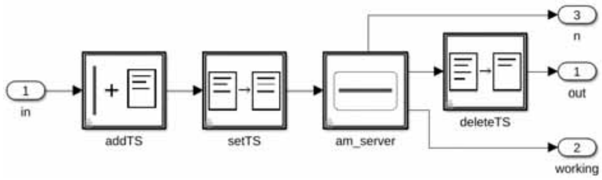


Figure 6: Coupled model of a server with entity-dependent service times.

A standard component in a transaction-based environment is the N-server, which can serve up to N incoming entities, each with its own service time. Its exact behaviour can be quite complicated and often is not transparent to the user of a commercial program. The NSA-DEVS description eliminates all ambiguities that arise e. g. with several incoming and outgoing entities at the same time.

The diagram in Fig. 7 describes the basic behaviour of the `am_nserver` atomic, where monitoring compliance with the maximum server capacity has been omitted for better readability. Among its properties are a list E of entities in the server, a corresponding list σ of remaining service times and a list $qOut$ of outgoing entities. An interesting difference to the simple server is the possibility that several entities can be ready at the same time. To handle this, the N-server moves all finished entities to $qOut$, changes to the state *emitting* and outputs them with a time delay of t_D. According to the rules stated above, t_D is predefined as 2ε, but may need to be enlarged in special applications. All details defining the exact behaviour can be found in the open source code of `am_nserver.m` [11].

With these atomics the three coupled models `Terminals`, `CPU` and the complete model `timeShared` can be assembled easily.

The `Terminals` model (cf. Fig. 8) starts with a generator `initialJobs` that creates N entities at time 0 (more precisely at times $n \cdot t_D$) with consecutive IDs. They get the attributes `startTime`, `outPort` and `remainingServiceTime`, which is set to the individual service times.

An N-server implements the individual waiting times. The `startTime` is set to the current simulation time, using the utility function *get_time*(), and the entities proceed to the CPU.

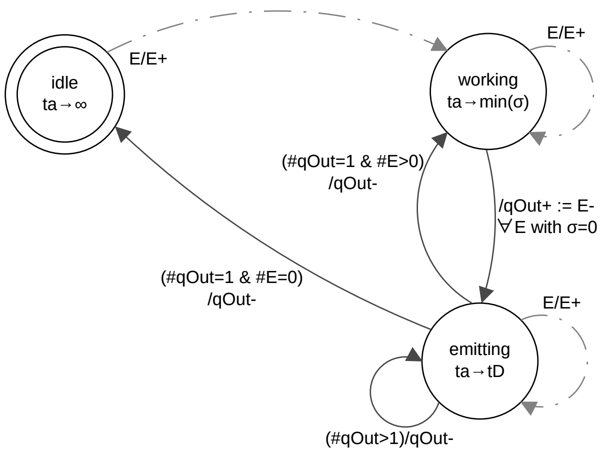


Figure 7: State diagram of the N-server component.

⟶	state transition due to internal event
⇢	state transition due to external event
ta	lifetime of the state
E	external event (entity at input)
E+	insert entity in list E
E-	remove entity from list E
#E	number of entities in list E

When a job is fully processed by the CPU, a corresponding ID is sent as external event to the input of coupled model `Terminals`, where it is promoted to a new job by the atomic model `constAdder`.

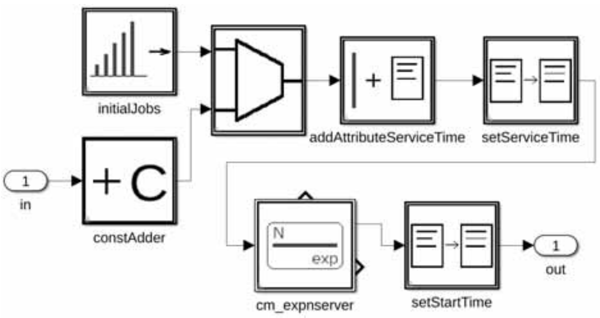


Figure 8: Coupled model `Terminals`.

The `CPU` model (cf. Fig. 9) starts with a queue for the waiting jobs, followed by a server representing the CPU proper. Its service time is set to $q + t_{swap}$ or less, if the job is almost ready. After processing, the entity attributes are updated: The time slice is subtracted from the `remainingServiceTime` and `outPort` is set to 1, if the job is ready, or 2 otherwise.

The internal feedback from the server to the queue signals to the queue whether the server is available.

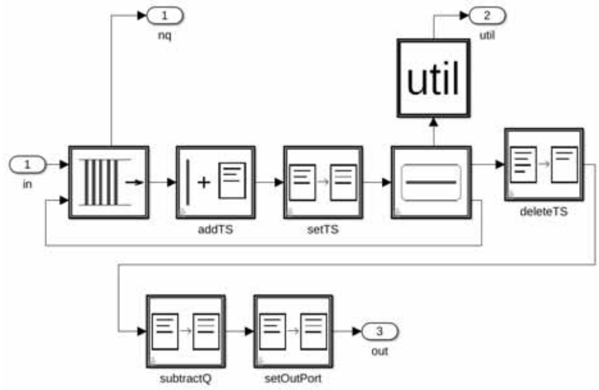


Figure 9: Coupled model `CPU`.

The complete model `timeShared` (cf. Fig. 10) shows the loop around the CPU that jobs are sent, until they have got their complete service time. The `distribute` atomic uses the `outPort` attribute to route finished jobs through the upper port.

Finally the response time is computed by subtracting the value of attribute `startTime` from the current time and the entity is terminated. The `terminator` counts the outgoing jobs and sends this value back to the coupled model `Terminals`, where new jobs are created. A `stop` atomic halts the simulation, after N_J jobs have been processed.

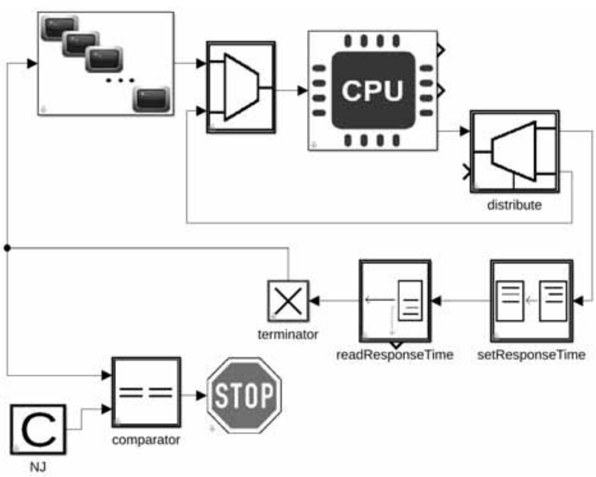


Figure 10: Complete coupled model `timeShared`.

Adding a few output blocks, one can gather enough information to get a complete picture of the model behaviour. The information displayed in Fig. 11 is sufficient to read off the waiting and service times of the jobs and follow each job individually through the model. This allows to thoroughly check the system behaviour. Especially intuitive is the plot of the remaining time after the CPU, which nicely displays the loops of the jobs around the CPU and the interaction of several jobs.

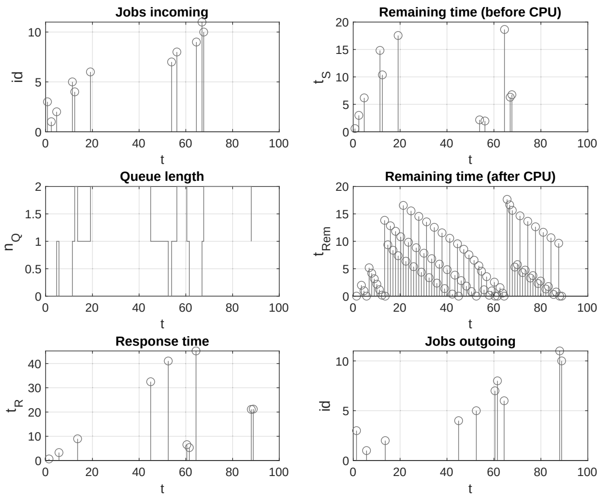


Figure 11: Detailed results of `timeShared`.

The component library contains the atomic `am_getmean` that calculates the running mean value of its input values, and the atomic `am_utilization` for the computation of the CPU utilization. Adding them to the `timeShared` model, one easily gets all requested statistical data. A typical example with $N = 20$ and $N_J = 1000$ is shown in Fig. 12. The results are similar to those of the SimEvents version of this model that had been used in [2], and consistent with the results shown in [12].

5 Conclusions

The implementation of the `timeShared` model has once again shown that NSA-DEVS is a conveniant basis for component-based modeling of discrete-event systems with a sound mathematical foundation. Especially, not one of the delay parameters had to be changed from its default value. This should be the typical case for systems with stochastical elements, where the probability of concurrent events is quite small.

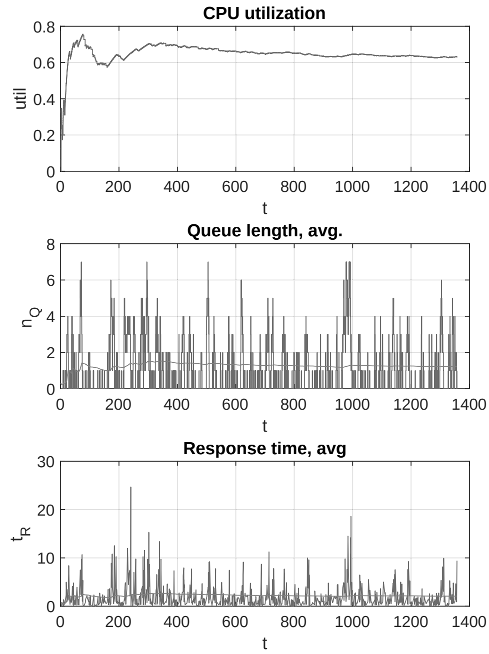


Figure 12: Statistical results of `timeShared`.

Furthermore, the toolset available freely from [11] has proven its versatility: After finding and implementing a suitable set of atomics for handling entities with variable attributes and adding some standard atomics to the library, the construction of a transaction-oriented application proceeded by standard graphical methods.

We invite all modelers interested in discrete-event modeling to try out these tools, ask for enhancements or even provide useful new atomics to the library!

During the design of the free NSA-DEVS simulator and the library, the focus has mainly been on correctness and simplicity. This shows, when measuring its performance: The simulation of `timeShared` with $N = 40$ and $N_J = 1000$ has a runtime of around 45 seconds on a recent PC platform, while the corresponding SimEvents version needs less than 2 s. Of course, the comparison is not quite fair, since SimEvents compiles the Matlab code before a run.

Nevertheless, there is definitely large potential for improvement by trading elegance of construction for runtime performance and by generally reducing the number of messages sent.

With the presented tools and methods, one can finally tackle the questions that have been raised in [2]:

> What are the shortcomings of current implementations? Which concepts or components are missing? How could a reasonable set of components be defined?

The atomic models for entity handling and the N-server introduced above are practical examples, how to precisely define fundamental building blocks due to their underlying NSA-DEVS based formulation.

Another step along these lines would be the introduction of versatile queue models that are capable of supporting all the applications denoted in the ARGESIM benchmark C22 [13]. To cite [2] once again:

> For the advancement of transaction-based modeling it is vital that it is based on a thorough theoretical analysis to reveal the fundamental abstractions and basic components that are necessary.

This is true more generally for all discrete-event based modeling. NSA-DEVS and corresponding tools seem to be a promising path to promote such a program.

Publication Remark

This contribution is the revised extended version of the conference version for

ASIM SST 2024 -
27th Symposium Simulation Technique - Munich 2024, published in
ASIM 2024 Tagungsband Langbeiträge,
ARGESIM Report 47, e-ISBN 978-3-903347-65-6,
Volume DOI 10.11128/arep.47
Article DOI 10.11128/arep.47.a4701, p 211-218.

References

[1] Dias LMS, Vieira AAC, Pereira GAB, Oliveira JA. Discrete simulation software ranking—A top list of the worldwide most popular and used tools. In: *2016 Winter Simulation Conference (WSC)*. IEEE. 2016; pp. 1060–1071.

[2] Austermann L, Junglas P, Schmidt J, Tiekmann C. Conceptional problems of transaction-based modeling and its implementation in SimEvents 4.4. *SNE Simulation Notes Europe*. 2017;27(3):137–142. doi: 10.11128/sne.27.tn.10383.

[3] Zeigler BP, Muzy A, Kofman E. *Theory of Modeling and Simulation*. San Diego: Academic Press, 3rd ed. 2019.

[4] Franceschini R, Bisgambiglia PA, Touraille L, Bisgambiglia P, Hill D. A survey of modelling and simulation software frameworks using Discrete Event System Specification. In: *Proc. of 2014 Imperial College Computing Student Workshop*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik. 2014; pp. 40–49.

[5] Preyser FJ, Heinzl B, Raich P, Kastner W. Towards Extending the Parallel-DEVS Formalism to Improve Component Modularity. In: *Proc. of ASIM-Workshop STS/GMMS*. Lippstadt. 2016; pp. 83–89.

[6] Preyser FJ, Heinzl B, Kastner W. RPDEVS: Revising the Parallel Discrete Event System Specification. In: *9th Vienna Int. Conf. Mathematical Modelling*. Wien. 2018; pp. 242–247.

[7] Junglas P. NSA-DEVS: Combining Mealy Behaviour and Causality. *SNE Simulation Notes Europe*. 2021; 31(2):73–80. doi: 10.11128/sne.31.tn.10564.

[8] Jammer D, Junglas P, Pawletta T, Pawletta S. A Simulator for NSA-DEVS in Matlab. *SNE Simulation Notes Europe*. 2023;33(4):141–148. doi: 10.11128/sne.33.sw.10661.

[9] Jammer D, Junglas P, Pawletta T, Pawletta S. Implementing Standard Examples with NSA-DEVS. *SNE Simulation Notes Europe*. 2022;32(4):195–202. doi: 10.11128/sne.32.tn.10623.

[10] Jammer D, Junglas P, Pawletta T, Pawletta S. Modeling and Simulation of a Real-world Application using NSA-DEVS. *SNE Simulation Notes Europe*. 2023; 33(4):149–156. doi: 10.11128/sne.33.tn.10652.

[11] CEA Wismar. *NSA-DEVS on GitHub*. `https://github.com/cea-wismar/NSA-DEVSforMATLAB`.

[12] Law AM. *Simulation Modeling and Analysis*. New York: McGraw-Hill, 5th ed. 2014.

[13] Junglas P, Pawletta T. Non-standard Queuing Policies: Definition of ARGESIM Benchmark C22. *SNE Simulation Notes Europe*. 2019;29(3):111–115. doi: 10.11128/sne.29.bn22.10481.

Automated Generation of Simulation Models for Production and Logistics Processes Using LLM-Based Multi-Agent Systems

Roman Krämer*, Jens Heger

Institute for Production Technology and Systems (IPTS), Leuphana University Lüneburg, Universitätsallee 1, 21335 Lüneburg, Germany; *roman.kraemer@leuphana.de

SNE 36(1), 2026, 9-14, DOI: 10.11128/sne.36.tn.10762
Sel. ASIM SST 2024 / ASIM SPL 2025 Postconf. Publ. 2025-09-30
Received Improved English: 2025-12-01; Accepted: 2025-12-20
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna
ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. Discrete-event simulation (DES) is a well-established method for analyzing and optimizing complex production and logistics systems. However, its use is often limited by high modeling effort and the need for specialized expertise. This article presents a novel multi-agent system based on Large Language Models (LLMs) that automates the creation and validation of simulation models. Building on a previous approach of ours, the new system employs an agent-based architecture designed to address issues such as context loss and the need for manual validation. Specialized agents handle tasks from requirements elicitation to results evaluation. The implementation uses open source frameworks "LangGraph" to structure agent interactions and "SimPy" to model the simulation logic. A case study demonstrates that the system can automatically and reproducibly model a complete production scenario from text-based descriptions. The results show realistic modeling and a significantly reduced modeling effort compared to both a manual approach and our previous system. The proposed approach lowers reliance on expert knowledge and makes simulation-based methods more accessible to non-specialist users.

Introduction

Growing complexity in modern production and logistics systems combined with volatile markets is driving demand for powerful planning and analysis tools. In this context, discrete event simulation has long established itself as a key method for planning and optimizing such systems.

Particularly in complex, dynamic systems with stochastic influences, where static or deterministic methods reach their limits and prove inadequate, simulation studies provide possibilities for accurate process modeling, identification of weak points and creating a sound foundation for decision-making as well as optimization measures [1].

In practice, however, their application is hampered by high modeling costs: Beyond requiring deep understanding of the specific production processes, practitioners need expertise for modeling and specialized software, most often necessitating external consultation. The resulting coordination processes are time-consuming and thus costly, often making simulations economically viable only for large scale projects. While these challenges aren't new, recent advances in the field of artificial intelligence (AI) and especially large language models (LLMs) present new opportunities for simplifying the modeling process and execution of simulation studies.

In our previous work [2], we presented a monolithic approach in which an LLM handled the entire modeling process from user interaction through code generation to output of results. While this approach successfully generated functional Python code for simulation models using the framework "SimPy", a critical limitation became apparent during the modeling process: the LLM lost focus over longer contexts, requiring numerous manual correction loops. The effort was merely shifted from model creation to code validation, which meant successful use of such an approach remained limited to experts still.

This paper extends that approach and introduces an LLM-supported multi-agent system, in which specialized agents with narrowly defined task focuses handle the automated creation of simulation models based on natural language text descriptions. To evaluate the improvements compared to the previous approach, we revisit the production scenario from our previous paper.

1 State of the Art

The rapid advancement of LLMs is driving their increasing adoption in corporate contexts. Systematic studies focusing on production environments indicate that LLMs are especially used for communication and information provisioning, predominantly as chatbots in scenarios of direct interaction with employees [3].

LLMs with added retrieval augmented generation (RAG) approaches find utilization in structured delivery of knowledge from companies' internal databases and, in some cases, serve as worker support tools for processes in production related tasks [4]. Within the domain of simulation and modeling, LLMs have been proposed and used to assist with simulation software usage by helping users navigate interfaces and provide context-specific recommendations for problem-solving [5]. Approaches for translating natural language descriptions into simulation models and automated code generation for simulation models have been investigated for general suitability using simple examples, particularly in the field of production logistics [6]. These approaches often focus on specific aspects, such as modeling of queues [7], through use of prompt engineering without further task decomposition into subtasks.

Consequently, focus drift and loss of focus during extended interactions is observed, necessitating manual correction efforts. Therefore, such approaches are only suitable for experienced users. Further research has explored approaches for automated structuring of customer requirements that can be imported into simulation environments for tasks such as layout planning [8].

LLM-based multi-agent systems for complex process automation represent an active area in current research. Of particular note is an approach for automated scientific manuscript generation [9]. One such manuscript generated by the system outlined in that approach was submitted to a scientific conference workshop and received ratings exceeding the average peer review acceptance threshold [10].

While multi-agent systems are extensively used experimentally, they have not yet explicitly found their way into the field of production and logistics simulations [11]. Approaches employing multi-agent systems for parameterizing exemplary, simplified simulation models have been described, though not within production or logistics contexts [12]. To our knowledge, no existing approach addresses the complete pipeline for generating simulation models in production and logistics contexts with a focus on accessibility for users without simulation expertise.

Facilitating the use of simulation as a methodology therefore remains an unresolved challenge.

2 Approach & Implementation

Building on the findings from our previous approach, and in particular with regard to the loss of focus, the limitations of a monolithic approach are addressed by following the example of real interdisciplinary collaboration. Therefore, both a clear division of labor and a structured validation strategy are introduced, establishing a collaborative multi-agent system. Since the system corresponds to a directed graph, "LangGraph" is used for implementation. This layer of abstraction also enables a flexible system that functions independently of a specific LLM or LLM provider. Additionally, the Python framework "SimPy" is used for implementing the simulation logic. The presented system, which automatically generates, runs and evaluates simulation models based on a natural language input description, aims to enable users without in-depth simulation knowledge to independently conduct simulations for decision support.

The agent-based architecture of the system comprises a total of five roles:

- **Agent 1 - Requirement Elicitor Agent:** Conducts a structured conversation with the user to collect all necessary requirements (e.g., entities, processes, resources, simulation goals) and summarizes them to the user. After user confirmation, no further input by the user is necessary.
- **Agent 2 - Simulation Plan Agent:** Creates a detailed simulation plan from that summary with all model components, logic, schedules, resources, error rules, and metrics as a basis for code generation.
- **Agent 3 - Code Generator Agent:** Uses the simulation plan to generate modular Python code with functions, classes and data collection mechanisms.
- **Agent 4 - Code Validator Agent:** Checks the code for correctness, logical consistency and complete implementation of requirements. Provides feedback for revision, if necessary, to the code generator, which revises the code and resends it to the code validator.
- **Agent 5 - Scenario Tester Agent:** Tests the model against the collected goals and objectives. Checks the implementation and limitations in the evaluation. If necessary, provides feedback for improvement to the code generator, which revises the code and resends it to the code validator.

Each agent works sequentially with built-in validation and confirmation steps, creating a reliable and modular pipeline for converting production system descriptions into executable SimPy simulation models (see Figure 1).

The used LLMs are "gpt-4.1-2025-04-14" from OpenAI and "claude-sonnet-4-20250514" from Anthropic, both representing the current state-of-the-art at the time of creation of this work. These models were chosen due to their availability and performance, but particularly for their large context windows of 1,047,576 tokens (approx. 750,000 words of English text) for GPT-4.1 and 200,000 tokens (approx. 150,000 words of English text) for Claude Sonnet 4 [13, 14], as maintaining contextual coherence proved to be a critical factor in the previous monolithic approach.

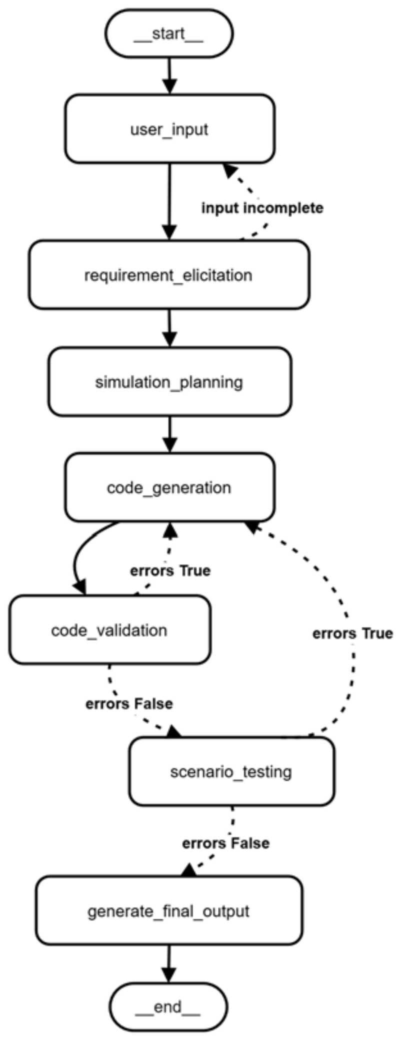


Figure 1: Structure and workflow of the multi-agent system using "LangGraph" for agent orchestration.

The experiments showed that reasoning models such as Anthropic's Claude Sonnet are well-suited for complex tasks such as code analysis and generation, but due to their reasoning capabilities, they also tend to digress when performing narrowly defined, structured tasks such as collecting explicitly defined requirements.

Based on these findings, Agent 1 uses OpenAI's non-reasoning model, while the other agents use Anthropic's reasoning model. The task and role descriptions (system prompts) for the individual agents were developed based on the procedures described in VDI 3633 Part 1 using the system's outputs for iterative refinements.

The complete implementation with instructions for setup and operation, as well as the scenario description for the initial prompt is available at the web link: https://github.com/romankraemer/SNE2025

3 Case Study

To validate and evaluate the improvements by the multi-agent system presented, we revisit the production scenario example from our previous work [2].

The scenario comprises a raw material storage, a buffer storage, a finished goods warehouse and two production machines. Two automated guided vehicles (AGVs) handle transport operations between storages and machines using a pull-based logic. At start, AGV-1 retrieves the required quantity of raw material for one product from the raw material storage and loads it into machine-1, which then begins processing.

Upon completion of this processing step, the intermediate product is transferred via a chute to the buffer storage located behind machine-1. AGV-2 collects the intermediate product, transports and loads it into machine-2. After the processing step in machine-2, the now finished product slides via a chute into the finished goods warehouse located directly behind machine-2.

At simulation start, the raw material storage contains 100 units of material. This initial quantity is chosen based on the processing times just to ensure that no material shortage occurs over the simulation period. One unit of raw material is required to manufacture one product.

The processing, handling, and transport times are modeled as deterministic, constant values in order to ensure reproducibility and traceability of the results. The time steps are shown in Table 1:

Time steps	Time in minutes
AGV-1 transport task: material storage – machine-1	2 (pickup, transport, unload, return)
AGV-2 transport task: buffer – machine-2	2 (pickup, transport, unload, return)
Processing time: machine-1	10
Processing time: machine-2	20
Simulation duration	480

Table 1: Production system properties and parameters.

As the only modification to the scenario, the AGV time steps were divided into four sub steps (pickup, transport, unload, return to start position), each with a duration of 0.5 minutes. So, the absolute cycle times remain unchanged. Notably, this was explicitly requested by Agent 1 during requirement elicitation, indicating an understanding of pull-based control in production simulation contexts and importance for clearly defined simulation states.

This scenario was deliberately designed with a bottleneck at machine-2 which has double the processing time compared to machine-1. The simulation objective is to model and track inventory levels in units across the three storages as well as the utilization of the production machines and AGVs. This data is then compared with the simulation results from our previous work.

4 Results

Following the methodology described in Chapter 2, the scenario including parameters, simulation objectives and desired output values is provided to the multi-agent system as a text description. Queries from the Requirement Elicitor Agent are addressed accordingly.

The multi-agent system captured the production system in our scenario correctly and reproducibly given the same or similar input (accounting for variations based on Agent 1's clarifying questions), generated a consistent simulation model, and executed it successfully. The results are visualized via a graphical dashboard and additionally stored in suitable data files for further use. The following diagrams are based on the output data files.

A comparison of the inventory level diagrams shows plausible results and overall consistency between the previous approach with manual correction loops during modeling (see Figure 2) and the now automated multi-agent system approach (see Figure 3).

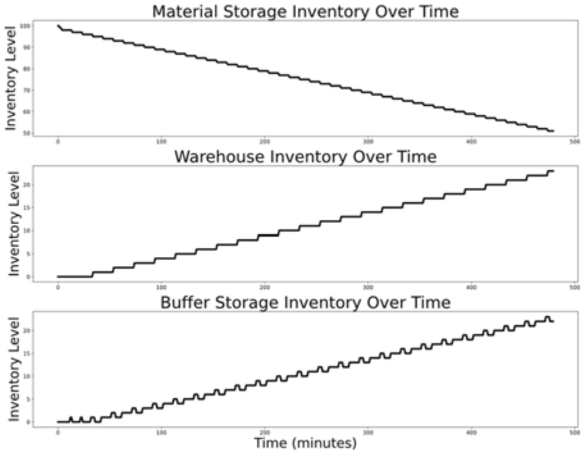


Figure 2: Inventory levels for material storage, warehouse, and buffer storage using the previous approach.

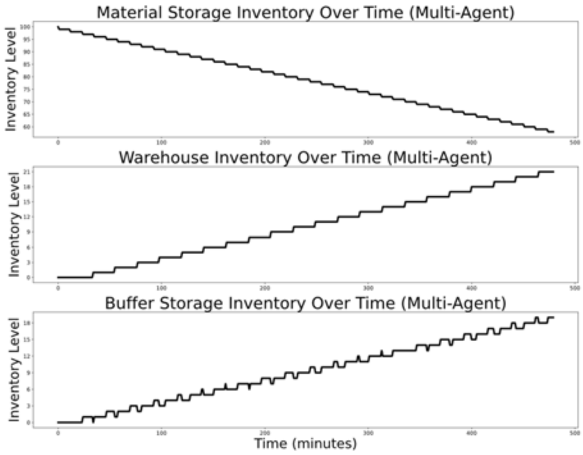


Figure 3: Inventory levels for material storage, warehouse, and buffer storage using the new approach.

However, direct comparison reveals some differences. Most notably, the production scenario achieves higher throughput with the previous approach.

While 23 products reach the finished goods warehouse in the simulation model created with the previous approach, only 21 do so with the new multi-agent system.

This difference stems from the previous system's implementation of AGVs and machines without sufficiently defined states and constraints: The AGVs effectively act as buffers, waiting with their payloads for the machines to become available. Since the transport and handling times are shorter than the processing times for both machines, the raw materials and intermediate products become available in the machines without any delay whenever the machines finish a task once the production system reaches steady state.

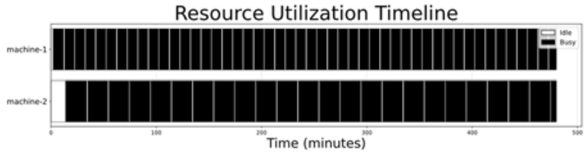


Figure 4: Machine utilization with previous approach.

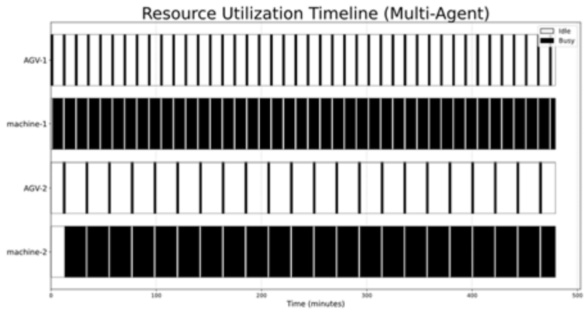


Figure 5: AGV and machine utilization with new approach.

This is evident in the machine utilization diagrams for the previous approach (cf. Figure 4).

Agent 1's system prompt specifies asking precise questions about the production scenario, including the definition of initial and final states as well as any control logic. After eliciting requirements from the user, the multi-agent system implemented a pull-based control for the AGVs, which is visible in the utilization diagram as machine idle time (cf. Figure 5).

In contrast to the previous simulation model, the pull-based control introduces realistic delays between the end of processing and the start of processing a new product due to transport and handling times. This yields lower but more realistic throughput values, which are also reflected in the utilization rates for both machines and AGVs during the simulation (cf. Table 2).

Resource	Prior approach	Multi-agent system
AGV-1	20,4% (98 min)	17,5% (84 min)
AGV-2	10,4% (50 min)	9,2% (44,2 min)
Machine-1	99,6% (478 min)	86,5% (415,2 min)
Machine-2	97,1% (466 min)	90,5% (434,4 min)
Finished products	23	21

Table 2: Utilization of individual resources in the models.

For utilization calculations of AGV-1 and AGV-2 in the previous approach, only raw travel times were used. If times during which the AGVs were waiting for the machines to become available were considered, the utilization would be closer to 100%.

In addition to creating diagrams and metrics, the new system includes a "Scenario Tester Agent". It compares the simulation model against the requirements collected by Agent 1, and after running the simulation, evaluates the results upon successful completion. If applicable, the agent issues recommendations for possible optimization actions. An exemplary recommendation for the production scenario with bottleneck added at machine-2 as used in this case study was:

"... If increased throughput is desired, improvements should focus on machine-2's processing time or capacity, as it is the limiting factor in the production system."

The system correctly identified the bottleneck at machine-2 and understood the simulation's objective, providing the user with specific recommendations for optimizing the production system.

5 Summary and Outlook

This paper demonstrates the potential of LLM-based multi-agent systems for production simulation and optimization.

Compared with our previous approach, the effort for requirements specification was significantly reduced. Now after an initial description, only a few targeted questions from the agent system were needed to generate a technical simulation plan which served as the basis to generate executable simulation models.

Novel to this approach is the complete automation of the process, from code generation, validation and execution to the analysis of simulation results and derivation of optimization recommendations. This addresses the main limitation of the prior approach, where majority of the effort was shifted from modeling to code validation as model complexity increased.

For our exemplary production scenario, an end-to-end run incurred an average cost of approx. US$0.20. This low cost, combined with the high degree of automation, shows potential for making simulation-based methods far more accessible even to users without deep domain expertise.

Nonetheless, the quality of the results remains highly dependent on the accuracy and completeness of the user-provided information. A basic understanding of key concepts and processes in modeling and simulation is still beneficial.

In the experiments conducted, it became evident that purely technical errors are easy to identify and correct. More challenging, however, is the handling and integration of domain knowledge, which often exhibits context-dependent variations or is based on experience-driven judgments. Nevertheless, the system demonstrates how simulation-based methods can be made more accessible with significantly less effort compared to manual model creation.

Acknowledgements

The project "TrICo - Transformation through Innovation and Cooperation in Communities" was funded by the Federal Ministry of Education and Research under grant number 03IHS284A. The author is responsible for the content of this publication.

Publication Remark

This contribution is the improved extended English version of the German conference versions for **ASIM SST 2024** - 27th Symposium Simulation Technique - Munich 2024 and **ASIM SPL 25** - 21. ASIM-Fachtagung Simulation in Produktion und Logistik - Dresden 2025:

Krämer R, Heger J. Beschleunigte Erstellung von Simulationsmodellen für Produktions- und Logistikprozesse mithilfe von GPT-basierten Large Language Models. ASIM 2024 Tagungsband Kurzbeiträge, ARGESIM Report 46, e-ISBN 978-3-903347-64-9, p 21-24. Vol. DOI 10.11128/arep.46

Krämer R, Heger J. Automatisierte Generierung von Simulationsmodellen für Produktions- und Logistikprozesse mithilfe LLM-basierter Multi- Agenten-Systeme. Simulation in Produktion und Logistik 2025. Techn. Univ. Dresden. DOI: 10.25368/2025.290

References

[1] Sutherland R, Özkul F, Wenzel S. Studie zum Einsatz der ereignisdiskreten Simulation in Produktion und Logistik. In Rose O, Uhlig T, editors. ASIM SST 2024 Tagungsband Langbeiträge. *27. ASIM Symposium Simulationstechnik*; 2024 Sep. Wien: ARGESIM Publisher Vienna. 185–192. DOI 10.11128/arep.47.a4716

[2] Krämer R, Heger J. Beschleunigte Erstellung von Simulationsmodellen für Produktions- und Logistikprozesse mithilfe von GPT-basierten Large Language Models. In Rose O, Uhlig T, editors. ASIM SST 2024 Tagungsband Kurzbeiträge. *27. ASIM Symposium Simulationstechnik*; 2024 Sep. Wien: ARGESIM Verlag. 21–24. DOI 10.11128/arep.46.

[3] Finkel P, Wurster P, Radler R. Large Language Models im Produktionsumfeld - Eine Potenzialanalyse zur Transformation von Produktionsabläufen in modernen Fabriken. *Industry 4.0 Science*. 2024; 40(6): 48–55. DOI 10.30844/I4SD.24.6.48.

[4] Heredia Álvaro JA, Barreda JG. An Advanced Retrieval-Augmented Generation System for Manufacturing Quality Control. *Advanced Engineering Informatics*. 2025; 64(1). DOI 10.1016/j.aei.2024.103007

[5] Anylogic. Exploring the Utility of ChatGPT as a Java Scripting Aid for AnyLogic Modeling. 2023. https://www.anylogic.de/blog/unlocking-the-power-of-simulation-modeling-in-chatgpt/. Accessed 18 Jul 2025.

[6] Jackson I, Jesus Saenz M, Ivanov D. From natural language to simulations: applying AI to automate simulation modelling of logistics systems. *International Journal of Production Research*. 2024; 62(4): 1434–1457. DOI 10.1080/00207543.2023.2276811.

[7] Obinwanne U, Feng W. A Case Study on AI to Automate Simulation Modelling. In Feng W, Rahimi N, Margapuri V, editors. *Software and Data Engineering.* Cham: Springer Nature Switzerland. 2025. 179–186. DOI 10.1007/978-3-031-75201-8_13

[8] Tinsel E-F, Lechler A, Riedel O, Verl A. Concept of an Initial Requirements-Driven Factory Layout Planning and Synthetic Expert Verification for Industrial Simulation Based on LLM. In IEEE, editor. 2024 IEEE 22nd International Conference on Industrial Informatics (INDIN). *2024 IEEE 22nd International Conference on Industrial Informatics*; 2024 Aug; Beijing, China: IEEE. 1–6. DOI 10.1109/INDIN58382.2024.10774366

[9] Lu C, Lu C, Lange RT, Foerster J, Clune J, Ha D. The AI Scientist: Towards Fully Automated Open-Ended Scientific Discovery. 2024. http://arxiv.org/pdf/2408.06292. Accessed 18 Jul 2025.

[10] Yamada Y, Lange RT, Lu C, Hu S, Lu C, Foerster J, et al. The AI Scientist-v2: Workshop-Level Automated Scientific Discovery via Agentic Tree Search. 2025. https://arxiv.org/pdf/2504.08066. Accessed 18 Jul 2025.

[11] Guo T, Chen X, Wang Y, Chang R, Pei S, Chawla NV, et al. Large Language Model Based Multi-agents: A Survey of Progress and Challenges. In Larson K, editor. Proc. of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24; 2024 Aug; Jeju, Korea: International Joint Conferences on Artificial Intelligence Organization. 8048–8057. DOI 10.24963/ijcai.2024/890

[12] Xia Y, Dittler D, Jazdi N, Chen H, Weyrich M. LLM experiments with simulation: Large Language Model Multi-Agent System for Simulation Model Parametrization in Digital Twins. In Facchinetti T, editor. *2024 IEEE 29th International Conference on Emerging Technologies and Factory Automation (ETFA)*. 2024. 1–4. DOI 10.1109/ETFA61755.2024.10710900

[13] Anthropic. Models overview. 2025. https://docs.anthropic.com/en/docs/about-claude/models/overview. Accessed 18 Jul 2025.

[14] OpenAI. Models: GPT-4.1. 2025. https://platform.openai.com/docs/models/gpt-4.1. Accessed 18 Jul 2025.

A Simple Supplier-based Market Model Offering Rich Dynamic Potential

Christian Iniotakis

Ulm University of Applied Sciences, Prittwitzstraße 10, D-89075 Ulm, Germany; *christian.iniotakis@thu.de*

SNE 36(1), 2026, 15-18, DOI: 10.11128/sne.36.sn.10763
Selected ASIM GMMS/STS 2025 Postconf. Publ.: 2025-10-22
Received Revised: 2025-11-09; Accepted: 2025-11-30
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna
ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. This work introduces a (toy) market model that relies on a simple, individual decision-making behaviour of suppliers: Which fraction of the current sales to spend for the next production? Within the model, this behaviour - together with characteristic properties of both the market and the particular cost setting - is decisive for the supplier's long-term existence on the market as well as the resulting market dynamics in general. Eventually, such a market can vanish, become stationary, exhibit market cycles or even go chaotic.

Introduction

Markets exist in an abundance of variations and are prime examples of complex systems in economy, e.g. [1, 2, 3]. As a basic feature, real markets as well as typical approaches for modelling them involve suppliers, demanders, and some type of interactions resulting in trading activities, e.g. [3].

From the simulation perspective, it is quite common to work with rather simple (toy) market models, which - despite their simplicity - still allow for qualitative insights into market behaviours, and are also helpful in the course of student education, e.g. [2, 4].

This work introduces a model which is particularly simple, while offering a suprising variety of inherent dynamic market patterns. The model only applies to some specific types of markets relying on certain assumptions and restrictions as briefly described in the following.

1 Model Assumptions

The market under consideration occurs in discrete time steps t and enables the trade of one particular, indifferentiable, commodity-like good. In other words, units from different suppliers are all valued to be of the same quality from the demanders' perspective.

Moreover, once produced all units need to be directly sold, e.g. they are perishing goods or there are only negligible buffering or storage options. All market participants are fully aware of the overall situation, such that the price P per sold unit is the same for every unit and solely depends on the total number (or quantity) Q of units sold at the market, also referred to as market size in the following.

The suppliers on this market are explicitly modelled, with each supplier n producing an individual number q_n of units. The production of each supplier is assumed to be fully scalable, just based on individual, but constant production costs c_n per unit, and without relevant dead time, i.e. any delay caused by actual production times can be neglected.

In contrast to the suppliers, the demanders are only taken into account in an aggregated way in form of a linearized demand curve, determining the resulting unit price $P(Q)$. The demand curve has a negative slope - the larger Q, the lower the unit price P - and is fully characterized by two parameters denoted as Q_{max} and P_{max} in the following. Q_{max} is the particular market size for which the unit price even drops to zero, commonly referred to as saturation quantity. The prohibitive price P_{max} is the maximum price per unit to be achieved in the limit of minimum supply $Q \to 0$.

Probably the most crucial feature of the model is how to implement the dynamical behaviour: How do suppliers decide about their next, future productions $q_n(t+1)$ based on the current market situation at time t or even the past?

For example, a rather simple, supposedly rational approach of an individual production increase or decrease - depending on the current market being profitable or not - leads to a closed simulation model with typical characteristics also found in real markets of that type, but is not in the scope here [5].

The model presented in this work, however, relies on an approach even simpler. After selling all units, a supplier just uses an individually chosen 're-investment' percentage ε_n of the sales for the production of the next units and simply takes the remainder as his or her personal profit, as sketched in Fig. 1.

This model assumption certainly is debatable, nevertheless some particular properties should be pointed out. Firstly, such a supplier accepts a profit that directly scales with the current sales, which is not as far-fetched as it might seem at first glance. Secondly, following this behaviour offers the attractive side aspect of never bearing an individual loss, at least in principle, which is rather hard to achieve on such a type of market otherwise.

Moreover, such a simple procedure of decision-making neither requires effort nor access to particular market or competitor information.

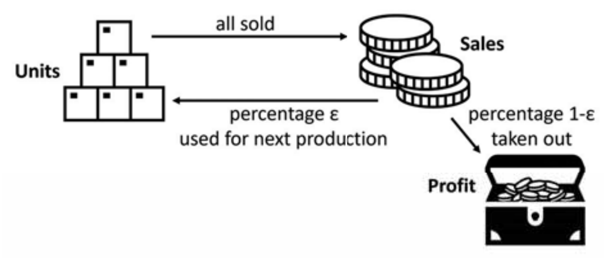


Figure 1: Sketch of simple supplier behaviour: After selling all units, a fixed percentage ε of the sales is re-used for producing the next units, while the remainder is taken out as profit.

2 Model Equations

Each supplier n starts with an initial number of units $\mathbf{q}_n(0)$. Here and in the following, quantities depicted in bold style are normalized to Q_{max} for convenience and therefore dimensionless.

The resulting dynamic behaviour is determined by the time stepping equations

$$\mathbf{q}_n(t+1) = r_n \cdot \mathbf{q}_n(t) \cdot (1 - \mathbf{Q}(t)) \tag{1}$$

together with the total market size

$$\mathbf{Q}(t) = \sum_n \mathbf{q}_n(t). \tag{2}$$

The individual parameter r_n is dimensionless, assumed to be constant over time and simply given by:

$$r_n = \varepsilon_n \cdot \frac{P_{max}}{c_n}.$$

Here, as introduced above, ε_n is the 're-investment' percentage, c_n are the production costs per unit, and P_{max} is the prohibitive price.

While the latter factor is identical for all suppliers and represents a property of market and good, the production costs depend on various aspects, such as purchasing and labour costs, the involved production technology as well as the excellence in applying it, etc.; typically, this factor is not too different between similar supplier settings.

In contrast, the factor ε_n is a fully individual parameter that can simply and freely be chosen without any restriction. In the following, the constraint $0 < r_n < 4$ is applied for all involved suppliers which ensures the total market size to stay within regular bounds $0 < \mathbf{Q} < 1$ at all times. As a consequence, the market can go on forever with both unit prices and quantities always positive.

3 Some Results

Competition

The model turns out to be highly competitive in the sense of a tough selection of suppliers surviving on the market in the long run.

As a key result, this existential issue is purely determined by the r_n-values. As long as all $r_n \leq 1$, the market will just die out completely. In any other case, it is exactly the (group of) supplier(s) with the maximum r to prevail over time. In other words, if the indiviual r_n of a given supplier is not the largest one, it inevitably means the long-term exit from the market. Some sample simulations illustrating this aspect are depicted in Figure 2.

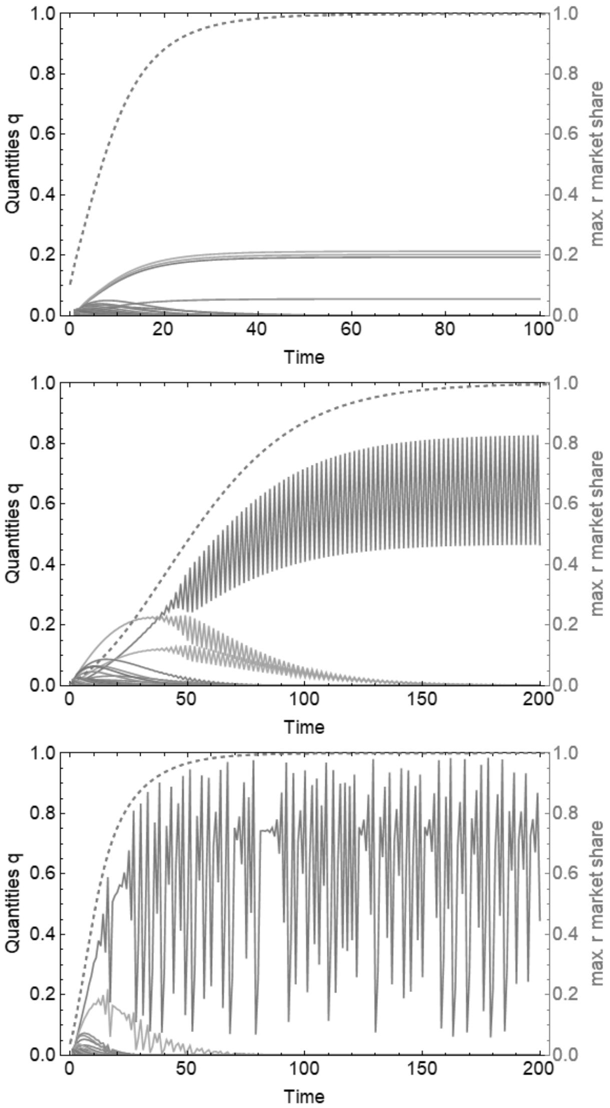


Figure 2: Dynamics in a competitive setting of 50 suppliers starting with initial quantities $\mathbf{q}_n(0)$ both random and small, as well as randomly assigned r_n. The red dashed line indicates the market share of the (group of) supplier(s) with maximum r.
(top) 4 suppliers with maximum $r = 3.0$, the market becoming stationary with $\mathbf{Q}_\infty = 2/3$.
(middle) 1 supplier with maximum $r = 3.34$, the market becoming cyclic with period 2.
(bottom) 1 supplier with maximum $r = 3.94$, the market becoming chaotic.

Long-term Market Dynamics

The model approaches a monopolistic or oligopolistic scenario in the long run. Exploiting the situation of only the supplier(s) with the (same) maximum r to be still alive and active allows to derive a simpler equation directly describing the long-term dynamics of the overall market:

$$\mathbf{Q}(t+1) = r \cdot \mathbf{Q}(t) \cdot (1 - \mathbf{Q}(t)). \qquad (3)$$

This step-wise dynamics turns out to be identical to the so-called logistic map, presumably the most prominent equation in the history of chaos, allowing for an overwhelming variety of dynamical behaviours just determined by the actual value of r [6].

As mentioned above, for $0 < r \leq 1$, no market will exist in the long run. For $1 < r \leq 3$, the system converges to a finite and stationary market of size $\mathbf{Q}_\infty = 1 - \frac{1}{r}$, as can be seen in Fig. 2, (top). For r slightly larger than 3, the potential stationary solution - in form of the fixed point mentioned above - becomes unstable, and the system shows a bifurcation leading to a dynamical pattern of period 2, cf. Fig. 2, (middle).

Generally, the interval $3 < r < 4$ contains periodic market cycles for any natural period and plenty of chaotic dynamics, as for the sample case depicted in Fig. 2, (bottom).

Analogous to [6], and for a better understanding of the overall situation, the long-term market sizes are indicated in Fig. 3 for the full range $0 < r < 4$ (top), as well as for some exemplary sub-range (bottom).

Clearly, the long-term market sizes are restricted to specific values or confined to certain bands, with rather stable islands to be located in close vicinity to chaotic regions. Thus, a small change in r can drastically affect the resulting market dynamical behaviour as typical for complex systems.

4 Further Aspects

This work shows that even simple, constant decision-making behaviours of suppliers can result in an abundance of different market dynamics. In particular, a market according to this model could even follow cycles of any period - at least in principle.

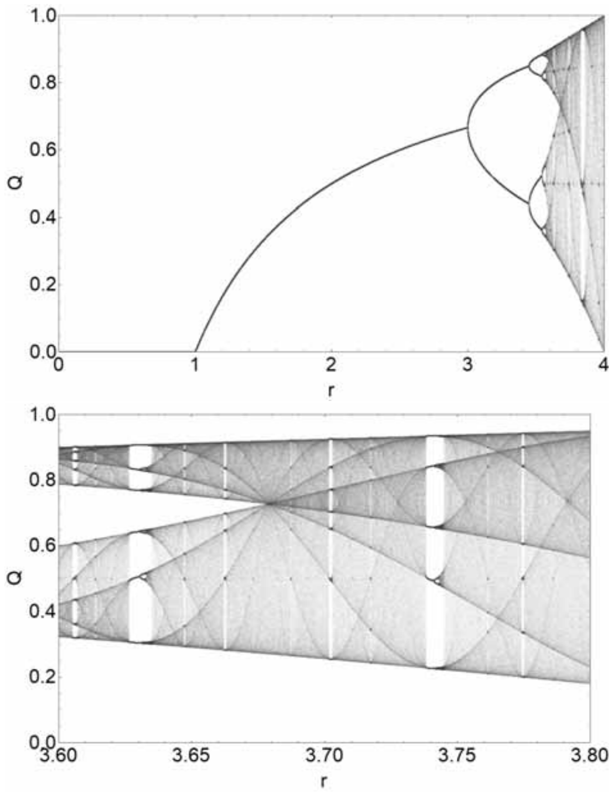


Figure 3: Long-term values for the total market size $\mathbf{Q}$ depending on r in the full relevant range **(top)** and as an exemplary zoom-in **(bottom)**, analogous to [6]. Illustrations indicate the last 100 market sizes after 10.000 time steps starting from $\mathbf{Q}(0) = 0.5$, for 20.000 different r-values along the x-axis.

This is in no contradiction to other time-related mechanisms and aspects also known to cause market cycles, such as dead times or supplier memories. Despite its simplicity and limitations, the model might help to indicate, in which markets to expect chaos - or not, cf. e.g. [7].

It should be noted, that even though this work sets the focus on total market sizes, the dynamics of the dimensionless unit price $\mathbf{P}$, normalized to P_{max}, is directly correlated via $\mathbf{P} = 1 - \mathbf{Q}$ for all times.

While a profit consideration is straightforward to achieve, e.g. [5], allowing a change of $r_n(t)$ over time is far beyond the scope of this work: On top of the inherent complexity already contained in the model for constant r_n, other layers of complexity are expected to emerge.

For example, suppliers simply adapting $\varepsilon_n(t)$ based on any strategy or investing parts of sales in cost-cutting activities in order to reduce $c_n(t)$ go along with all kinds of interactions assumed to lead to overall market patterns far from trivial.

From a supplier's perspective, there is also the immanent dilemma between the basic need for ensuring long-term existence on the market, while still somehow receiving (or even maximizing) profit.

Publication Remark

This contribution is the revised version of the Workshop proceedings version for
ASIM Workshop GMMS/ STS 2025
- Munich/Oberpfaffenhofen 2025,
published in
ASIM Workshop GMMS/STS 2025 Tagungsband
ARGESIM Report 48, e-ISBN 978-3-903347-66-3,
Volume DOI 10.11128/arep.48, p 73-76.

References

[1] Arthur WB. Foundations of complexity economics. *Nat Rev Phys*. 2021; 3: 136-145. DOI 10.1038/s42254-020-00273-3

[2] Bouchaud J-P. The (unfortunate) complexity of the economy. *Phys. World*. 2009; 22(04): 28-32. DOI 10.1088/2058-7058/22/04/39

[3] Farmer JD, Foley D. The economy needs agent-based modelling. *Nature*. 2009; 460: 685-686. DOI 10.1038/460685a

[4] Sayama H. *Introduction to the Modeling and Analysis of Complex Systems*. Open SUNY Textbooks, Milne Library; 2015. 478 p.

[5] Iniotakis C. Unpublished

[6] May RM. Simple mathematical models with very complicated dynamics. *Nature*. 1976; 261: 459-467. DOI 10.1038/261459a0

[7] Faggini M, Parziale A. More than 20 years of chaos in economics. *Mind Soc*. 2016; 15: 53-69. DOI 10.1007/s11299-015-0164-1

AGV Traffic Management: Simulation-Based Deadlock Resolution and Collision Reduction via Deep RL with PPO

Mustafa Jelibaghu[1], Michael Eley[1], Oliver Rose[2], Alexander Palatnik[1],
Leon Roth[1], Tim Thorwart[1]

[1]University of Applied Sciences Aschaffenburg, Würzburger Straße 45, 63743 Aschaffenburg, Germany;
{mustafa.jelibaghu, michael.eley, alexander.palatnik, s200991, s201070}@th-ab.de

[2] University of the Bundeswehr Munich, Werner-Heisenberg-Weg 39, 85579 Neubiberg, Germany;
oliver.rose@unibw.de

SNE 36(1), 2026, 19-27, DOI: 10.11128/sne.36.tn.10764
Selected ASIM SST 2024 Postconf. Publication: 2025-06-30
Received Revised: 2026-01-15; Accepted: 2025-02-20
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna
ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. This paper explores the application of multi-agent reinforcement learning using the Proximal Policy Optimization (PPO) algorithm for resolving deadlocks in material flow systems with Automated Guided Vehicles (AGVs). A multi-agent strategy that optimizes the dynamics and interactions of multiple AGVs in real-time is implemented. The integration of the Population Based Training (PBT) algorithm from Ray enables continuous adaptation and improvement of learning processes. Subsequent modifications to the reward system have also been implemented to enhance the model's efficiency and effectiveness. The efficacy of the proposed approach is evaluated using a material flow simulation for a real industrial use case. The results demonstrate significant improvements in reducing collisions and increasing throughput within the system. This study highlights the potential of multi-agent reinforcement learning and specifically the PPO algorithm, to enhance the performance and efficiency of material flow systems with AGVs.

Introduction

Industries increasingly rely on automated systems for production and logistics, which brings about complexity in management, particularly with deadlocks where automated guided vehicles (AGVs) block each other, halting operations.

Traditional methods to resolve these deadlocks (Xu et al. 2014; Hussain et al. 2015), such as waiting or rerouting AGVs, are inadequate for larger systems due to scalability and adaptability issues. This paper proposes that Reinforcement Learning (RL), especially Multi-Agent Reinforcement Learning (MARL), can effectively address these challenges. It focuses on the Proximal Policy Optimization (PPO) algorithm, combined with Population Based Training (PBT) and targeted reward adjustments, to develop a robust and safe system for multi-agent environments.

The paper is organized as follows: Following a comprehensive literature review in Chapter 1, Chapter 2 describes the used simulation environment, the specific challenges faced throughout the project and how they were addressed. Finally, Chapter 3 presents and discusses the obtained results, while lastly Chapter 4 outlines perspectives for future research directions and the further development of the model.

1 Literature

1.1 Reinforcement Learning

Reinforcement learning (RL) is an area of machine learning in which an agent learns to make decisions by interacting with its environment. This learning paradigm has evolved considerably since the seminal work of Sutton and Barto (1998). At its core, RL is about developing a strategy or policy that maximizes the agent's cumulative long-term reward based on the actions it takes in different states.

The application of RL in complex decision-making tasks, has shown promise in the past. The ability of RL to learn from experience and adapt to dynamic environments makes it ideal for dealing with the uncertainties and complexity associated with such systems (Zhang et al. 2021; Choi et al. 2022).

The difference between Multi-Agent Reinforcement Learning (MARL) and Single-Agent Reinforcement Learning (SARL) lies in the number of agents that learn and make decisions in the environment.

While SARL focuses on scenarios in which a single agent controls all AGVs, MARL deals with situations in which multiple agents act simultaneously.

MARL is particularly relevant to the problem of deadlocks in logistics systems, as it considers the interaction of multiple AGVs and can develop strategies for cooperative behavior to effectively avoid or resolve conflicts and deadlocks [Stone & Veloso, 2000].

1.2 Proximal Policy Optimization (PPO)

The Proximal Policy Optimization (PPO) algorithm, presented by Schulman et al. (2017), is a further development in the family of policy gradient methods in RL. PPO aims to improve the stability and efficiency of the learning process by finding a balance between the agent's exploration ability and the utilization of what has already been learned.

In contrast to its predecessors, such as the Trust Region Policy Optimization (TRPO) algorithm, PPO offers a simplified yet effective method for policy optimization, making it particularly suitable for use in dynamic and complex environments such as those found in automated logistics systems.

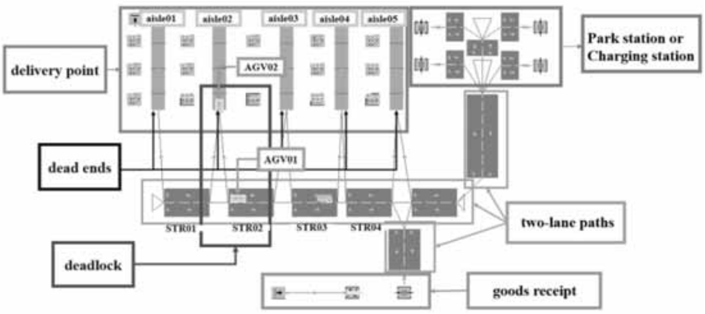


Figure 1:Illustration of a deadlock with three AGVs at the beginning of dead ends.

1.3 Population Based Training (PBT)

Population Based Training (PBT) is an approach for hyperparameter tuning that combines the advantages of genetic algorithms and hand-guided optimization. Developed by Jaderberg et al. (2017), PBT enables adaptive and time-efficient optimization of the learning processes of AI models. PBT dynamically adjusts hyperparameters as the model is trained, leading to a continuous improvement in model performance.

The main benefit of PBT lies in its ability to adapt to changing conditions within the training environment. Compared to conventional hyperparameter tuning methods, optimal configurations are achieved more effectively and quickly.

By combining these advanced techniques - MARL with PPO and dynamic adaptation through PBT - this project aims to develop a robust system for automated logistics that can avoid deadlocks while increasing the efficiency and reliability of the overall system.

2 Simulation Model

There has been a growing interest in using RL for deadlock resolution in intralogistics systems. In (Müller et al. 2022; Jelibaghu et al. 2023), the authors proposed a deadlock resolution method using a single RL agent. The agent was trained to learn how to resolve deadlocks by observing the states of the system and taking actions that lead to desired outcomes.

The agent was able to achieve high levels of deadlock resolution and collisions avoidance in (Müller et al. 2022).

As part of the research project, a real-world application for an AGV system was considered and modelled in Plant Simulation (cf. Figure 1).

The application is a high-bay warehouse with five aisles that the AGVs can only enter and exit from one side (dead ends). There are three AGVs available that have the task of moving pallets from the goods receipt, where the orders are created automatically and assigned to the AGVs (well known as dispatching), to the high rack. At the beginning the AGVs are located at the park station. A deadlock situation is shown in Figure 1. The deadlock occurs because the AGV01 currently located on the STR02 wants to enter aisle02.

At the same time, the AGV02 wants to leave aisle02. The simulation model is a digital twin of the logistics system, enabling scenario testing and performance optimization. It tracks material movement and component performance to identify bottlenecks and improvement areas.

The AGVs are controlled by an artificial neural network, which is implemented using Python code and connected to the simulation model and the AGV agents via the COM interface.

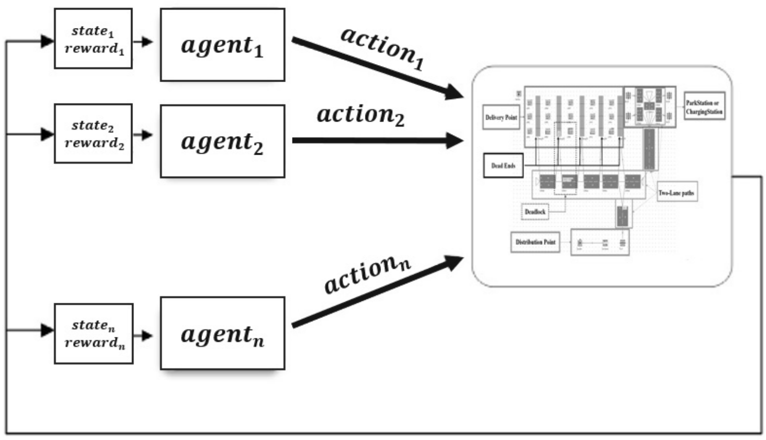


Figure 2: Agents in a multi-agent reinforcement learning (MARL).

Training configurations	
Episodes per training	300
Steps per epsiode	2000
Trials per training	5
Concurrent trials	1
Perturbation interval	5
Observation Space	
General	ID of the considered AGVs
For each agent *i*	*x* and *y* position of agent *i* Current speed of agent *i* *x* and *y* position of the target destination of agent *i*
Action Space	
Possible actions	MoveForward, MoveBackward, Stop
Reward-System	
Perform action	-1
Pick up order	+100
Complete order	+5000
Not-loaded agent exits aisle	+10
Agent enters occupied aisle	-10
Loaded agent exits aisle	-10

Table 1: Summary of initial action space and reward-system.

We decided to reduce the complexity of the environment, by reducing the number of agents to two. This adjustment aimed to provide a clearer view of how well our agent could learn and adapt, allowing for a more detailed observation.

To track the hyperparameters and the progression of rewards throughout the training, we utilized Weights & Biases (wandb).

This platform enabled us to gain insights into the performance of our model and observe the development of hyperparameters over the course of training. These adjustments resulted in clearer and more informative training progressions, allowing for more thorough analyses of the model's behavior and effectiveness.

In **Table 1** we present the initial setup of training configurations, action space and reward system, to show the baseline from which we began observing and adjusting the models performance and learning abilities.

As **Table 1** shows, at first the trials of the trainings were carried out serially and not in parallel. This had a considerable influence on the hyperparameters.

3 Experiments and Results

Our work builds on the work of (Müller et al. 2022; Jelibaghu et al. 2023). We propose a deadlock resolution method using a team of RL agents that is based on a real-world intralogistics system. We aim to evaluate our method on a number of different scenarios and demonstrate that it is able to achieve high levels of deadlock resolution and performance enhancement. In Figure 2 we show the MARL functionality.

Compared to Single Agent Reinforcement Learning (SARL), the MARL approach is a decentralized approach. This means that each AGV is considered as an independent decision maker.

In the initial phase of our study, we set out to intentionally provoke a deadlock scenario by initializing two or five agents and assigning the same drop-off location within the warehouse for the first ten orders. This approach quickly proved to be too aggressive or challenging, as it led to the AGVs consistently getting stuck. The high number of AGVs and task concentration offered little room for exploration. The resulting constant number of deadlocks highlighted the inadequacy of such a complex setup. Making it difficult to observe and understand the behavior of the reward system, the hyperparameters, and the learning efficacy of our agents.

The analysis is divided into five distinct sections to reflect the evolution of our model through various stages of adjustments.

Initially, the investigation begins with an examination of results obtained from the baseline setup, which represents our model's performance under initial conditions without any alterations to the reward system or hyperparameters tuning strategy. This foundational setup, shown in **Table 2**, is crucial for establishing a benchmark against which the impact of subsequent modifications can be measured.

After establishing a baseline understanding, the focus shifts to the outcomes following strategic adjustments to the reward system. This part dives into how modifying the reward parameters influences the learning trajectory and decision-making processes of the agents.

Hyperparameter	Search space function	Search space
Minibatch size	Randint	[4; 4000]
SGD-iterations	Randint	[3; 30]
Clip-parameter	Uniform	[0.1; 0.3]
Learning rate α	Uniform	[0.000005; 0.003]
KL-Coefficient	Uniform	[0.3; 1]
KL-target range	Uniform	[0.003; 0.03]
Discount factor γ	Uniform	[0.8; 0.9997]
GAE parameter λ	Uniform	[0.9; 1]
Value function coeff.	Uniform	[0.5; 1]
Entropy coefficient	Uniform	[0; 0.01]

Table 2: Hyperparameter search spaces and functions.

The following part of this chapter explores the effects of refining the tuning mechanism. It examines how the implementation of a more dynamic and responsive tuning approach impacts the model's performance.

Lastly, we show the ability and the potential of our model to reduce the number of collisions while increasing the number of completed orders. This gives an outlook on the capability of the model to successfully resolve deadlocks.

2.1 Baseline Setup

We begin by examining the performance of one agent (SARL) before expanding our analysis to a scenario involving two agents (MARL). To be clear, we built the following scenario. First we ran and analyzed 1 agent in the environment and then we took the same model and increased it to 2 agents. **Figure 3** shows the result of the analysis. We use the result of the single agent as a benchmark. We analyze the maximum rewards achieved during the training. This metric is indicative of the agent's ability to optimize its behavior within the environment, with the highest reward in a series of trials representing the peak efficiency reached by the agent.

Out of five trials conducted during this phase, only the best-performing trial is considered for detailed analysis. This selection criteria ensures that we capture the agent's optimal performance potential under the baseline settings (cf. **Table 1**).

In the initial graphical analysis, contrasting performance outcomes between single-agent and dual-agent setups provide insightful revelations about their operational efficiency within the simulation. The **Figure 3** refers to 1 agent and 2 agents results. 2 agent refers precisely to the accumulated reward of 2 agents. Initially, it might be assumed that the single agent would outperform due to its sole occupancy of the environment, facing no competition or interference.

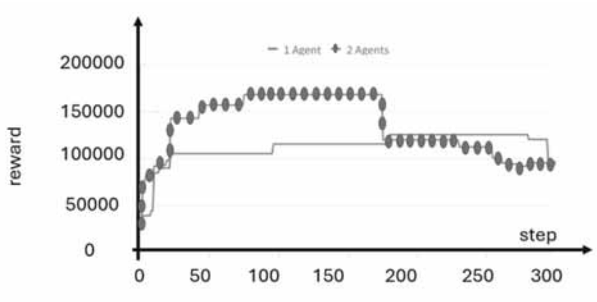


Figure 3: Max reward during training with multi agents.

However, our findings show a different picture; the dual-agent setup achieved a higher maximum reward (cf. **Figure 3**). It demonstrates the cumulative advantage of having double the number of steps available compared to a single agent. This increased action potential, combined with effective learning behavior, indeed resulted in higher rewards for the dual-agent training scenario. It emphasized the significance of collaborative efforts in navigating complex environments.

Despite having twice as many steps at their disposal, the fact that the dual agents did not achieve double the reward of a single agent can be directly attributed to the escalated complexity introduced by their interaction. This complexity presents both challenges and opportunities for learning and optimization, highlighting the delicate balance between cooperation and competition in a shared environment.

Upon closer examination across all five trials for each training setup, it becomes evident that the two-agent model not only outperforms the single-agent framework in terms of maximum rewards but also in cumulative mean reward.

This shows that the enhanced learning outcomes facilitated by multi-agent interactions are significant. The shared policy learning dynamic allows the system to benefit from a richer set of information for policy adjustments. This increases the likelihood of achieving successful, stable training outcomes.

2.2 Reward System

Table 3 shows the updated reward system, detailing the strategic modifications made to enhance navigation in the simulated environment. These changes were strategically implemented to swiftly move agents out of the aisles, thereby reducing potential deadlocks and easing traffic congestion.

This refinement entailed modifying the existing movement incentives for “*MoveForward*”, “*MoveBackward*”, and “*Stop*” actions, which are integral to the agents' navigation through the simulation environment.

The standard penalty assigned to any step taken remained set at -1, preserving the agents' motivation to minimize unnecessary movements. However, to promote more efficient behavior post-order completion, executing a “*MoveForward*” or a "*Stop*” action was assigned a larger penalty of -5, while a reward of +2 was allocated for a “*MoveBackward*” action.

Reward-System	
Perform action	-1
Unloaded AGV performs *MoveForward* on aisle	-5
Unloaded AGV performs *MoveForward* on aisl	-5
Unloaded AGV performs *MoveBackward* on aisle	+2
Pick up order	+100
Complete order	+5000
Unloaded AGV exits aisle	+25
Loaded AGV exits aisle	-10
AGV enters occupied aisle	-10
Unloaded AGV enters aisle	-50

Table 3: Adjusted reward-system.

This careful calibration of rewards and penalties was designed to maintain a balance where agents are discouraged from counterproductive movements without overshadowing the adverse implications of such actions, thereby sustaining the learning impact.

To further encourage the desired behavior, the action of successfully exiting an aisle was assigned a higher reward of +25. Conversely, to deter agents from exploiting this system by repeatedly entering and exiting the aisle without carrying a load, a hefty penalty of -50 was imposed on unloaded travel on the street. This modified incentive system was implemented to guide the agents towards behaviors that enhance overall efficiency in the logistics environment.

Figure 4 illustrates an improvement in the maximum reward output compared to **Figure 3**, which represents the results from the initial setup. This enhancement confirms that the modifications to the reward system have been effective, showing a positive impact on the overall performance.

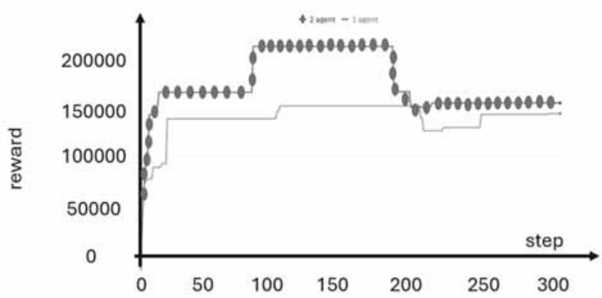


Figure 4: Max reward of training with new reward setup.

2.3 Hyperparameters

In our exploration of hyperparameters and the implementation of the tuning process, the utilization of the Population Based Training (PBT) algorithm played a pivotal role. Despite careful inclusion of all relevant parameters within the PBT tuner, along with the definition of search spaces and corresponding search functions, it was noted that several values remained at their initial setting throughout the training period. This phenomenon of no adaption applied to all hyperparameters except for the clip coefficient. Unexpectedly, the clip coefficient exhibited variations that extended beyond the predefined search space boundaries. At the start of each trial, a random value for lambda was selected from within the search space, establishing the initial conditions under which each trial would proceed.

To leverage the full capabilities of the PBT mechanisms, we adjusted the number of concurrent trials to five, allowing for simultaneous training of all trials. This setup positioned each trial as an integral component of the PBT's population, fostering a dynamic learning environment. The perturbation interval, set at every five episodes, served as a benchmark for evaluating and comparing trials.

This interval determined the feasibility of continuing a trial based on its performance. This necessitates adjustments to align with more successful trials or terminating trials that failed to show promise. It should be mentioned here that training the trials in parallel leads to significantly longer training times and to a partially incomplete recording of the courses in wandb.

This strategic approach aimed to optimize the learning process by facilitating real-time adaptations and fostering a competitive yet collaborative environment among the trials. The performance outcomes, captured in Figure 5, illustrate the impact of these adjustments on the model's learning efficiency.

Due to gaps in data collection for these parallel trials, the data of some trials was not fully tracked by wandb. Follow several reasons can be attributed: Technical Issues, System Overload, Network Interruptions, Human Error and Software Bugs. Nevertheless, the results in Figure 5 showcase clear improvement in the collected max-reward, compared to the baseline setup with serial trials (**Figure 3**). These results suggest that parallel training, despite requiring longer training durations, has a positive impact on the agents' performance. Such high-reward instances demonstrate the potential benefits of conducting training in parallel. It demonstrates that this approach can significantly enhance the learning outcomes and operational efficiency.

The primary advantage of conducting parallel trials in PBT lies in the dynamic adjustment of hyperparameters in real time. While most parameters remained constant during serial trials, the training involving parallel trials saw dynamic adaptations. This leads and enhances learning behaviour, making the training process more effective. By terminating less promising trials, the algorithm, ensures that no resources are wasted on prolonging the training time unnecessarily. Thereby, it puts the focus on the most promising configurations.

This approach allows for the rapid identification of superior hyperparameter configurations compared to manual tuning, The capability to dynamically adjusting and refining the training process in real time represents a pivotal shift towards more agile and responsive agent training methodologies.

Figure 6 provides a detailed visualization of the hyperparameter values across all trials. It is evident that many trials exhibit similar or identical hyperparameter values, which is consistent with the observation from serial trainings that hyperparameters tend to remain constant. Nevertheless, the trials that underwent dynamic adjustments generally show improved performance, reinforcing the advantage of real-time hyperparameter tuning.

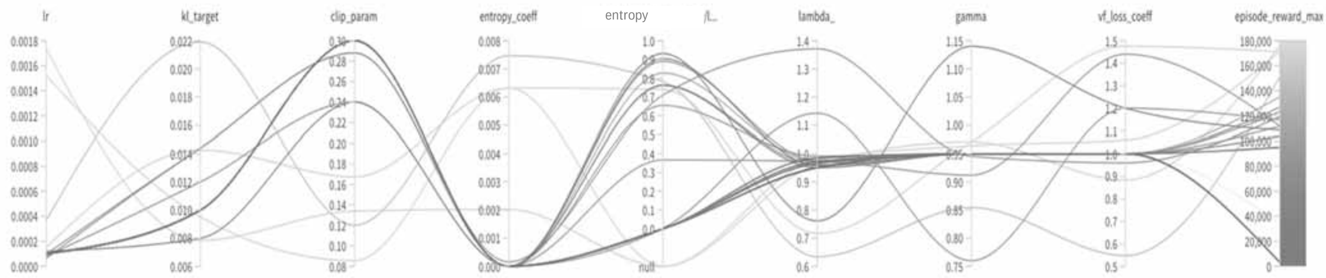


Figure 6: Hyperparameter ranges and their impact on the max reward.

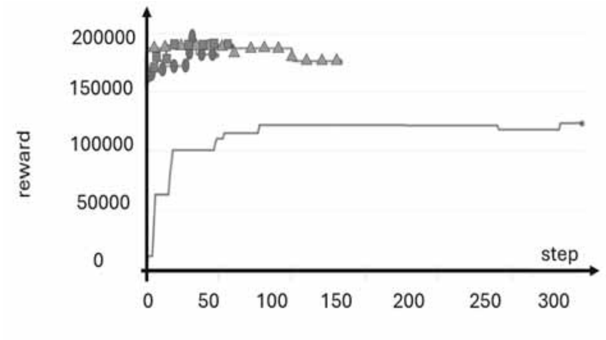


Figure 5: Max reward of training with 2 agents and concurrent trials.

However, this does not negate the possibility of achieving good performance in serial trials, as previously demonstrated.

When analysing the hyperparameters of the parallel training trials, it is essential to recognize that the values presented represent only the average due to the dynamic adjustments made during the process. From these averages, it is observable that a higher entropy coefficient, which introduces more randomness into the decision-making process, can be beneficial to the results. An increase in entropy generally encourages exploration, which can prevent the agents from getting stuck in suboptimal behaviors.

Furthermore, a slight uptick in the learning rate, paired with a reduced clip parameter, appears to be advantageous. The learning rate determines how quickly the model adapts to new information, while the clip parameter helps in stabilizing the policy update.

Adjusting these parameters could lead to a more efficient learning process by balancing the rate of adaptation with the stability of learning. However, pinpointing the exact values that yield the best performance is challenging, as a wide array of configurations have resulted in favorable outcomes. Therefore, while certain trends in hyperparameter adjustments can be identified as generally positive, the diversity in effective configurations emphasizes the complexity and adaptive nature of the learning environment. This variability shows the need for a comprehensive hyperparameter search when tailoring the model to maximize performance.

2.4 Collision avoidance

In the final section of the analysis, we demonstrate the capability of our model to not only avoid collisions but also to reduce their occurrence over the course of training, while simultaneously enhancing the throughput of completed orders.

We present three Figures to illustrate these developments: Figure 7 displays the reduction in the number of collisions throughout the training sessions.

Figure 8 shows the corresponding increase in the number of successfully completed orders. These completed orders lead to the reward shown in Figure 9.

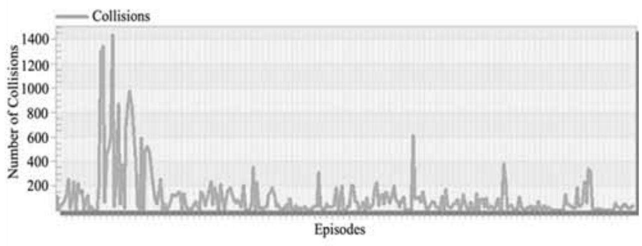


Figure 7: Number of cumulative collisions of 2 agents during training process.

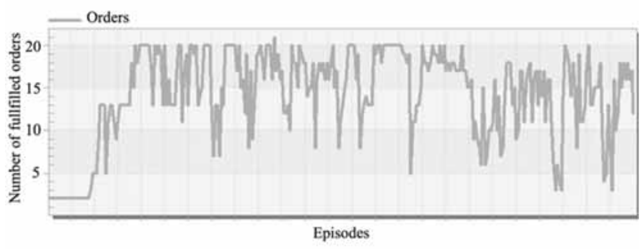


Figure 8: Number of cumulative completed orders of 2 agents during training process.

These figures clarify that the MARL approach effectively enhances operational efficiency by increasing throughput and minimizing disruptions caused by collisions.

Furthermore, these results indicate that the model possesses significant potential to avoid and resolve deadlocks, as collisions and deadlocks present very similar scenarios within our simulations.

This demonstrates substantial improvements in both safety and efficiency of the AGV-fleet.

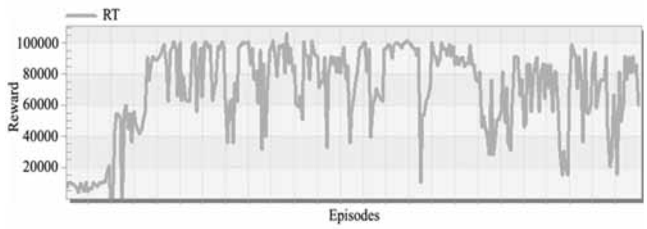


Figure 9: Accumulated reward of 2 agents during training process.

2.5 Testing phase

Eventually, we wanted to verify the policy our agents have learned over the period of training. Therefore, we established checkpoints to capture and store the policy and hyperparameter configurations learned by the agents at specific training stages. By reloading these checkpoints into a model, we were able to verify that the agents were indeed learning.

During the 100-episode learning phase the agents were still allowed to continue exploring, with the data loaded from the checkpoints serving as starting points for further learning and adaptation.

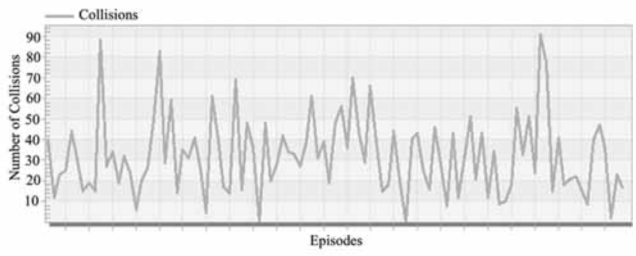


Figure 10: Number of cumulative collisions of 2 agents during testing phase.

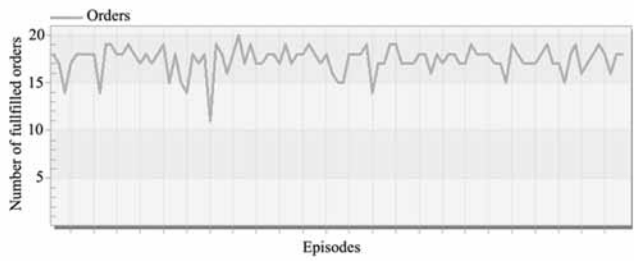


Figure 11: Number of cumulative completed orders of 2 agents during testing phase.

Figures 10 to 12 offer a clear visualization of the outcomes from the checkpoints applied during our testing phase. Figure 10 illustrates the cumulative number of collisions, which have remained consistently low from the beginning of the testing phase, suggesting that the collisions are well-managed throughout the training.

Figure 11 shows the cumulative number of completed orders, where it is evident that the orders are completed at a high rate, consistently ranking in the upper performance tiers of our training data.

Lastly, Figure 12 shows the cumulative rewards occrued, which are also high from the outset of the testing phase.

These visuals collectively demonstrate that the checkpoints contain effective policy and hyperparameter configurations, and that the agents did not require a reset or additional exploration phases to achieve these results. This underscores the robustness and efficiency of the learning mechanisms we have implemented.

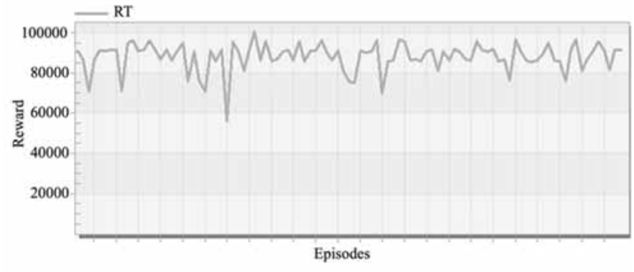


Figure 12: Sum of collected reward of 2 agents during testing phase.

3 Conclusion and Outlook

In summary, this study has demonstrated that the Proximal Policy Optimization (PPO) approach coupled with Population Based Training (PBT) for hyperparameter tuning is suitable for efficiently coordinating AGV fleets to reduce collisions and deadlocks and thus throughput to increase.

Possible extensions of the approach address the question of how to increase the number of agents. This will allow us to better understand the scalability of our approach and to refine the interplay between agents. Furthermore, we plan to experiment with different configurations of the perturbation interval, a critical mechanism in PBT that can significantly influence learning outcomes.

In parallel, further adjustments to the reward system are on the horizon. These modifications aim to enhance the learning process even more, and we intend to evaluate the impact of these changes, especially when combined with parallel trials. With these adjustments, we anticipate uncovering even richer insights into the model's performance.

While our findings have shown promise, the capabilities of our model and the process of refining its performance is ongoing. There are still many opportunities to further improve the model and gain deeper understanding, driving the evolution of AGV coordination in logistics.

Publication Remark

This contribution is the revised version of the conference version for

ASIM SST 2024 - 27th Symposium Simulation Technique - Munich 2024,

published in

ASIM 2024 Tagungsband Langbeiträge
ARGESIM Report 47, e-ISBN 978-3-903347-65-8,
Volume DOI 10.11128/arep.47
Article DOI: 10.11128/arep.47.a4705, p 203-210.

References

[1] Xu J, Zheng Z, Lyu M. CGA-based deadlock solving strategies towards vehicle sensing systems. EURASIP Journal on Wireless Communications and Networking 2014 (2014). DOI 10.1186/1687-1499-2014-214

[2] Hussain S, Kumar S, Janardh G. Deadlock avoidance and re-routing of automated guided vehicles in flexible manufacturing systems. International Journal of Advances in Production and Mechanical Engineering (IJAPME) 1 (2015) 4.

[3] Sutton RS, Bart, AG. Reinforcement Learning: An Introduction. MIT Press (1998).

[4] Zhou T, Tang D, Zhu H, Zhang Z. Multi-agent reinforcement learning for online scheduling in smart factories, Robotics and Computer-Integrated Manufacturing, Volume 72, 2021, https://www.sciencedirect.com/science/article/pii/S0736584521000855

[5] Choi H-B, et al. MARL-based Optimal Route Control in Multi-AGV Warehouses. 2022 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), Jeju Island, Korea, Republic of, 2022, pp. 333-338. DOI 10.1109/ICAIIC54071.2022.9722643

[6] Schulman J, Levine S, Abbeel P, Jordan M, Moritz P. (2017). Proximal Policy Optimization Algorithms. arXiv:1707.06347.

[7] Jaderberg M, Dalibard V, Osindero S, Czarnecki WM, Donahue J, Razavi A, Vinyals O, Green T, Dunning I, Simonyan K, Fernando C, Kavukcuoglu K. (2017). Population Based Training of Neural Networks. arXiv:1711.09846.

[8] Stone P, Veloso M. Multiagent systems: A survey from a machine learning perspective. (2000). Autonomous Robots, 8(3), 345-383.

[9] Müller M, Reggelin T, Kutsenko I, Zadek H, Reyes-Ruiano. Towards deadlock handling with machine learning in a simulation based learning environment. Proceedings of the 2022 Winter Simulation Conference 11-14 December 2022 Singapore, 2022, pp. 1485-1496.

[10] Jelibaghu M, Eley M, Palatnik A. 2023. Simulation-Based Resolution of Deadlocks in Automated Guided Vehicles using Deep Reinforcement Learning. ASIM Fachtagungen Simulation in Production and Logistics, September 2023 Technische Universität Ilmenau, Germany.

Simulation and Optimization Framework for Stochastic Resource-Constrained Project Scheduling Problems with Continuous Random Variables

Mathias Kühn, Rico Zöllner, Konrad Handrich*, Thorsten Schmidt

Institute of Material Handling and Industrial Engineering, TU Dresden, Münchner Platz 3, 01062 Dresden, Germany; *konrad.handrich@tu-dresden.de

SNE 36(1), 2026, 29-35, DOI: 10.11128/sne.36.tn.10765
Selected ASIM SST 2024 Postconf. Publication: 2025-06-30
Received Revised: 2026-01-19; Accepted: 2025-02-12
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna
ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. We introduce a simulation and optimization framework for stochastic resource-constrained project scheduling problems. The stochasticity is represented by modeling job durations as continuous random variables without time discretization, while fluctuations are captured by simulating a sufficiently large number of realizations. The objective is to analyze the relationship between variability in the input parameters and the resulting makespan in order to enable a priori estimates. Estimation approaches derived from simulations of small instances can then be extrapolated to problems involving a larger number of jobs or more complex characteristics.

Introduction

Resource-constrained project scheduling problems (RCPSP) are of widespread relevance. From a theoretical perspective, they serve as research objects for combinatorial optimization, the development of heuristics and event-discrete simulation techniques. From the practical perspective, they arise in manufacturing contexts, worker allocation schemes, medical operation and surgery planning and even in packing problems – just to mention a few. In many real-world problems, process parameters and other quantities tend to fluctuate or are only known up to certain uncertainties. Consequently, improved modeling means incorporating such aspects.

We address this by considering parameters as random variables of a given distribution instead of fixed numbers.

A review of the literature underlines the importance of RCPSPs: early attempts go back to [1, 2, 3] and recent surveys addressing the variants of such problems are for example [4, 5]. In general, heuristics and the genetic algorithm play an important role [6, 7, 8], some further state-of-the-art algorithms can be found in [9, 10, 11, 12, 13]. The works [13, 14, 15, 16] pay attention to stochastic RCPSP mainly with discrete random variables.

However, the NP-hardness of RCPSP impedes finding the exact optimum of larger problem instances within a reasonable time.

Consequently, we are therefore restricted to optimizing sufficiently small problem instances. We then extrapolate the obtained findings.

Our approach focuses on continuous random variables because many relevant processes or phenomena can be modeled by normally distributed, Gamma or Beta distributed quantities.

A stochastic RCPSP is characterized firstly by the dependencies of its inner parts (called jobs), secondly by the duration of the jobs (continuous random variables) and thirdly by the number of available resources. The aim is to minimize the makespan.

Basically, we are interested in some aspects of the systematics with special attention to stochastic influences:

- How are the distributions of the input parameters and the distribution of the objective function related?

- Regarding the reliability, is it possible to estimate the variation of the objective function using the variation of the input parameters?

To address these questions, we established a four-part simulation and analysis framework:

1. Random Number Generator: provides an appropriate set of random numbers of a prescribed distribution with expectation value and standard deviation as parameters to encode the duration of all jobs.
2. Structure Analyzer: investigates the dependencies of the jobs to find critical paths (dependencies are encoded in a directed network graph).
3. Optimizer: executes the optimization of the makespan for all stochastic realizations of the RCPSP.
4. Output Analyzer: evaluates the resulting sample of realizations w. r. t. statistical characteristic values and distribution function fits.

Our paper is organized as follows: Section 1 provides the methodical basis in terms of a precise definition of the problem class scope and short notes on the Gamma and the Beta distribution while also describing the simulation procedure. Section 2 is dedicated to selected results. We summarize in Section 3.

1 Model Setup and Simulation Approach

In this section, we explain the employed methods. First, we precisely define the scope of a stochastic RCPSP. Second, we present the framework for solving such RCPSPs. We supplement these two main part with short notes on the Gamma and the Beta distribution.

1.1 Stochastic RCPSP

To focus on the underlying principle, we refrain here from features like multiple projects, multiple modes as well as transfer times or type representatives. A stochastic RCPSP is therefore described by:

- jobs $j = 1...J$ and their respective durations $d_j \in \mathbb{R}_{\geq 0}$ considered as continuous random variables; each job starts exactly once and must not be interrupted
- successor matrix $S \in \{0,1\}^{JxJ}$ defined by

$$S_{j_1 j_2} = \begin{cases} 1 & \text{if job } j_2 \text{ follows after } j_1 \\ 0 & \text{otherwise} \end{cases} \qquad (1)$$

- number of renewable resources $R \in \mathbb{N} \cup \{\infty\}$, where each job occupies exactly one resource during its execution and $R = \infty$ refers to the unconstrained problem

In summary, a stochastic RCPSP is a tuple $(J, R, (d_j), (S_{j_1 j_2}))$ supplemented by the stochastic parameters entering the vector (d_j).

1.2 Simulation and Optimization Framework

The framework consists of the following four parts.

Random Number Generator:
To simulate stochastic RCPSPs, we generated a sufficiently large sample of realizations. For each stochastic RCPSP and each job, we keep the expected value μ_j ($j = 1...J$) of the duration d_j fixed. Furthermore, the coefficient of variation η is the same for all jobs, such that the standard deviation σ_j of the duration d_j follows from $\sigma_j = \eta \cdot \mu_j$.

In addition, we choose the type of distribution (Gamma or Beta distribution, we do not consider normally distributed random variables since they may take negative values). The sample size varies from 10,000 up to one million.

Structure Analyzer:
Dealing with continuous random variables precludes simple linear optimization models with binary decision variables (see [17, 18] for instance). Since we refrain here from a time discretization, we choose the following structural approach to incorporate the limit on the number of resources.

Let the successor matrix S be given. As a first step, the tool evaluates the potentially maximal number $R_{\max}(S)$ of resources needed such that there would be no queue (and the problem is unconstrained independently of d_j). Typically, $R_{\max}(S) > R$. As a second step, the tool collects all those successor matrices $\tilde{S}$ with $\tilde{S}_{j_1 j_2} \geq S_{j_1 j_2}$ for all j_1, j_2 and $R_{\max}(\tilde{S}) = R$. Considering $\tilde{S}$ instead of S, the RCPSP becomes unconstrained.

In summary, the idea is to solve rapidly (many) unconstrained PSPs to find the optimum of the original RCPSP. That is the key point because the main issue lies in the efficient selection of all relevant $\tilde{S}$ matrices. To handle this challenge, we strongly employ the close connection between successor matrices and posets and the knowledge of such sets (generation, isomorphism classes, structural dependencies, [19, 20, 21, 22, 23]).

Optimizer:
The task of minimizing the makespan is equivalent to finding the shortest path in a weighted directed graph (encoded in $\tilde{S}$). To this end, we employ both Gurobi and the Python package Networkx to go through the list of the relevant $\tilde{S}$ matrices. We perform this evaluation for all stochastic realizations.

Output Analyzer:
This analyzing tool processes the sample of all cycle times and computes statistical quantities, for example mean value, standard deviation, coefficient of variation, skewness. In addition, it fits the data to a given type of distribution depending on multiple parameters. Particularly interesting is the relation in terms of appropriate parametrizations between input and output quantities.

All in all, the framework is completely automatized and can be combined with a simulation framework for heuristics [24, 25, 26] and with an AI tool for finding promising fit parametrizations for the Output Analyzer. However, the results of the following section focus on stochastic fluctuations.

1.3 The Gamma and the Beta Distribution

We briefly summarize some well-known properties of these two distributions both quite convenient for modeling process times or human working durations.

Let X denote a random variable with its expectation value μ, its standard deviation σ and the dimensionless coefficient of variation $\eta = \frac{\sigma}{\mu}$. The probability density function of the Gamma distribution reads as

$$f^{(\Gamma)}_{\alpha,\beta}(x) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x} \tag{2}$$

with $x \geq 0$ (arbitrarily large values are admissible), $\alpha = \frac{\mu^2}{\sigma^2}$ and $\beta = \frac{\mu}{\sigma^2}$; analogously, we have

$$f^{(B)}_{\alpha,\beta}(x) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1} \tag{3}$$

for a Beta distribution with x ranging in [0,1] (linearly scalable to the interval $[x_{\min}, x_{\max}]$),

$$\alpha = \mu\left(\frac{\mu}{\sigma^2}(1-\mu)-1\right) \tag{4}$$

and

$$\beta = \frac{\mu}{\sigma^2}(1-\mu)^2 + \mu - 1. \tag{5}$$

The cumulative distribution functions are denoted by $F^{(\Gamma)}_{\alpha,\beta}$ and $F^{(B)}_{\alpha,\beta}$, respectively.

2 Selected Results

We picked three instances for our case study: 20 and 50 jobs for the unconstrained PSP and 10 jobs for investigating the RCPSP.

As mentioned above, a universal coefficient of variation η is fixed (see below) and the expectation value μ_j of the duration d_j is chosen randomly in the interval [0.3, 0.7].

A key result of our investigation is that we can describe the distribution of the objective function with high accuracy by a product ansatz having only two factors and four parameters even though the problem class contains many parameters.

In addition, this statement remains true for all values of the resource number R. Note that it is not sufficient to fit a simple Gamma (Beta) distribution if the input parameters are Gamma (Beta) distributed. Indeed, the second factor in the product is necessary.

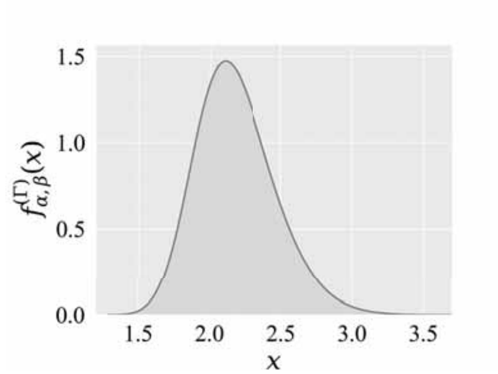


Figure 1: Histogram (blue) of the makespan x and best fits (red) according to (6) for instance with $J = 20$, Gamma distribution and $\eta = 0.3$ (sample size one million).

Let us take a closer look at the product ansatz.
We chose

$$F^{(D)}_{\alpha_1,\alpha_2,\beta_1,\beta_2}(x) = F^{(D)}_{\alpha_1,\beta_1}(x) \cdot F^{(D)}_{\alpha_2,\beta_2}(x) \tag{6}$$

with $D \in \{\Gamma, B\}$ which is inspired by the following facts:

- The sum of multiple Gamma distributed, independent random variables is Gamma distributed.

- The sum of multiple Beta distributed, independent random variables is approximately Beta distributed.
- The cumulative distribution function of the maximum of multiple, independent random variables is equivalent to the product of the cumulative distribution functions of the single random variables.

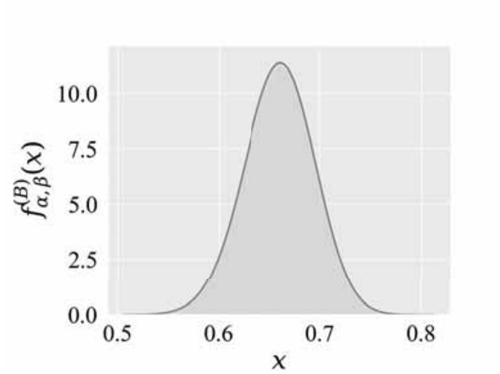


Figure 2: Histogram (blue) and best fits (red) according to (6) for instance with $J = 20$, Beta distribution, $\eta = 0.1$; the variable x is scaled to [0,1] by dividing the makespan by the longest path (sample size one million).

Here, the statements of the first two bullet points refer to the stochastic behavior along chains of jobs while the statement of the third bullet point refers to the comparison of such chains to select the critical path. However, this inspiration does not pay very much attention to the mathematical subtleties (especially the stochastic independence of the random variables).

2.1 Results of the Unconstrained PSP

We consider four samples of unconstrained problems: $J = 20$ with Gamma distribution (see Figure 1) and Beta distribution (see Figure 2), $J = 50$ with Gamma distribution (Figure 3) and Beta distribution (Figure 4). All figures display the histogram of the sample (size one million) and the best approximation according to equation (6).

The four-parameter fit exhibits a sufficiently high quality for all considered problems even though the problem class depends on far more parameters. Regarding the 50-job instances, the distribution becomes more symmetric.

As a next step, we compare dimensionless quantities. Figure 5 shows the coefficient of variation η_{out} of the objective function as a function of the coefficient of variation η_{in} of the input. Over a large number of instances, we see a stable parameter dependency.

The behavior η_{out} vs. η_{in} is close to a power law

$$\eta_{\text{out}} = a \cdot \eta_{\text{in}}^b \tag{7}$$

with $a = 0.3227 \pm 7 \cdot 10^{-4}$ and $b = 0.7874 \pm 3 \cdot 10^{-3}$ (red curve in Figure 5).

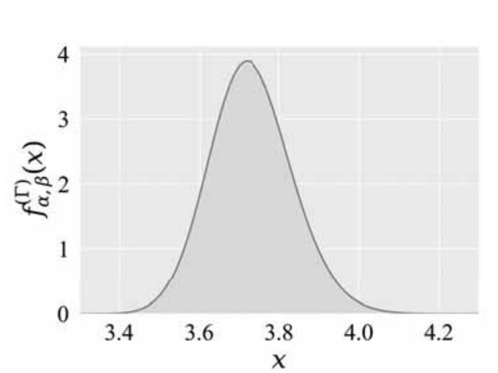


Figure 3: Histogram (blue) of the makespan x and best fits (red) according to (6) for instance with $J = 50$, Gamma distribution, $\eta = 0.3$ (sample size one million).

In addition, the objective function varies less than the input quantities. This stability in combination with the appropriate fit (6) allows an a priori estimation because the fit parameters of (6) can be estimated by a structural analysis (part of the Structure Analyzer) based on what is called the grade distribution [19, 20].

Such an estimation can be refined by a small sample. Therefore, after less runs the simulation gives already insights of the stochastic behavior of the objective function.

2.2 Results of the RCPSP

Let us now turn to the 10-job RCPSP. Mainly we are interested in two aspects:
First, does the fit ansatz (6) remain appropriate?
Second, what dependencies exist between the resource bound R and the stochastic quantities μ_{out} and η_{out}?

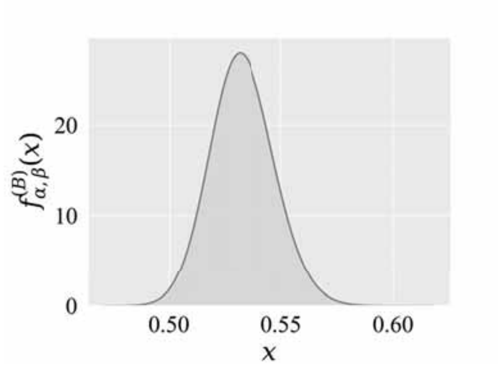


Figure 4: Histogram (blue) and best fits (red) according to (6) for instance with $J = 50$, Beta distribution, $\eta = 0.1$; x is scaled to [0,1] (sample size one million).

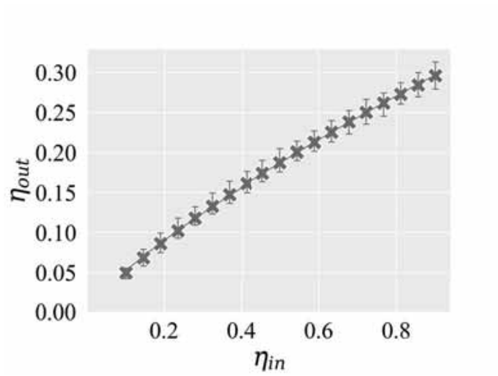


Figure 5: Coefficient of variation of the objective function η_{out} as a function of the coefficient of the variation η_{in} of the input distributions ($J = 20$, Gamma distribution, 100 instances per η_{in} value, sample size 10,000). The errorbars indicate minimum, mean (crosses) and maximum. The red curve depicts the power law regression.

Figure 6 shows the distribution (density function) of the objective function for a 10 job Gamma distributed sample for various values of R.

Note that if $R \geq R_{\max}(S)$ the problem is unconstrained (independently of R). Regarding the first question, the answer is yes (such that the histograms are not shown in Figure 6), equation (6) covers all of the following cases: for the case $R = 1$, the execution of the jobs is equivalent to a single chain. So in this special case, the second factor in equation (6) becomes obsolete.

For small values of R, the problem instance remains constrained for all possible realizations. For further increasing R, the resultant distribution contains a mixture of realizations, where the durations are such that the problem is de facto unconstrained, and realizations, where the resource constraint is indeed active.

For $R \geq R_{\max}(S)$, the problem is unconstrained for all duration vectors (d_j). Typically, $R_{\max}(S) \leq J$.

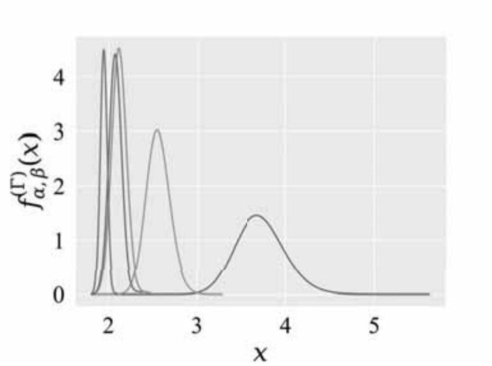


Figure 6: Distribution (density function) of the makespan for varying resource number: $R = 1$ (blue), 2 (orange), 3 (green), 4 (red) and $5 = R_{\max}$ (purple).
All cases match the fit (6). The sample size is 40,000 and $J = 10$ with Gamma distribution.

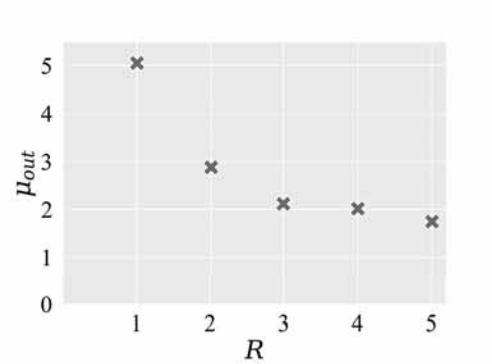


Figure 7: Dependence of expectation value μ_{out} on the resource number for a pool of Gamma and Beta distributed instances with varying μ. We have $J = 10$ and $R_{\max} = 5$ fixed such that the problem is not subject to resource limitations for $R \geq 5$.
The errorbars are not shown because they are negligible compared to the size of the crosses.

Clearly, a decreasing R causes an increasing μ_{out}. Figures 7 and 8 provide a more detailed and quantitative picture by showing μ_{out} and η_{out} as a function of R for several instances. As it is the case in Figure 5, the dependency can be captured by a rather simple regression.

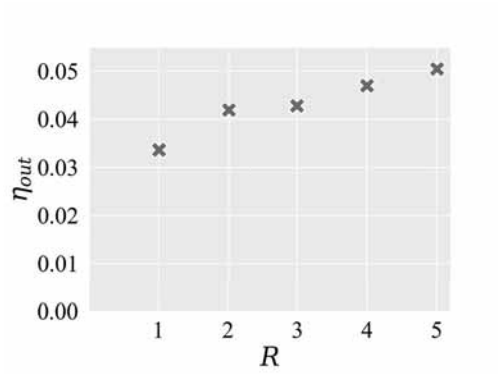


Figure 8: Dependence of the coefficient of variation η_{out} on the resource number for a pool of Gamma and Beta distributed instances with varying μ. We have $J = 10$ and $R_{\max} = 5$ fixed such that the problems is not subject to resource limitations for $R \geq 5$. The errorbars are not shown because they are negligible compering the size of the crosses.

Summarizing the previous examples, the stochastic characteristic values and the shape of the distribution of the makespan as objective function are predictable both for unconstrained and constrained PSP.

3 Summary and Outlook

The purely continuous treatment reveals close connections between the input distributions and the distribution of the makespan as objective function. Based on simulation studies, we can estimate the shape and all related stochastic properties of the resultant distribution since we found that the proposed product fit suffices for all practical purposes even though it seems hard to prove its validity by rigorous mathematical arguments.

Although solving a single realization of an RCPSP remains challenging for a larger number of jobs, considering a population significantly relaxes the situation, smooths discontinuous aspects of the combinatorial optimization problem and eventually enables statements about confidence intervals of the makespan having practical relevance.

Without aiming to explore the problem landscape too far, it seems natural to attack more sophisticated RCPSP (multi-mode and with transfer times and type representatives, for instance) with the continuous approach and to investigate the influence of these extended features on the shape of the objective function distribution.

Hereby, questions of continuous dependency and stability are of particular interest. Innovative tools in event-discrete simulation and optimization will provide a valuable contribution to this.

Publication Remark

This contribution is the revised version of the conference version for ASIM SST 2024 - 27th Symposium Simulation Technique - Munich 2024, published in
ASIM 2024 Tagungsband Langbeiträge
ARGESIM Report 47, e-ISBN 978-3-903347-67-3,
Volume DOI 10.11128/arep.47
Article DOI 10.11128/arep.47.a4710, p 167-173.

Acknowledgement

The work of Mathias Kühn and Rico Zöllner is supported by DFG project 418727532. The authors gratefully thank Ella Jannasch for her support.

References

[1] Pritsker AAB, Waiters LJ, Wolfe PM. Multiproject scheduling with limited resources: A zero-one programming approach. *Management science*. 1969;16(1):93–108.

[2] Mohanty Ru, Siddiq M. Multiple projects-multiple resources-constrained scheduling: some studies. *The International Journal of Production Research*. 1989;27(2):261–280.

[3] Kolisch R, Sprecher A, Drexl A. Characterization and generation of a general class of resource-constrained project scheduling problems. *Management science*. 1995;41(10):1693–1703.

[4] Issa S, Tu Y. A survey in the resource-constrained project and multi-project scheduling problems. *Journal of Project Management*. 2020;5(2):117–138.

[5] Hartmann S, Briskorn D. An updated survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of operational research*. 2022; 297(1):1–14.

[6] Singh A. Resource constrained multi-project scheduling with priority rules & analytic hierarchy process. *Procedia engineering*. 2014;69:725–734.

[7] Gonçalves JF, Mendes JJ, Resende MG. A genetic algorithm for the resource constrained multi-project scheduling problem. *European journal of operational research*. 2008;189(3):1171–1190.

[8] Zhu L, Lin J, Li YY, Wang ZJ. A decomposition-based multi-objective genetic programming hyper-heuristic approach for the multi-skill resource constrained project scheduling problem. *Knowledge-based systems*. 2021; 225:107099.

[9] Liu Y, Jin S, Zhou J, Hu Q. A branch-and-bound algorithm for the unit-capacity resource constrained project scheduling problem with transfer times. *Computers & Operations Research*. 2023; 151:106097.

[10] Liu Y, Zhou J, Lim A, Hu Q. A tree search heuristic for the resource constrained project scheduling problem with transfer times. *European Journal of Operational Research*. 2023;304(3):939–951.

[11] Ren Y, Lu Z, Liu X. A branch-and-bound embedded genetic algorithm for resource-constrained project scheduling problem with resource transfer time of aircraft moving assembly line. *Optimization Letters*. 2020;14(8):2161–2195.

[12] Poppenborg J, Knust S. A flow-based tabu search algorithm for the RCPSP with transfer times. *Or Spectrum*. 2016;38:305–334.

[13] Kadri RL, Boctor FF. An efficient genetic algorithm to solve the resource-constrained project scheduling problem with transfer times: The single mode case. *European Journal of Operational Research*. 2018;265(2):454–462.

[14] Hagstrom JN. Computational complexity of PERT problems. *Networks*. 1988;18(2):139–147.

[15] Deblaere F, Demeulemeester E, Herroelen W. Proactive policies for the stochastic resource-constrained project scheduling problem. *European Journal of Operational Research*. 2011; 214(2):308–316.

[16] Rostami S, Creemers S, Leus R. New strategies for stochastic resource-constrained project scheduling. *Journal of Scheduling*. 2018;21:349–365.

[17] Artigues C, Demassey S, Neron E. *Resource-constrained project scheduling: models, algorithms, extensions and applications*. John Wiley & Sons. 2013.

[18] Neumann K. Project scheduling with changeover times—Modelling and applications. In: *Proc. Int. Conf. Ind. Eng. Prod. Manage.*, vol. 1. 2003; pp. 30–36.

[19] Stanley RP. *Enumerative Combinatorics Volume 1 second edition*. 2011.

[20] Stanley RP. *Enumerative Combinatorics Volume 2*. 2001.

[21] Brinkmann G, McKay BD. Counting unlabelled topologies and transitive relations. *Journal of Integer Sequences*. 2005;8(2):3.

[22] Brinkmann G, McKay BD. Posets on up to 16 points. *Order*. 2002;19:147–179.

[23] Cordella LP, Foggia P, Sansone C, Vento M. A (sub) graph isomorphism algorithm for matching large graphs. *IEEE transactions on pattern analysis and machine intelligence*. 2004;26(10):1367–1372.

[24] Zöllner R, Kühn M, Handrich K, Schmidt T. Automated generation and simulation of hyper heuristics for stochastic multi-mode multi-project resource-constrained project scheduling problems with setup times. *20 ASIM Fachtagung Simulation in Produktion und Logistik 2023*;pp. 403–412.

[25] Handrich K, Zöllner R, Kühn M, Schmidt T. Optimum Probability Distribution versus Heuristic Probability Distribution in Continuous Resource Constrained Project Scheduling Problems. 2025;.

[26] Zöllner R, Kühn M, Handrich K, Schmidt T. Stochastic Resource-Constrained Project Scheduling Problems with Continuous Random Variables. *ASIM SST*. 2024;pp. 167–173.

Model-Based Analysis of a Diesel Generator Operating in islanded Mode under Dynamic Load Conditions

Daniel Jörss[1], Maximilian Ringel[1], Bert Buchholz[2], Christian Fink[1*]

[1]Department of Mechanical / Process and Environmental Engineering, Wismar University of Applied Sciences, Philipp-Müller-Str. 14, 23966 Wismar, Germany; *Christian.Fink@hs wismar.de

[2]Chair of Piston Machines and Internal Combustion Engines, Rostock University, Albert-Einstein-Straße 2, 18059 Rostock, Germany

SNE 36(1), 2026, 37-44, DOI: 10.11128/sne.36.tn.10766
Selected ASIM GMMS/STS 2025 Postconf. Publ.: 2025-10-22
Rec. Revised English: 26-02-11; Accepted: 2026-02-20
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna
ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. This paper presents a simulation model of a power generation unit consisting of an internal combustion engine, a synchronous generator and electrical loads. The internal combustion engine model is primarily based on thermodynamic and mechanical submodels that describe the processes occurring inside the engine. For the combustion model, a phenomenological modeling approach is applied, in which the injection profile obtained from a coupled hydraulic simulation serves as an input to the combustion model. The synchronous generator is described using a per-unit (PU) based Simscape™ model. By coupling the two models, the interaction between the internal combustion engine and the synchronous generator is simulated in order to analyze the overall behavior of the power generation unit under various operating conditions. The overall model enables a very accurate prediction of the generator behavior and serves as a basis for the optimization of operating strategies and system configuration. For validation of the simulation model, measurements were carried out on a power generator under varying load conditions. The simulation results show a high level of agreement with the measurement data under dynamic operating conditions.

Introduction

In the course of the energy transition, decentralized power generation is gaining increasing importance.

This also applies to combined heat and power (CHP) units, which simultaneously generate heat and electricity and therefore exhibit high overall efficiency. The core of such systems typically consists of internal combustion engines, which can be operated in a CO_2-neutral manner in the future due to the increasing availability of hydrogen. The operation of power generators in isolated grids under dynamic load conditions imposes high requirements on the operating behavior of the driving internal combustion engine.

In conventional configurations, where the engine is directly coupled to a synchronous generator, load fluctuations can lead to speed variations of the unit and, consequently, to fluctuations in grid frequency. Since many electrical consumers are highly sensitive to frequency deviations, it is essential to keep these within narrow limits.

Therefore, especially manufacturers of larger generator sets are required to quantify the effects of possible load scenarios on the grid frequency provided by the generator already at an early design stage.

Against this background, a simulation model was developed that enables the calculation of the operational behavior of an internal combustion engine in combination with a generator and a variable electrical load scenario based on physical principles.

The model is based on a crank-angle-resolved process calculation in order to obtain a more precise description of the system behavior and to enable optimization of the system configuration with respect to a wide range of parameters.

In this paper, the model of a power generation unit is presented and the obtained simulation results are compared with experimental data from a test bench.

1 Internal Combustion Engine Model

For the simulation environment of the internal combustion engine, a modeling approach based on established submodels was selected.

In this zero-dimensional (0D) model, the combustion chamber is considered as an ideally mixed system at any point in time, and the state variables pressure and temperature are represented as functions of time or crank angle.

Using the energy balance according to the first law of thermodynamics, the change in internal energy dU within the combustion chamber can be described as follows [1]:

$$\frac{dU}{dt} = -p\frac{dV}{dt} + \frac{dQ_B}{dt} - \frac{dQ_W}{dt} + h_I\frac{dm_I}{dt} - h_E\frac{dm_E}{dt} \quad (1)$$

The first term represents the work performed on the piston in the form of volume change work $-p\frac{dV}{dt}$. Furthermore, $\frac{dQ_B}{dt}$ denotes the heat release due to combustion, $\frac{dQ_W}{dt}$ the wall heat transfer, $h_I\frac{dm_I}{dt}$ the enthalpy flux through the intake valve, and $h_E\frac{dm_E}{dt}$ the enthalpy flux through the exhaust valve [2].

To solve the mass and energy balances, models were implemented for the heat release due to combustion (burn rate), heat transfer between the in-cylinder gas and the combustion chamber walls, the gas exchange process, and the change in internal energy within the cylinder.

The calculation of the internal energy is based on the caloric equation of state for ideal gases:

$$du = c_v \cdot dT \quad (2)$$

The specific heat capacity c_v is determined using a component-based model according to Grill [3]. In this approach, the in-cylinder gas is assumed to be a mixture of ideal gas species. A prerequisite is the prior calculation of the equilibrium composition of the in-cylinder gas as a function of temperature, air–fuel ratio, and pressure.

Based on this approach, the specific internal energy is used to determine the specific heat capacity c_v according to Equation (3) [3]:

$$c_v = \left(\frac{\partial u}{\partial T}\right)_p + \left(\frac{\partial u}{\partial p}\right)_T \cdot \frac{p}{T} \quad (3)$$

1.1 Combustion model

A key component in modeling the in-cylinder process is the combustion model. This model describes the temporal evolution of heat release from the chemical energy stored in the fuel (reaction enthalpy released during combustion).

Various modeling approaches can be used to represent the burn rate, including phenomenological or empirical burn rate models.

In the present simulation model, a combined phenomenological combustion model based on the approaches of Barba [4] and Chmela [5] is employed. The input variable for this model is the injection rate, which is calculated using an implemented injector model [6].

The calculation of the combustion process is based on two submodels: one for the pilot combustion and one for the main combustion. For the combustion of the pilot injection, a spherical propagation of the flame front originating from a single ignition point is assumed.

The flame radius is determined by the turbulent flame speed s_{turb}. The flame surface area can be expressed as follows [7]:

$$A_{Flame} = 4 \cdot \pi \cdot r_{Flame}^2 \quad (4)$$

Assuming a homogeneous air–fuel mixture, the mass burning rate for turbulent combustion originating from a single ignition point can be described by:

$$\dot{m}_{FuelBurn\ I} = k_1 \cdot \rho_{ub} \cdot s_{turb} \cdot A_{Flame} \quad (5)$$

However, it can be assumed that ignition does not occur at only one localized point in the combustion chamber, especially once the pilot combustion has progressed. Therefore, following [4,7,8], a second approach for modeling the combustion process is implemented:

$$\dot{m}_{FuelBurn\ II} = k_2 \cdot \frac{1}{\Lambda_{mc}^2} \cdot \frac{s_{turb}}{r_{mc}} \cdot m_{FuelVap} \quad (6)$$

The total burned fuel mass is obtained from the superposition of Equations (5) and (6) [8]:

$$\dot{m}_{FuelBurn} = \frac{\dot{m}_{FuelBurn\ I} \cdot \dot{m}_{FuelBurn\ II}}{\dot{m}_{FuelBurn\ I} + \dot{m}_{FuelBurn\ II}} \quad (7)$$

The combustion model for the main injection is also described using two submodels: one for premixed combustion and one for diffusion-controlled combustion.

First, the spray penetration depth S is calculated using an approach proposed by Najar [9]:

$$S_{\rho_g} = 265,7 \cdot \left(\frac{m_f}{\sqrt{Q_{hyd}}} \right)^{0,445} \cdot \rho_g^{-0,387} \tag{8}$$

Figure 1 illustrates the schematic representation of the spray for different air–fuel ratio regions in the combustion model. The region close to the nozzle represents a fuel-rich zone with limited air availability. As the spray penetrates further into the combustion chamber, the local air–fuel ratio increases accordingly.

Based on the lambda-dependent regions, a mixture volume V_{mix} can be determined in which the evaporated fuel and the amount of air defined by the local air–fuel ratio are present [5]:

$$V_{mix} = m_F \left(\frac{1}{\rho_{F,vap}} + \frac{\lambda \cdot L_{min}}{\rho_{Air}} \right) \tag{9}$$

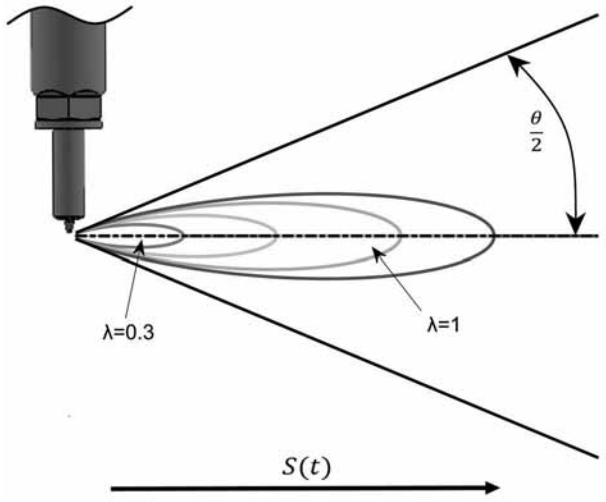


Figure 1: Schematic representation of the air–fuel ratio regions within the injection spray.

Within the model, two fuel pools are distinguished. The amount of fuel injected during the ignition delay period is assigned to the premixed combustion, whereas the fuel injected after the start of combustion is attributed to diffusion-controlled combustion.

The ignition delay is divided into a physical and a chemical part [4]:

$$\tau_{ID} = \tau_{ID,phy} + \tau_{ID,chem} \tag{10}$$

The individual components are calculated using Equations (11) and (12):

$$\tau_{ID,phy} = c_0 \cdot u_{Drop0}{}^{-1.68} \cdot d_{D.eff}{}^{0.88} \tag{11}$$

$$\tau_{ID,chem} = c_1 \cdot \left(\frac{p_{cyl}}{p_0} \right)^{c_2} \cdot \lambda_{Zn}{}^{c_3} \cdot e^{\frac{T_{Act}}{T_G}} \tag{12}$$

To account for the temporal variation of pressure and temperature during the ignition delay, the following integral formulation is applied:

$$1 = \int_{t_{SOI}}^{t_{SOC}} \frac{1}{\tau_{ID}} dt \tag{13}$$

The ignition event occurs when the integral reaches unity. After the start of combustion, the heat release rate of the premixed combustion phase is calculated according to [5]:

$$\frac{dQ_{B.Pre}}{dt} = c_{pre,1} \cdot \lambda_{pre} \cdot L_{min} \cdot e^{-\frac{c_{pre,2} \cdot T_{Act}}{T_G}} \cdot \frac{m_{f,Pre}{}^2}{V_{mix}} \cdot H_U \cdot (t - t_{SOC})^2 \tag{14}$$

The diffusion-controlled combustion is modeled using a frequency-based approach according to Barba [4] and is described by:

$$\frac{dQ_{B.Diff}}{dt} = \frac{u'}{l_{Diff}} \cdot m_{FuelVap} \cdot H_U \tag{15}$$

By introducing a characteristic mixing length l_{Diff} and the turbulent mixing velocity u', the following formulation is obtained:

$$\frac{dQ_{B.Diff}}{dt} = \frac{\sqrt{c_G \cdot c_m{}^2 + c_{Kin} \cdot k}}{\sqrt[3]{\frac{V_{cyl}}{\lambda \cdot n_{nozzle}}}} \cdot m_{FuelVap} \cdot H_U \tag{16}$$

The turbulent kinetic energy k is calculated using the following equation based on a k-ε model [10]:

$$\frac{dk}{dt} = -\frac{2}{3} \cdot \frac{k}{V_{cyl}} \cdot \frac{dV_{cyl}}{dt} - \varepsilon_{Diss} \cdot \frac{k^{1,5}}{l} + \left(\varepsilon_q \cdot \frac{k_q^{1,5}}{l} \right)_{\phi > ZOT} + \varepsilon_E \cdot \frac{dk_E}{dt} + \varepsilon_D \cdot \frac{c_m^3}{l} \tag{17}$$

It is assumed that premixed and diffusion-controlled combustion start simultaneously.

In this case, a time-delay function $F_{Pre-Diff}$ is introduced to shift the diffusion combustion rate and to account for the delayed chemical reactions governing diffusion-controlled combustion immediately after the start of combustion [8]:

$$F_{Pre-Diff} = \left(\frac{m_{FuelBurn\ pre}}{m_{FuelVap\ pre}} \right)^e \tag{18}$$

Considering this delay function, the total heat release rate is given by:

$$\frac{dQ_{B.Diff}}{dt} = \frac{dQ_{B.Pre}}{dt} + F_{Pre-Diff} \cdot \frac{dQ_{B.Diff}}{dt} \tag{19}$$

Figure 2 illustrates the individual heat release functions as well as their superposition.

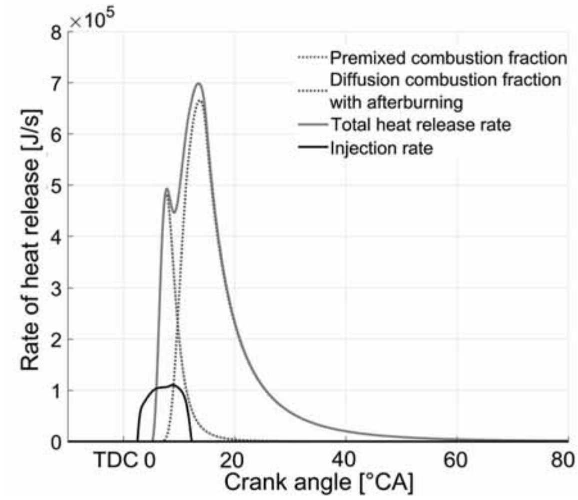


Figure 2: Modeling heat release rate.

1.2 Heat transfer model

One of the most complex submodels is the description of heat transfer between the working gas in the combustion chamber and the chamber walls.

In general, the wall heat flux in the cycle simulation is calculated using a Newtonian heat transfer approach:

$$\dot{Q}_W = \alpha \cdot A(\varphi) \cdot (T_G - T_W) \tag{20}$$

In Equation (20), A denotes the heat transfer area, which is composed of the cylinder head surface, the piston crown area, and the instantaneous lateral surface of the combustion chamber depending on the piston position.

For the calculation of the heat transfer coefficient α, a model according to Bargende is applied [11]:

$$\alpha = 253,5 \cdot V_{cyl}^{-0,073} \cdot T_m^{-0,477} \cdot p_{cyl}^{0,78} \cdot w^{0,78} \cdot \Delta \tag{21}$$

The term w represents a velocity relevant for heat transfer, while the combustion-related term Δ accounts for, among other effects, the varying temperature differences between burned and unburned gas as well as a burn-rate-dependent weighting factor X.
Both terms are calculated using Equations (22) and (23):

$$w = \frac{1}{2} \cdot \sqrt{\frac{8 \cdot k}{3} + c_K^2} \tag{22}$$

$$\Delta = \left[X \frac{T_b}{T_G} \frac{T_b - T_W}{T_G - T_W} + (1 - X) \frac{T_{ub}}{T_G} \frac{T_{ub} - T_W}{T_G - T_W} \right]^2 \tag{23}$$

In Equation (22), the specific turbulent kinetic energy k is again calculated using a k-ε model.

For the temperature calculation, a single-zone approach is applied in which no explicit separation between burned and unburned zones is performed.

According to Bargende [11], the temperature of the unburned gas T_{ub} is calculated from the start of combustion (SOC) using a polytropic relationship:

$$T_{ub} = T_{G,SOC} \cdot \left(\frac{p_{cyl}}{p_{cyl,SOC}} \right)^{\left(\frac{n-1}{n}\right)} \tag{24}$$

The temperature of the burned gas T_b is then obtained as:

$$T_b = \frac{1}{X} \cdot T_G + \frac{X-1}{X} \cdot T_{ub} \tag{25}$$

1.3 Two-zone model

In order to predict nitrogen oxide (NO) emissions, a two-zone combustion model is introduced in addition to the single-zone approach.

In this model, the combustion chamber is divided into an unburned and a burned zone, which are separated by an infinitesimally thin flame front.

A mass exchange from the unburned to the burned zone is considered using a phenomenological approach according to Kožuch [10].

According to Kožuch, the unburned zone is described by:

$$\left(\frac{V_{ub}}{V_{cyl}}\right)^{\frac{2}{3}} \cdot \frac{dQ_w}{dt} - p\frac{dV_{ub}}{dt} + \frac{dm_{ub}}{dt} \cdot (u_{ub} + R_{ub} \cdot T_{ub})$$
$$= m_{ub} \cdot \left[\frac{\partial u_{ub}}{\partial T}\frac{dT_{ub}}{dt} + \frac{\partial u_{ub}}{\partial p}\frac{dp}{dt}\right] + u_{ub} \cdot \frac{dm_{ub}}{dt} \qquad (26)$$

Analogously, the energy balance of the burned zone is given by:

$$\left(1 - \left(\frac{V_{ub}}{V_{cyl}}\right)^{\frac{2}{3}}\right) \cdot \frac{dQ_w}{dt} + \frac{dQ_B}{dt} - p\frac{dV_b}{dt} - \frac{dm_{ub}}{dt} \cdot h_{ub}$$
$$= m_b \cdot \left[\frac{\partial u_b}{\partial T}\frac{dT_b}{dt} + \frac{\partial u_b}{\partial p}\frac{dp}{dt} + \frac{\partial u_b}{\partial \lambda}\frac{d\lambda_b}{dt}\right] + u_b \cdot \frac{dm_b}{dt} \qquad (27)$$

Additionally, a mass transfer from the unburned zone into the burned zone is considered. This represents gas that bypasses the flame front and is directly mixed into the burned zone, allowing regulation of temperature and equivalence ratio.
The corresponding mixing function g is given by [10]:

$$g = c_g \cdot \rho_{ub} \cdot u_{Turb} \cdot V_b^{\frac{2}{3}} \cdot n_{nozzle} + c_{ga} \cdot \frac{dm_B}{dt} \qquad (28)$$

Together with an additional mass flow term, the total mass transfer from the unburned to the burned zone becomes:

$$\frac{dm_{ub}}{dt} = -\left[g + (1 + x_{R,st}) \cdot \lambda_F \cdot L_{st} \cdot \frac{1}{H_U} \cdot \frac{dQ_B}{dt}\right] \qquad (29)$$

Figure 3 illustrates the calculated temperature curves of the unburned and burned zones in comparison with the mass-averaged temperature obtained from the single-zone model.

Based on the calculated temperature history of the burned zone and the equilibrium composition of the species, the NO concentration is determined using chemical kinetics.

The calculation is based on the extended Zeldovich mechanism [12][13], which describes NO formation as:

$$\frac{d[NO]}{dt} = k_{1,v}[O][N_2] + k_{2,v}[N][O_2] + k_{3,v}[N][OH]$$
$$- k_{1,r}[NO][N] - k_{2,r}[NO][O] - k_{3,r}[NO][H] \qquad (30)$$

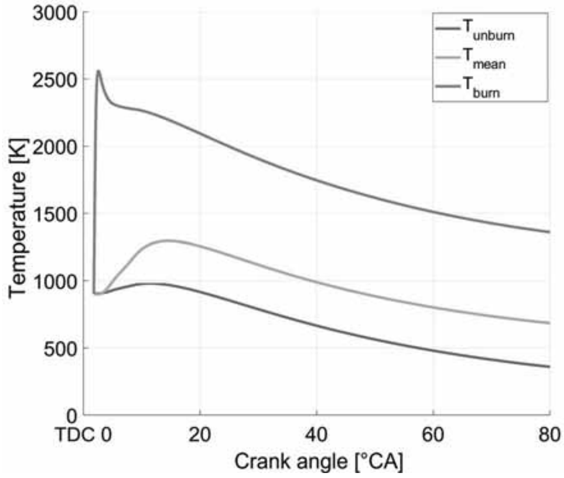


Figure 3: Temperature profiles of burned and unburned zones.

2 Turbocharger Model

In exhaust gas turbocharging, the residual exhaust gas energy available after the power stroke is utilized to increase the engine's power density. The turbine is driven by the exhaust gas enthalpy flux, which passes radially through the turbine and sets the turbine wheel and shaft into rotational motion.

This motion is transmitted to the compressor wheel, thereby supplying the internal combustion engine with an increased amount of fresh air. Considering the power balance between turbine and compressor under steady-state conditions and neglecting mechanical losses, the turbine power P_T converted from exhaust gas enthalpy equals the compressor power P_C :

$$P_T = P_C \qquad (31)$$

The power of the turbine and the compressor is determined by the mass flow rate $\dot{m}_i$, the specific isentropic enthalpy difference $\Delta h_{i,is}$, and the isentropic efficiency $\eta_{i,is}$ [14]:

$$P_T = \dot{m}_T \cdot \Delta h_{T,is} \cdot \eta_{T,is} \qquad (32)$$
$$P_C = \dot{m}_V \cdot \Delta h_{V,is} \cdot \frac{1}{\eta_{V,is}} \qquad (33)$$

The dynamic behavior of the turbocharger is described by the power balance between turbine and compressor.

From the power difference, the rotational inertia of the rotating assembly J_{TC}, and the mechanical efficiency η_m, the change in angular velocity ω_{TC} (or rotational speed) of the turbocharger (TC) can be derived. Applying the conservation of rotational energy yields:

$$\frac{d\omega_{TC}}{dt} = \frac{P_T \cdot \eta_m - P_V}{J_{TC} \cdot \omega_{TC}} \tag{34}$$

3 Synchronous Generator

For the simulation of the generator, a per-unit (PU) based synchronous generator model from the Simscape^{TM} library is employed [15]. The per-unit system normalizes all relevant system parameters to defined base values, resulting in dimensionless quantities. Since most generator manufacturers provide their machine parameters in per-unit notation, this approach allows for a realistic parameterization of the simulation model.

The input variable of the generator model is the rotational speed of the internal combustion engine. The output variable is the electromagnetic torque, which acts as the load torque on the combustion engine.

4 Model Validation

To evaluate the transient load behavior, the developed simulation model was validated using experimental data obtained from a test bench.

Figure 4 shows the transient response of the simulation compared to measurements for a load step from 2 kW to 13 kW. The load step is applied at a simulation time of three seconds. As a consequence of the load increase, the engine speed and thus the generator frequency decrease. The speed controller of the internal combustion engine responds by increasing the injected fuel mass, which raises the indicated torque to compensate for the increased power demand. At a simulation time of approximately seven seconds, the nominal frequency of 50 Hz is reached again and remains stable thereafter.

Overall, the simulated frequency response shows very good agreement with the experimental measurements. The gradient of the measured torque curve is slightly lower than that predicted by the simulation; however, the overall torque behavior matches the simulation results well.

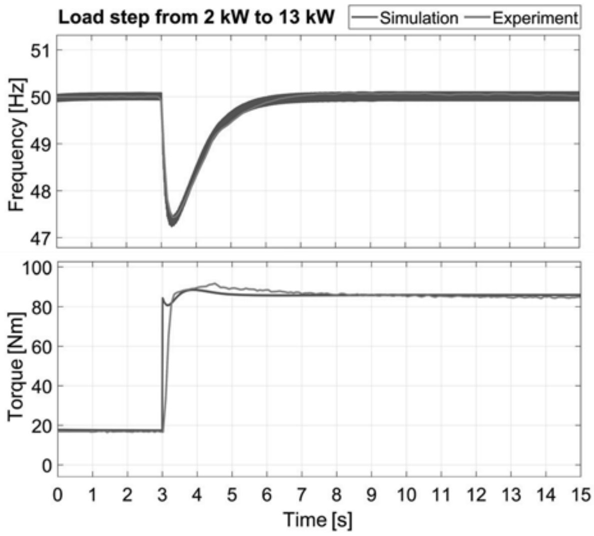


Figure 4: Simulated and measured frequency and torque response during a load step.

For further validation of the model under this load scenario, the simulation results were compared with experimental data obtained from a diesel engine test bench. Figure 5 shows a comparison of the in-cylinder pressure traces during the high-pressure phase for the load points of 2 kW and 13 kW.

In addition, the corresponding heat release rates were derived from the measured pressure traces in order to enable a direct comparison with the simulation results. The comparison of the calculated cylinder pressure and heat release curves shows very good agreement between simulation and measurement.

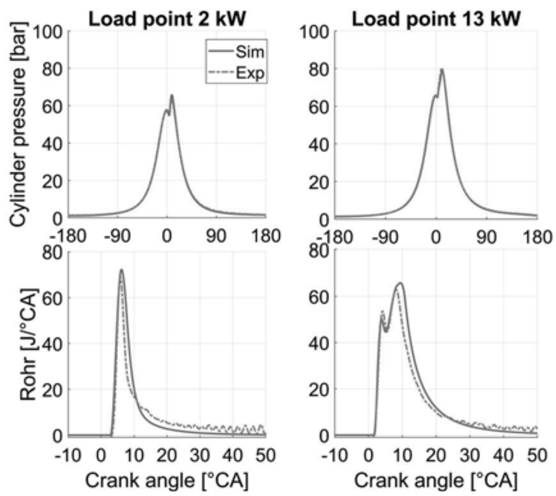


Figure 5: Comparison of in-cylinder pressure and heat release rate at 2 kW and 13 kW load points.

Figure 6 illustrates the dynamic behavior of the turbocharger during the load step using the parameters exhaust pressure, boost pressure, and turbocharger speed. The increase in injected fuel mass during the load step leads to a rise in exhaust gas enthalpy, which increases the pressure in the exhaust manifold and results in a higher turbine power output.

Consequently, the turbocharger speed increases and, due to the higher air mass flow through the compressor, the boost pressure in the intake manifold rises.

The comparison with the experimental data shows that the simulated exhaust pressure and turbocharger speed exhibit a high level of agreement with the measured curves in terms of their dynamic behavior.

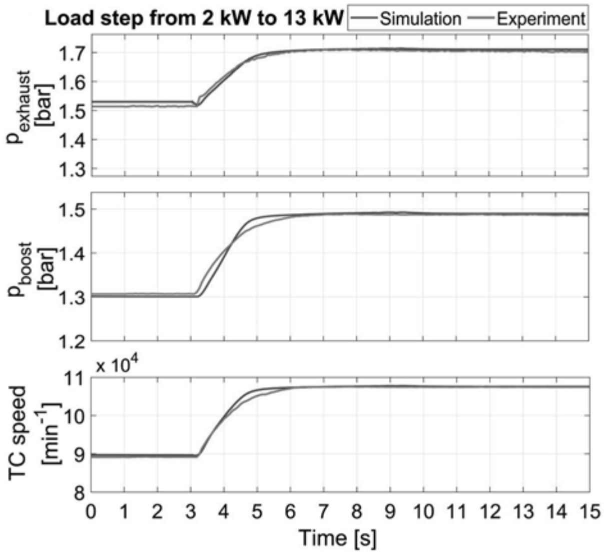


Figure 6: Exemplary transient behavior of the turbocharger (TC).

5 Summary

The modeling approach presented in this paper enables a comprehensive simulation of the operational behavior of power generation units operating in isolated grids with dynamic load conditions using a highly time-resolved process simulation.

In contrast to conventional modeling approaches, the combustion model presented here allows the simulation of multiple injection strategies due to its phenomenological formulation. In addition, the implemented two-zone model enables the prediction of nitrogen oxide emissions within the simulation framework.

The comparison of simulation results with experimental measurements from a generator test bench demonstrates very good agreement, confirming the applicability of the proposed modeling approach.

Future work will focus on investigating the influence of different fuel properties and their impact on the operational behavior of power generation units under dynamic operating conditions.

Acknowledgments

The results presented in this paper were obtained within the framework of the publicly funded project SIDYN (FKZ: 13FH043PX8) and the doctoral program of Wismar University of Applied Sciences. The authors gratefully acknowledge the financial support provided by the German Federal Ministry of Education and Research (BMBF) and Wismar University of Applied Sciences.

Publication Remark

This contribution is the revised Englisch version of the Workshop proceedings German version for **ASIM GMMS/STS 2025** - ASIM Workshop GMMS/STS 2025 - Munich/Oberpfaffenhofen 2025, published in
ASIM Workshop GMMS/STS 2025 Tagungsband
ARGESIM Report 48, e-ISBN 978-3-903347-66-3,
Volume DOI 10.11128/arep.48, p 13-20.

Nomenclature

ε_{Diss}	Dissipation constant [-]
ε_{D}	Swirl constant [-]
ε_{E}	Injection constant [-]
ε_{q}	Squish flow constant [-]
λ	Air–fuel equivalence ratio [-]
Λ_{mc}	Air–fuel ratio mixture cloud [-]
λ_{Zn}	Local air–fuel equivalence ratio [-]
$\rho_{F,vap}$	Fuel vapor density [kg/m^3]
ρ_{g}	Gas density [kg/m^3]
τ_{ID}	Ignition delay [s]
$c_{0,1}$	General constant [-]
c_{K}	Piston speed [m/s]
c_{m}	Mean piston speed [m/s]
c_{pre}	Calibration parameter [m^3/(kg·s^3)]
$d_{D.eff}$	Effective nozzle diameter [mm]
e	Exponent [-]
h	Specific enthalpy [J/kg]
H_{U}	Lower heating value [J/kg]

k	Turbulent kinetic energy [m^2/s^2]
$k_{1,2}$	Calibration constants [-]
$k_{i,v,r}$	Reaction rate constant [$m^3/(mol \cdot s)$]
l_{Diff}	Mixing length [m]
l_l	Characteristic length [m]
L_{st}	Stoichiometric air requirement [kg/kg]
$m_{FuelVap}$	Evaporated fuel mass [kg]
m_f	Cumulative fuel mass [kg]
n	Polytropic exponent [-]
n_{Nozzle}	Number of injector holes [-]
p_0	Ambient pressure [bar]
p_{cyl}	Cylinder pressure [bar]
p_{boost}	Boost pressure [bar]
$p_{exhaust}$	Exhaust pressure [bar]
Q_B	Released heat (combustion) [J]
Q_{hyd}	Hydraulic injector flow rate [cm^3/30 s at 100 bar]
R	Specific gas constant [J/(kg·K)]
r_{mc}	Radius mixture cloud [m]
$Rohr$	Rate of heat release [J/°CA]
S_{ρ_g}	Spray penetration length at a gas density [mm]
T_{Act}	Activation temperature [K]
$T_{G,m}$	Mean in-cylinder gas temperature [K]
t_{SOC}	Start of combustion [s]
t_{SOI}	Start of injection [s]
T_W	Wall temperature [K]
u	Specific internal energy [J/kg]
u_{Drop0}	Initial droplet velocity [m/s]
u_{Turb}	Turbulent velocity [m/s]
V_{cyl}	Cylinder volume [m^3]
$x_{R,st}$	Stoichiometric residual gas fraction [-]

References

[1] Pischinger R, Klell ., Sams T. *Thermodynamik der Verbrennungskraftmaschine* 3.Auflage, Wien, SpringerWienNewYork, 2009

[2] Tschöke H, Mollenhauer K, Maier R. *Handbuch Dieselmotoren* 4.Auflage, Wiesbaden, Springer Vieweg, 2018

[3] Grill M. *Objektorientierte Prozessrechnung von Verbrennungsmotoren*, Stuttgart, University of Stuttgart, Ph.D. thesis, 2006

[4] Barba C. *Erarbeitung von Verbrennungskennwerten aus Indizierdaten zur verbesserten Prognose und rechnerischen Simulation des Verbrennungsablaufes bei Pkw-DE-Dieselmotoren mit Common-Rail Einspritzung*, Zürich, ETH Zürich, Ph.D. thesis, 2001

[5] Pirker G, Chmela F, Wimmer A. *ROHR Simulation for DI Diesel Engines Based on Sequential Combustion Mechanisms*, SAE Paper 2006-01-0654, 2006

[6] Jörss D, Ringel M, Buchholz B, Fink C. *Gekoppelte Simulation des Einspritz- und Verbrennungsvorgangs eines Industrie-Dieselmotors*, ASIM Workshop 2023 STS / GMMS / EDU, Magdeburg, 2023, DOI: 10.11128/arep21

[7] Rether D, Grill M, Schmid A, Bargende M. *Quasi-Dimensional Modeling of CI-Combustion with Multiple Pilot- and Post Injections*, SAE Paper 2010-01-0150, 2010

[8] Warth M. *Comparative investigation of mathematical methods for modeling and optimization of Common-Rail DI diesel engines*, Zürich, ETH Zürich, Ph.D. thesis, 2005

[9] Najar I, Fink C, Harndorf H, Pinker F, Buchholz B. *"Anwendungsorientierte Modelle zur Berechnung von Diesel-Sprays" in 10. Tagung Diesel- und Benzindirekteinspritzung 2016*, Wiesbaden, Springer Vieweg, 2017

[10] Kožuch, P. *Ein phänomenologisches Modell zur kombinierten Stickoxid- und Rußberechnung bei direkteinspritzenden Dieselmotoren*, Stuttgart, University of Stuttgart, Ph.D. thesis, 2004

[11] Bargende M. *Ein Gleichungsansatz zur Berechnung der instationären Wandwärmeverluste im Hochdruckteil von Ottomotoren*, Darmstadt, Technical University of Darmstadt, Ph.D. thesis, 1991

[12] Zeldovich YA. *The Oxidation of Nitrogen in Combustion and Explosions*, Acta Physicochimica, USSR, Vol.21 (577-628), 1946

[13] Lavoie GA, Heywood JB, Keck JC. *Experimental and theoretical study of nitric oxide formation in internal combustion engines*, Combustion science and technology 1970, Vol.1 (313–326)

[14] Pucher H, Zinner K. *Aufladung von Verbrennungsmotoren* 4.Auflage, Berlin Heidelberg, Springer Vieweg, 2012

[15] The MathWorks, Inc. *https://de.mathworks.com/help/sps/powersys/ref/synchronousmachinepufundamental.html*, accessed: January 14, 2025.

Forecast-Based MPC of a Hydrogen-Integrated Multi-Energy System with Dynamic Building Modeling

Samuel Peifer*, Peter Renze

Institute for Energy Technology and Energy Economics, University of Applied Science, Albert-Einstein-Allee 55, 89081 Ulm, Germany; *samuel.peifer@thu.de

SNE 36(1), 2026, 45-53, DOI: 10.11128/sne.36.tn.10767
Selected ASIM GMMS/STS 2025 Postconf. Publ.: 2025-10-22
Revised English: 26-01-2026; Accepted: 2026-02-15
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna
ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. This paper presents a system-level modeling and control framework for a small-scale power-to-X (P-t-X) facility coupled with a dynamic building model. The considered system configuration and parameterization are based on a real-world energy research facility at the Technische Hochschule Ulm. A forecast-based model predictive control (MPC) strategy is compared to a conventional rule-based approach. The MPC is formulated as a mixed-integer linear program (MILP), integrates time series forecasts, and limits daily start events of key conversion units via an ε-constraint. Results indicate the potential of predictive operation to improve self-sufficiency and self-consumption while respecting operational and building comfort constraints.

Introduction

Hydrogen produced from renewable electricity is expected to play a key role in a future climate-neutral energy system [1]. The fluctuating nature of renewable generation leads to temporal mismatches between supply and demand, requiring flexible storage and operation strategies. In this context, the combination of hydrogen and battery energy storage is a promising solution [2].

Sector coupling and waste heat utilization further increase system flexibility, particularly when power-to-X (P-t-X) systems are integrated with building thermal dynamics.

However, suitable operation strategies remain challenging, especially under operational constraints such as limited start events of conversion units.

This paper presents a system-level modeling and control framework for a small-scale P-t-X facility coupled with a dynamic building model. A rule-based operation strategy is compared with a forecast-based model predictive control (MPC) approach formulated as a mixed-integer linear program (MILP).

The MPC integrates time series forecasts and limits electrolyzer and gas turbine starts via an ε-constraint, aiming to improve self-sufficiency and self-consumption while respecting operational and comfort constraints.

The remainder of the paper is structured as follows: Section 1 introduces the system, Section 2 the data basis, and Section 3 the building model. Section 5 presents the MPC formulation, followed by results and conclusions.

1 System Description

The considered system is an electrically and thermally integrated power-to-gas (PtG) and power-to-power (PtP) infrastructure coupled to a building energy system.

As shown in Figure 1, electrical energy is supplied by photovoltaic generation and the grid and consumed by the building load, an electrolyzer, a hydrogen driven gas turbine, and a battery energy storage system (BESS).

Electrical surpluses are converted to hydrogen via the electrolyzer and stored in a hydrogen energy storage system (HESS), while deficits are covered by reconversion through the gas turbine or grid import.

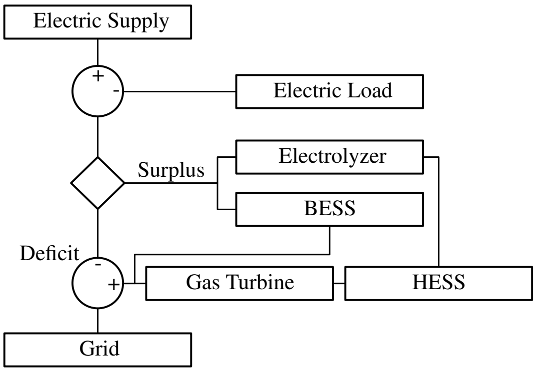


Figure 1: System model of the P-t-G / P-t-P infrastructure [3].

Thermal integration is realized by waste heat recovery from the gas turbine and an electrically driven heat pump (Figure 2).

Thermal energy is used to supply the building demand or charge a sensible thermal energy storage (TES), which provides additional flexibility for balancing heat supply and demand.

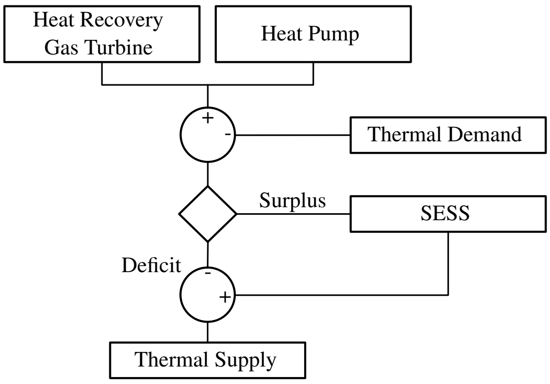


Figure 2: System model of the thermal integration and heat recovery [3].

The building thermal demand is modeled dynamically using a 3R2C grey-box model (Section 3), providing time-varying heating and cooling demands for the energy management system.

The resulting multi-energy system enables flexible coupling of electrical, thermal, and hydrogen energy flows and forms the basis for the optimization-based operation strategy.

2 Data and Preprocessing

2.1 Data sources and signals

The data set comprises time series of photovoltaic power generation, electrical load, thermal demand, and meteorological variables, including ambient temperature, global horizontal irradiance, wind speed, and wind direction.

The data cover the period 2023–2025 and are obtained from the building management system of Technische Hochschule Ulm (TH Ulm), University of Applied Sciences.

Figure 3 illustrates exemplary annual meteorological time series, highlighting both seasonal patterns and short-term variability relevant for load forecasting and building thermal modeling.

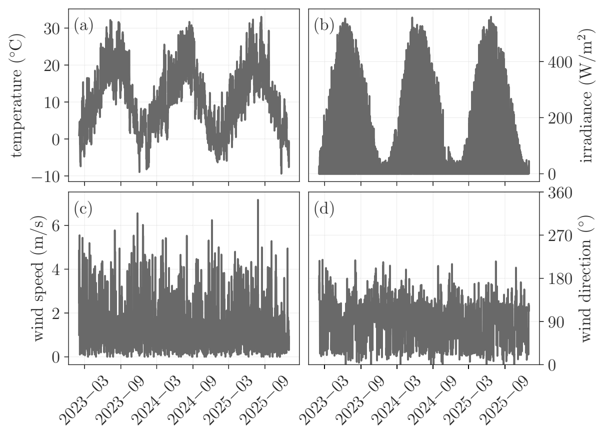


Figure 3: Exemplary annual time series of meteorological variables (2023–2025).

Monthly aggregated energy quantities of photovoltaic generation, electrical load, and thermal demand are shown in Fig. 4, highlighting their pronounced seasonal mismatch.

2.2 Temporal resolution and alignment

All signals are aligned to a common hourly time grid. Higher-resolution data are aggregated in an energy-conserving manner, while lower-resolution data are upsampled using zero-order hold.

Time stamps are converted to a consistent time zone, and short data gaps are filled by interpolation.

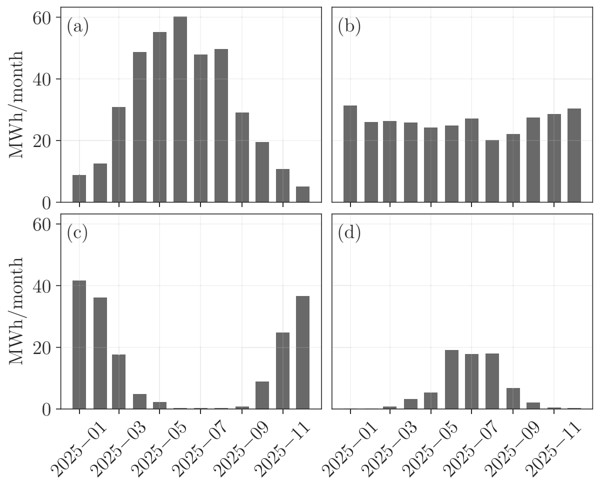


Figure 4: Monthly aggregated energy quantities of photovoltaic generation (a), electrical load (b), and thermal demand (c,d) for the period 2023–2025.

3 Building Model and Calibration (3R2C + IPOPT)

3.1 Grey-box model structure

The building thermal behavior is modeled using a ventilation-aware 3R2C grey-box model following [4]. Two thermal states are considered: the indoor air temperature T_{in} and the effective envelope temperature T_{e}.

The model is driven by ambient temperature T_{a}, solar gains Q_{s}, and HVAC thermal input Q_{h}, allowing the separation of fast indoor air dynamics and slower envelope dynamics.

3.2 Continuous-time 3R2C dynamics

The continuous-time dynamics of the 3R2C model are given by

$$C_{\text{in}}\frac{dT_{\text{in}}}{dt} = \frac{T_{\text{e}}-T_{\text{in}}}{R_{\text{in,e}}} + \frac{T_{\text{a}}-T_{\text{in}}}{R_{\text{in,a}}} + f_{\text{h}}Q_{\text{h}} + A_{\text{in}}Q_{\text{s}}, \tag{1}$$

$$C_{\text{e}}\frac{dT_{\text{e}}}{dt} = \frac{T_{\text{in}}-T_{\text{e}}}{R_{\text{in,e}}} + \frac{T_{\text{a}}-T_{\text{e}}}{R_{\text{e,a}}} + (1-f_{\text{h}})Q_{\text{h}} + A_{\text{e}}Q_{\text{s}}. \tag{2}$$

For the discretization a discrete time step Δt is used and the model is transformed into the state-space form

$$\mathbf{T}_{k+1} = \mathbf{A}_d(\theta)\,\mathbf{T}_k + \mathbf{B}_d(\theta)\,\mathbf{u}_k, \tag{3}$$

which is embedded as linear constraints in the MPC after calibration.

3.3 Parameter estimation as NLP (IPOPT)

The parameter vector

$$\theta = \{R_{\text{in,e}}, R_{\text{in,a}}, R_{\text{e,a}}, C_{\text{in}}, C_{\text{e}}, A_{\text{in}}, A_{\text{e}}, f_{\text{h}}, T_{\text{e},0}\}$$

is identified by solving a nonlinear program subject to the discretized model dynamics. Calibration minimizes a weighted sum of the indoor air temperature tracking error

$$J_T(\theta) = \frac{1}{N}\sum_{k=1}^{N}\left(T_{\text{in},k}(\theta) - T_{\text{in},k}^{\text{meas}}\right)^2, \tag{4}$$

In contrast to the method proposed by Panagi et al. [4], we additionally include the normalized deviation between simulated and measured monthly thermal energy demand,

$$J_E(\theta) = \frac{1}{12}\sum_{m=1}^{12}\left(\frac{E_m^{\text{sim}}(\theta) - E_m^{\text{meas}}}{\max(\varepsilon_E, E_m^{\text{meas}})}\right)^2. \tag{5}$$

Both objectives are combined as

$$\min_{\theta} J(\theta) = w_T J_T(\theta) + w_E J_E(\theta), \tag{6}$$

subject to the discretized model equations and parameter bounds.

3.4 Model evaluation

Model performance is assessed on both the calibration data set and an independent test period.

Figure 5 shows the indoor air temperature during the best-fitting calibration window, illustrating that the identified parameters accurately reproduce the short-term thermal dynamics of the building.

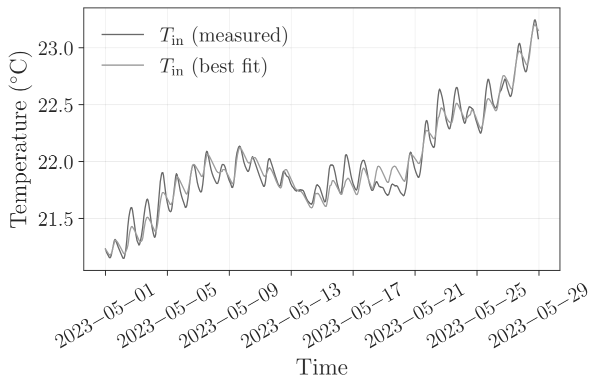


Figure 5: Measured and simulated indoor air temperature during the calibration window (best training fit).

Generalization capability is evaluated on the unseen test year 2025.

Figure 6 compares simulated and measured monthly thermal energy demand, demonstrating that the model reproduces the aggregated thermal energy balance over an annual horizon with sufficient accuracy for predictive control applications.

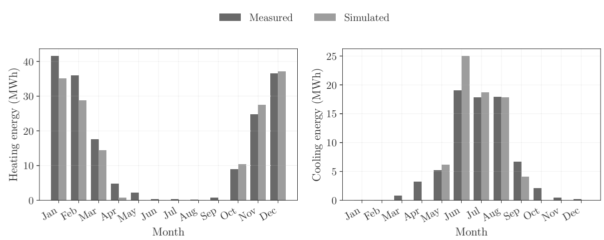


Figure 6: Comparison of measured and simulated monthly thermal energy demand for the test year 2025.

3.5 Thermal Component Modelling (1R1C)

Thermal components such as the electrolyzer and gas turbine are represented by a first-order resistance–capacitance (1R1C) model following [5].

The discrete-time temperature dynamics are given by the formula

$$T_{k+1} = T_k + \frac{\Delta t}{C}\left(Q_{\mathrm{in},k} - \frac{T_k - T_{\mathrm{a},k}}{R}\right), \tag{7}$$

and are consistently embedded into the MPC formulation.

Model parameters are calibrated such that a warm-up period of 15 min is required before production starts. The electrolyzer and gas turbine efficiencies are modeled as load-dependent and represented by a piecewise linear approximation using SOS2 constraints [6].

4 Forecasting Models

4.1 Role in MPC

Short-term forecasts of exogenous signals are required for predictive control in multi-energy systems, as photovoltaic generation, electrical load, and ambient temperature directly affect storage scheduling and unit commitment decisions.

Following [7], data-driven forecasting models are therefore integrated into the MPC framework to provide horizon-wise predictions over the control horizon.

Multivariate Long Short-Term Memory (LSTM) networks are employed due to their ability to capture diurnal patterns and temporal dependencies in energy and weather-related time series [8].

Separate models are trained for irradiance, PV power, electrical load, and ambient temperature to account for their differing dynamics.

4.2 Model setup and training

All forecasting models are trained on synchronized hourly data from 2023–2024 using a strict chronological split, while the year 2025 is reserved for out-of-sample evaluation and rolling MPC operation.

Multichannel input sequences of length $L = 24\,\mathrm{h}$ are used and normalized using Min–Max scaling. Temporal ordering is preserved by disabling shuffling.

Lightweight, target-specific LSTM architectures are employed. Linear output activation is used for temperature forecasting, whereas softplus activation is applied to power-related targets to enforce non-negativity.

A seasonal naive forecast with a 24 h lag serves as a baseline. The global training configuration and target-specific network settings are summarized in Tables 1 and 2.

Setting	Value
Train/Val window	2023-02-01 .. 2024-12-31
Chronological split	Train: 90%, Val: 10% (tail)
Test window	2025-01-01 .. 2025-12-31
Rolling export (MPC)	2025-01-01 .. 2025-12-31
Input sequence length (L)	24 h
Forecast horizon (H)	24 h
Batch size	128
Epochs	200
Scaler	Min–Max (0..1)
Multichannel inputs	True
Shuffle training blocks	False
Seasonal naive baseline	24 h

Table 1: Global configuration for all LSTM forecast models.

Target	LSTM units	Output activation
T_{air}	64	linear
PV power	32	softplus
Load	32	softplus
Irradiance	32	softplus

Table 2: Target-specific LSTM output configuration. Softplus enforces non-negativity for power and irradiance targets.

4.3 Forecast accuracy

Forecast accuracy is evaluated on the full test year 2025 using standard error metrics. Table 3 compares the best-performing LSTM models with a seasonal naive baseline. Across all targets, the LSTM models substantially reduce prediction errors, particularly for electrical load and PV power, which are most relevant for MPC decision-making.

Target	Model	RMSE	MAE	R^2
(G_h)	SN	99.56 W/m^2	44.27 W/m^2	0.762
	LSTM	**79.48 W/m^2**	45.91 W/m^2	**0.848**
(P_{PV})	SN	35.73 kW	15.74 kW	0.743
	LSTM	**28.28 kW**	15.45 kW	**0.839**
(P_{load})	SN	15.80 kW	9.12 kW	0.311
	LSTM	**7.91 kW**	5.54 kW	**0.827**
(T_{air})	SN	3.20 °C	2.43 °C	0.843
	LSTM	**2.43 °C**	1.81 °C	**0.909**

Table 3: Forecast accuracy on the test year 2025. SN denotes a seasonal naive baseline (24 h). LSTM uses the full feature set.

5 MPC formulation (MILP with ε-constraint on starts)

5.1 Decision variables

Over the prediction horizon $k = 1,\ldots,H$, the MPC optimizes electrical and thermal power flows, storage states, and binary commitment variables.

The decision vector $\mathbf{x}$ comprises:

(i) electrical grid exchange, battery charge/discharge, electrolyzer power, and gas turbine electric output;

(ii) thermal grid exchange, TES charge/discharge, gas turbine heat, and heat pump output;

(iii) storage states of BESS, HESS, and TES;

(iv) building states (indoor and envelope temperatures); and

(v) binary variables enforcing unit commitment and mutually exclusive operating modes.

5.2 Objective and MILP

The primary objective minimizes weighted electrical and thermal grid interactions:

$$J_1 = \sum_{k=1}^{H} \Big(w_{\mathrm{imp}} \big(P^k_{\mathrm{grid,imp}} + Q^k_{\mathrm{hot,imp}} + Q^k_{\mathrm{cold,imp}} \big) + w_{\mathrm{exp}} \big(P^k_{\mathrm{grid,exp}} + Q^k_{\mathrm{hot,exp}} + Q^k_{\mathrm{cold,exp}} \big) \Big) \Delta t. \tag{8}$$

with $w_{\mathrm{imp}} > 0$ and typically $w_{\mathrm{exp}} \ll w_{\mathrm{imp}}$.

At each MPC step, the following mixed-integer linear program is solved:

$$\begin{aligned}
\min_{\mathbf{x}} \quad & J_1(\mathbf{x}) && (9)\\
\text{s.t.} \quad & J^{\mathrm{ely}}_{\mathrm{start}}(\mathbf{x}) \le \varepsilon_{\mathrm{ely}}, \quad J^{\mathrm{gt}}_{\mathrm{start}}(\mathbf{x}) \le \varepsilon_{\mathrm{gt}}, && (10)\\
& \mathbf{f}_{\mathrm{el}}(\mathbf{x},\mathbf{d}) = \mathbf{0}, \quad \mathbf{f}_{\mathrm{th}}(\mathbf{x},\mathbf{d}) = \mathbf{0}, && (11)\\
& \mathbf{g}_{\mathrm{stor}}(\mathbf{x}) \le \mathbf{0}, \quad \mathbf{g}_{\mathrm{uc}}(\mathbf{x}) \le \mathbf{0}, && (12)\\
& \mathbf{f}_{\mathrm{bld}}(\mathbf{x},\mathbf{d}) = \mathbf{0}, && (13)
\end{aligned}$$

where $\mathbf{d}$ collects exogenous forecasts (PV generation, electrical demand, weather).

The resulting MILP is implemented in PYTHON using the PYOMO modeling framework and solved with Gurobi [9].

At each MPC step, the MILP is solved to obtain an optimal control sequence, of which only the first control action is applied to the simulation model in a receding-horizon fashion.

5.3 ε-constraint on start events (Ely/GT)

To limit start events, the electrolyzer (Ely) and gas turbine (GT) are explicitly constrained. Binary start indicators detect off–on transitions:

$$\delta_{\mathrm{u,start}}^{k} \geq \delta_{\mathrm{u}}^{k} - \delta_{\mathrm{u}}^{k-1}, \qquad \mathrm{u} \in \{\mathrm{ely}, \mathrm{gt}\}, \quad \forall k \geq 1, \tag{14}$$

with known initial states δ_{u}^{0} from the previous MPC step. The total number of starts over the horizon is

$$J_{\mathrm{start}}^{\mathrm{u}} = \sum_{k=1}^{H} \delta_{\mathrm{u,start}}^{k}, \qquad \mathrm{u} \in \{\mathrm{ely}, \mathrm{gt}\}, \tag{15}$$

and constrained as

$$J_{\mathrm{start}}^{\mathrm{ely}} \leq \varepsilon_{\mathrm{ely}}, \qquad J_{\mathrm{start}}^{\mathrm{gt}} \leq \varepsilon_{\mathrm{gt}}, \tag{16}$$

with $\varepsilon_{\mathrm{ely}} = \varepsilon_{\mathrm{gt}} = 3$ in this study.

5.4 Constraints

Electrical power balance.

$$P_{\mathrm{pv}}^{k} + P_{\mathrm{gt,el}}^{k} + P_{\mathrm{bat,dis}}^{k} + P_{\mathrm{grid,imp}}^{k} = P_{\mathrm{load}}^{k} + P_{\mathrm{ely,el}}^{k} + P_{\mathrm{preheat}}^{k} + P_{\mathrm{bat,ch}}^{k} + P_{\mathrm{HP,heat,el}}^{k} + P_{\mathrm{HP,cool,el}}^{k} + P_{\mathrm{grid,exp}}^{k}, \tag{17}$$

Here, P_{preheat}^{k} denotes the additional electrical power demand associated with start-up pre-heating of the electrolyzer and the gas turbine, which is activated during off–on transitions.

Thermal power balance (heating).

$$Q_{\mathrm{gt,heat}}^{k} + Q_{\mathrm{HP,heat}}^{k} + Q_{\mathrm{TES,dis}}^{k} + Q_{\mathrm{grid,imp}}^{k} = Q_{\mathrm{dem}}^{k} + Q_{\mathrm{TES,ch}}^{k} + Q_{\mathrm{grid,exp}}^{k}. \tag{18}$$

The heat pump is modeled with thermal power as the decision variable, while the corresponding electrical power consumption is imposed by a linear equality constraint:

$$P_{\mathrm{HP,heat,el}}^{k} = \frac{Q_{\mathrm{HP,heat}}^{k}}{\mathrm{COP}_{\mathrm{HP,heat}}}. \tag{19}$$

An analogous formulation is applied to cooling, including the corresponding thermal balance, electrical coupling via the energy efficiency ratio (EER), and thermal energy storage.

Storage dynamics (example: BESS).

$$E_{\mathrm{BESS}}^{k+1} = E_{\mathrm{BESS}}^{k} + \left(\eta_{\mathrm{ch}} P_{\mathrm{bat,ch}}^{k} - \frac{1}{\eta_{\mathrm{dis}}} P_{\mathrm{bat,dis}}^{k} \right) \Delta t, \tag{20}$$

$$E_{\mathrm{BESS,min}} \leq E_{\mathrm{BESS}}^{k} \leq E_{\mathrm{BESS,max}}. \tag{21}$$

Analogous formulations apply to hydrogen storage (HESS) and thermal energy storage (TES). Binary variables enforce mutually exclusive operating modes and unit commitment constraints for the battery, electrolyzer, and gas turbine.

6 Case Setup

6.1 Considered control strategies

Four control strategies are evaluated under identical system conditions to assess the impact of predictive control and forecast accuracy:

- **Case 1 – Rule-Based Control (RB):** Heuristic control with fixed priority rules and no explicit forecasting or optimization.
- **Case 2 – MPC with Seasonal Naive Forecast:** MPC using a 24 h seasonal naive forecast.
- **Case 3 – MPC with LSTM Forecasts:** MPC using LSTM-based forecasts for photovoltaic generation, electrical load, thermal demand, and ambient temperature.
- **Case 4 – MPC with Perfect Forecast:** Idealized MPC with perfect knowledge of all exogenous signals, serving as an upper performance bound.

6.2 System parameters

Table 4 summarizes the parameters used in the MPC–SIM framework. The configuration represents a small-scale, grid-connected P-t-X system with coupled electrical, thermal, and hydrogen subsystems. State-of-charge limits and operational constraints are enforced for all storage components.

The current installation at the Technische Hochschule Ulm includes a 20 kW electrolyzer and a high-pressure hydrogen storage with a capacity of approximately 533 kWh. Since both components are scheduled for expansion, the parameters listed in Table 4 represent an extended configuration to assess future system operation and scalability.

Subsystem	Parameter	Value
BESS	E_{BESS}	[39, 349] kWh
	P_{BESS}	± 194 kW
	η_{BESS}	0.95
H_2 Storage	E_{H2}	[170, 2133] kWh
Electrolyzer	$P_{\mathrm{Ely,el}}$	[32, 80] kW
	$\eta_{\mathrm{Ely,el}}$	piecewise ($\eta_{\max} = 0.60$)
	$T_{\mathrm{Ely,set}}$	60°C
Gas Turbine	$P_{\mathrm{GT,el}}$	[50, 100] kW
	$\eta_{\mathrm{GT,el}}$	piecewise ($\eta_{\max} = 0.32$)
	$\eta_{\mathrm{GT,total}}$	0.90
	$T_{\mathrm{GT,set}}$	600°C
Heat Pump	$P_{\mathrm{HP,el}}$	≤ 50 kW
	COP / EER	4.0/4.0
TES (hot / cold)	E_{TES}	hot: [58, 580] kWh
		cold: [35, 350] kWh
	P_{TES}	± 100 kW
	η_{TES}	0.98
Building (3R2C)	R, C, A, f_h	calibrated
	T_{set}	22°C / 24°C

Table 4: Compact overview of system parameters used in the MPC–SIM framework.

7 Results and Discussion

7.1 Annual KPIs

Table 5 summarizes the annual electrical and thermal performance of the investigated control strategies. Electrical performance is assessed using self-consumption and self-sufficiency.

Electrical self-consumption is defined as the ratio of locally generated electricity directly utilized within the system to total local generation,

$$\text{Self-Consumption} = \frac{E_{\mathrm{selfcons,el}}}{E_{\mathrm{local,gen,el}}}, \tag{22}$$

while electrical self-sufficiency denotes the fraction of end-use demand covered by local generation,

$$\text{Self-Sufficiency} = \frac{E_{\mathrm{selfcons}}}{E_{\mathrm{enduse}}}. \tag{23}$$

Overall self-sufficiency is defined analogously and accounts for combined electrical and thermal end-use demand.

Compared to the rule-based reference, all MPC-based strategies significantly increase electrical self-consumption and self-sufficiency.

The MPC with perfect forecasts (MPC-P) achieves the highest values and thus represents an upper performance bound.

Using imperfect forecasts, both the LSTM-based and seasonal naive MPC maintain performance close to this optimum, indicating robust behavior under forecast uncertainty.

In the thermal domain, heating self-sufficiency remains high for all cases, while cooling self-sufficiency is substantially improved by MPC-based operation. Overall, the results confirm that forecast-based MPC consistently enhances electrical and thermal system autonomy without requiring perfect foresight.

7.2 Time series comparison

To complement the annual KPIs, a qualitative comparison of the operational behavior is provided based on exemplary time series of electrolyzer and gas turbine operation. Figures 7 and 8 illustrate the temporal utilization patterns of the different control strategies over the year.

For the rule-based reference, the electrolyzer is predominantly operated during summer periods, while the gas turbine is mainly activated in the morning hours. These patterns reflect the direct response to instantaneous system states and fixed priority rules.

In contrast, all MPC-based strategies result in more structured and temporally coherent operation patterns, reflecting anticipative scheduling and improved coordination between components.

Metric [%]	RB	MPC-P	MPC-LSTM	MPC-SN
Electrical self-consumption	81.39	90.31	89.51	89.37
Electrical self-sufficiency	72.56	76.78	75.77	75.86
Overall self-sufficiency (el+th)	76.37	81.63	80.78	80.79
Thermal self-sufficiency (heating)	99.72	94.75	94.21	94.41
Thermal self-sufficiency (cooling)	54.51	94.34	93.53	92.66

Table 5: Annual electrical and thermal self-sufficiency and self-consumption for the investigated cases.

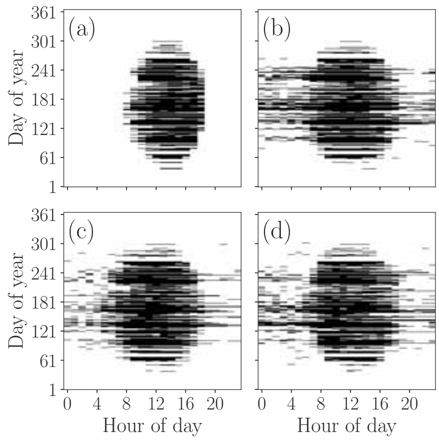


Figure 7: Annual operation patterns of the electrolyzer for the four control strategies: (a) rule-based, (b) MPC-P, (c) MPC-SN, and (d) MPC-LSTM.

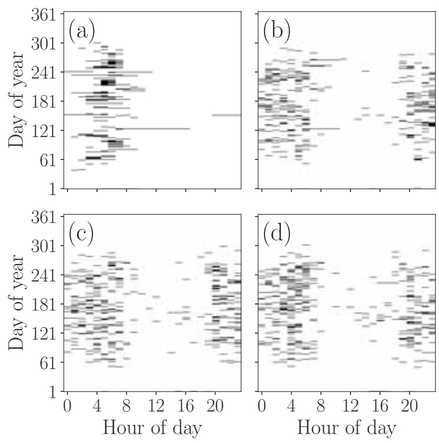


Figure 8: Annual operation patterns of the gas turbine for the four control strategies: (a) rule-based, (b) MPC-P, (c) MPC-SN, and (d) MPC-LSTM.

8 Conclusion and Outlook

This paper presented a forecast-based MPC framework for the operation of a hydrogen-integrated multi-energy system coupled with a dynamic building model. The MPC was formulated as a MILP with an ε-constraint on unit start events and evaluated under different forecast assumptions.

The results show that MPC-based operation consistently outperforms a rule-based reference with respect to electrical and thermal self-sufficiency and self-consumption.

While perfect forecasts yield the highest performance, the use of imperfect forecasts—both LSTM-based and seasonal naive—leads to only marginal reductions. This indicates that the coordinated utilization of flexible components enabled by MPC is more influential than forecast accuracy for the considered setup.

The optimization employs a weighted-sum objective, yielding Pareto-optimal solutions for a given set of weights.

Future work will extend the formulation towards a true multi-objective framework, for example using lexicographic optimization, to systematically explore trade-offs between electrical and thermal objectives.

In addition, the influence of the prediction horizon length on performance and operational behavior will be investigated, particularly with respect to computational effort, forecast uncertainty, and components with slow dynamics.

Further research will address uncertainty-aware optimization and experimental validation of the proposed control framework.

Publication Remark

This contribution is the extended and improved English version of the Workshop Proc. German version for
ASIM Workshop GMMS/STS 2025
Munich/Oberpfaffenhofen 2025,
published in
ASIM Workshop GMMS/STS 2025 Tagungsband
ARGESIM Report 48, e-ISBN 978-3-903347-66-3,
Volume DOI 10.11128/arep.48, p 5-12.

References

[1] Gregor, Liselotte. Hydrogen as an energy carrier for a climate-neutral economy. *EU hydrogen policy*. 2021;.

[2] Kalchschmid V, Erhart V, Angerer K, Roth S, Hohmann A. Decentral Production of Green Hydrogen for Energy Systems: An Economically and Environmentally Viable Solution for Surplus Self-Generated Energy in Manufacturing Companies. *Sustainability*. 2023;15(4). URL `https://www.mdpi.com/2071-1050/15/4/2994`

[3] Dominic A, Giliomee J, Zenner T, Winter M, Schullerus G, Booysen M. Green Hydrogen for Future Energy Demand in Germany. In: *2024 IEEE 8th Energy Conference (ENERGYCON)*. 2024; pp. 1–6.

[4] Panagi S, Aresti L, Christodoulides P, Spanias C, Aristidou P. Tuning Performance of Grey-Box Models for Thermal Building Applications. *Preprint*. 2025; Preprint. Available under CC BY 4.0 license. URL `https://doi.org/10.31224/4879`

[5] Huang C, Zong Y, You S, Træholt C. Economic model predictive control for multi-energy system considering hydrogen-thermal-electric dynamics and waste heat recovery of MW-level alkaline electrolyzer. *Energy Conversion and Management*. 2022;265:115697. URL `https://www.sciencedirect.com/science/article/pii/S0196890422004939`

[6] Zhadan A, Martemyanov A, Allahverdyan A, Petrosian O, Gao H. Linearization Method for MINLP Energy Optimization Problems. *Scientific Reports*. 2025; 15(1):27853. URL `https://doi.org/10.1038/s41598-025-11380-5`

[7] Jung M, da Costa Mendes PR, Önnheim M, Gustavsson E. Model Predictive Control when utilizing LSTM as dynamic models. *Engineering Applications of Artificial Intelligence*. 2023;123:106226. URL `https://www.sciencedirect.com/science/article/pii/S0952197623004104`

[8] Kreuzer D, Munz M, Schlüter S. Short-term temperature forecasts using a convolutional neural network — An application to different weather stations in Germany. *Machine Learning with Applications*. 2020;2:100007. URL `https://www.sciencedirect.com/science/article/pii/S2666827020300074`

[9] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual. 2024. URL `https://www.gurobi.com`

[10] Drgoňa J, Arroyo J, Cupeiro Figueroa I, Blum D, Arendt K, Kim D, Ollé EP, Oravec J, Wetter M, Vrabie DL, Helsen L. All you need to know about model predictive control for buildings. *Annual Reviews in Control*. 2020;50:190–232. URL `https://www.sciencedirect.com/science/article/pii/S1367578820300584`

Object-Oriented Implementation of a Simulator for Linear Implicit Equilibrium Dynamics

Dirk Zimmer

Institute of System Dynamics and Control, DLR, Münchener Straße 20, 82234 Weßling, Germany; *dirk.zimmer@dlr.de*

SNE 36(1), 2026, 55-59, DOI: 10.11128/sne.36.tn.10768
Selected ASIM SST 2024 Postconf. Publication: 2025-06-30
Received Revised: 2025-12-19; Accepted: 2025-12-29
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna
ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. Models based on linear implicit equilibrium dynamics form a special but very useful sub-set of DAE systems that can be applied for the simulation of technical physical systems. Since these are based on differential-algebraic equations, a transformation into executable code must be performed for the purpose of model evaluation. One way to achieve this is the object-oriented formulation of the simulation code itself. To explore this path a dedicated simulator prototype has been implemented and is outlined here. The long-term goal is to define alternative compilation targets for a Modelica compilers that enable highly scalable simulation code for very large systems.

1 What is Linear Implicit Equilibrium Dynamics?

Linear Implicit Equilibrium Dynamics (LIED) is technically defined as a special class of Differential Algebraic Equation (DAE) Systems.

Formal Definition

A DAE system with potential state derivatives $\dot{\boldsymbol{x}}$, time t and algebraic variables $\boldsymbol{w}$

$$\mathbf{0} = \boldsymbol{F}(\dot{\boldsymbol{x}}, \boldsymbol{x}, \boldsymbol{w}, t)$$

is defined as LIED system when it can be transformed into the following form:

$$\begin{bmatrix} \mathbf{w}_E \\ \dot{\boldsymbol{x}}_E \end{bmatrix} = \mathrm{g}(\boldsymbol{x}_I, \boldsymbol{x}_E, t)$$

$$\mathbf{A}(\boldsymbol{x}_I, \boldsymbol{x}_E, \mathbf{w}_E) \begin{bmatrix} \mathbf{w}_I \\ \dot{\boldsymbol{x}}_I \end{bmatrix} = \mathrm{f}(\boldsymbol{x}_I, \boldsymbol{x}_E, \mathbf{w}_E, t)$$

We see that both the algebraic variables as well as the state derivatives can be split into a fully explicit part ($\dot{\boldsymbol{x}}_E$; $\boldsymbol{w}_E$) and a part ($\dot{\boldsymbol{x}}_I$; $\mathbf{w}_I$) with a linear system in implicit form expressed by the regular matrix $\mathbf{A}$. Furthermore, the following conditions shall hold true:

$$\dot{\boldsymbol{x}}_E \cap \dot{\boldsymbol{x}}_I \subseteq \dot{\boldsymbol{x}}$$
$$\boldsymbol{w}_E \cap \boldsymbol{w}_I \supseteq \boldsymbol{w}$$
$$\dot{\boldsymbol{x}}_E \cap \dot{\boldsymbol{x}}_I \cap \boldsymbol{w}_I \supseteq \dot{\boldsymbol{x}}$$
$$\dot{\boldsymbol{x}}_E, \dot{\boldsymbol{x}}_I, \boldsymbol{w}_E, \boldsymbol{w}_I \text{ are all disjoint}$$

These conditions essentially mean that it is allowed to perform certain symbolic mechanism of index reduction such as the dummy derivative method [5] originating from Pantelides [6]. Using this method, states variables of $\boldsymbol{x}$ can be transformed to algebraic variables in $\boldsymbol{w}_I$ and further derivatives may be added to $\boldsymbol{w}_I$ or $\boldsymbol{w}_E$. In practice, this is important because it means that the linear implicit dynamics can be expressed by far fewer states than suggested by the vector $\boldsymbol{x}$ of the original DAE formulation.

Informal Explanation

The formal definition above may be primarily perceived as a relatively strong restriction on the model equations and not many systems may be intuitively expected to fall into this category. Surprisingly, LIED can be applied successfully for the object-oriented modelling of complex thermofluid architectures [7],[8] or to mechanical systems with stiff contacts [9].

The idea is that the non-linear behaviour of the slow mode is explicitly expressed whereas the fast dynamics that typically is needed to uphold non-linear constraints is expressed by a linear implicit system that fulfils the constraint in its equilibrium. Hence the name: linear implicit equilibrium dynamics. The equilibrium dynamics is thereby often a replacement dynamic and only an approximation of reality (as all modelling is).

1 What is LIED good for?

As the above references demonstrate, LIED has been applied using Modelica [1] for the object-oriented modeling of thermofluid or mechanical systems.

To this end, it is necessary to use triplets as interface of the model components that consist in a signal for the explicit non-linear part and a pair of potential as presented in Table 1.

Domain	Signal	Potential	Flow
trans. mechanics	position: r [m]	velocity: v [m/s]	force: f [N]
rotational mechanics	angle: φ [rad]	angular velocity: ω [rad/s]	torque: τ [Nm]
thermofluid streams	Thermo-dynamic state: $\widehat{\Theta}$	inertial pressure r [Pa]	mass-flow rate: $\dot{m}$ [kg/s]

Table 1: Connection triplets for the object-oriented modelling of LIED Systems.

More background on the derivation of these triplets can be found in [10]. It goes beyond the scope of this paper how the equations are formulated in detail but the two Modelica model diagrams shown in Figure 1 and 2 may illustrate the practical usefulness.

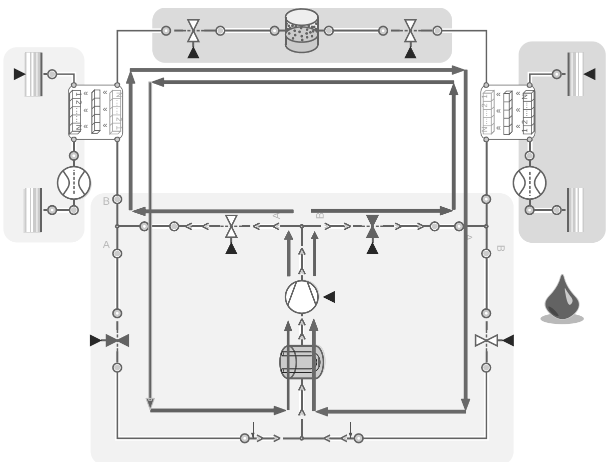


Figure 1: Model diagram of a reversible heat pump systems using the ThermoFluid Stream Library [8].

Especially the ThemoFluid Stream Library has meanwhile become a popular OpenSource library both by academia [4] and by industry [7].

LIED systems have very benevolent characteristics for object-oriented modelling. Following simple connection rules, the resulting matrix $\mathbf{A}(\boldsymbol{x}_I, \boldsymbol{x}_E, \boldsymbol{w}_E)$ will be regular and an a-priori statement on solvability can be given [7]. This makes this class of modeling very robust and prevents many computational simulation errors.

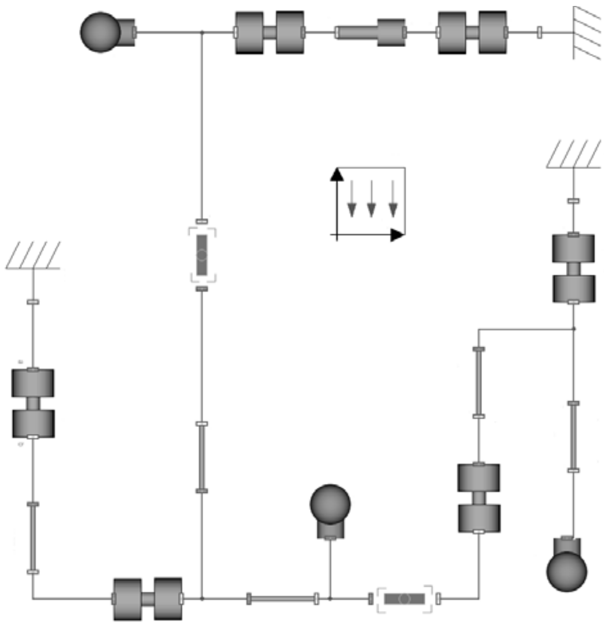


Figure 2: Model diagram of a mechanical system for a kinematic using the Dialectic Mechanics library [9].

2 A Dedicated Simulator

2.1 Why?

The libraries are currently developed using Modelica and since LIED systems are a subset of DAE systems and Modelica can deal with DAEs in general, any Modelica compiler can be used to generate simulation code out of LIED systems. A dedicated simulator is thus not needed. Existing and mature simulation software can be used.

However, Modelica compilers are very complex and algorithms for the processing of general DAEs involve drawbacks: all Modelica compilers create a "flat" model, where all equations are collected in a single set. For very large systems, this poses problems primarily because large amounts of code are generated.

Out of practical experience, many LIED systems have very benevolent structural characteristics that enable a compilation already on the component level. A compiler for standard Modelica cannot exploit these benefits, albeit they may greatly increase the ability to deal with large systems or with variable structure systems.

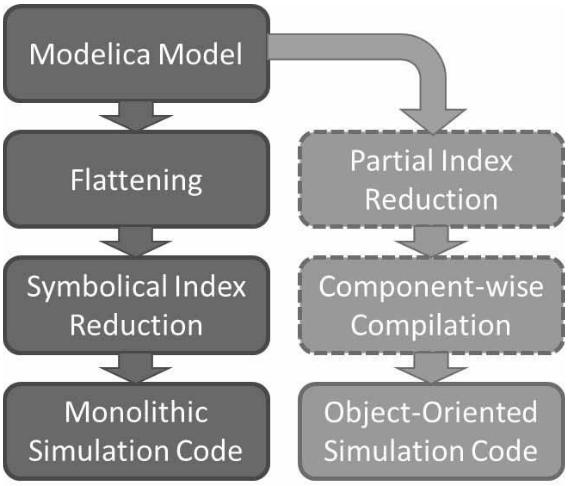


Figure 3: Illustration of different pathways for the compilation of simulation code. The blue boxes on the left illustrate the classic compilation of Modelica where all equations are collected in a single set. The green boxes on the right illustrate a potential pathway for LIED systems. This paper focuses on the compile target which is the object-oriented simulation code.

To explore alternative compile targets dedicated for LIED systems, a prototypical simulator has been developed with the name `zimsim`.

2.2 Computational structure and implementation of blocks

Figure 5 presents a simple planar mechanical system of a crane crab (a pendulum attached to a slider). To each component of the system 3 computational blocks are assigned of different color: blue, green, and orange. The blue and green blocks form thereby a computational sequence directed from the root whereas the orange signal leads to the root. For this particular domain of LIED systems, the blue signal contains the positional state and undergoes non-linear transformations. The transformation of the green and orange signal forms a linear response which can be used to solve the linear equations system that spans across the components.

```
class RevoluteJoint: public Component
    Signal flangeTo, flangeOn
    double phi, phi_der, w, z, residualTorque;
    ContinuousState position{ phi,phi_der };
    ContinuousState velocity{ w,z };
    Tearing zeroTorque{ phi_der, residualTorque };
    void evalState();
    void evalKinetic();
    void evalImpulse();
    virtual void metainfo(Meta& meta)
```

Figure 4: C++ class diagram of a revolute component.

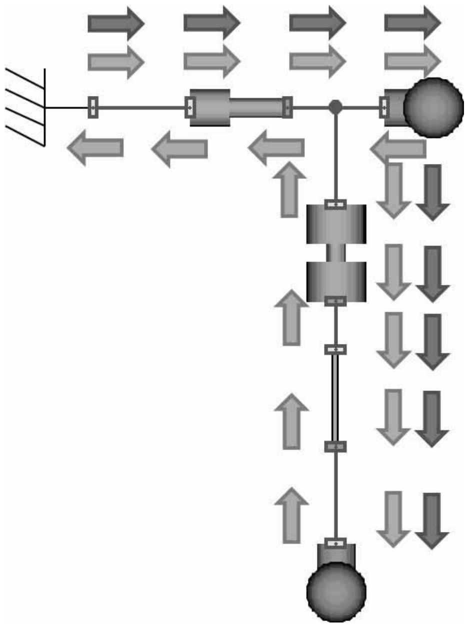


Figure 5: Crane crab modelling diagram.

In practice this means that the code for each component can be represented by dedicated C++ class and the computational blocks can be expressed by C++ member functions. This is illustrated in Figure 4 with matching colors.

2.3 Handling of meta information

It is necessary to register the state-variables and their derivatives in the system. This is done by member objects. In similar vein, tearing variables and residuals are registered. The tearing variable belongs to the green signal and used to probe the system whose linear response can be assessed by the residual belonging to the orange signal flow. Furthermore, for each block (or member function) the dependence on the input and output signals needs to be registered.

For these purposes each component class must contain a virtual member function called `metainfo`. This function takes a crawler object by reference and depending on the implementation of the crawler different metainformation may be extracted.

2.4 Ordering of blocks, compacting and extracting of sub-blocks.

When instantiating a component class, all structural relevant meta data are collected. With this information, it is now possible to put all calls to member functions of all components in correct order. First a partial order is created based on the signal dependence, then the linear systems (of different dimension) marked by tearing and residual variables are compacted and in the last stage the dynamic part is compacted (meaning that the all constant evaluations are placed upfront and all calls not needed for derivative evaluation are put last). With this ordering the linear subsystems can be constructed and solved and the overall model can be evaluated.

2.5 ODE simulation

Runge-Kutta solvers with fixed step size of order 1 to 4 as well as Backward Euler and ESDIRK23 [3] with variable step-size control have been implemented as numerical ODE solvers.

2.6 Overall Simulator Software Design

The major classes of the simulator are:

- `ModelEvaluation` instantiates the model, collects structural meta-information, orders the computational blocks and offer model evaluation
- `Simulator` implements numerical ODE solver
- `Recorder` collects references to desired outputs and generates output either to file, memory or TCP port.

3 First Scaling Results

3.1 The scaling experiments

The crane crab model of Figure 5 has been used to perform a simple scaling experiment. Using a logarithmic grid of factor 4, 1 to 16384 crane crabs have been instantiated and simulated using `zimsim`. The same exercise has been performed within a commercial Modelica tool. Here, this generates models ranging from 206 equations up to 3.1 million equations. In Figures 6 and 7, we use the equation number since this quantity is more familiar to Modelica users.

3.2 Scaling results in terms of memory usage

Peak memory usage for model translation, compilation and simulation is orders of magnitudes lower using `zimsim`. The likely reasons are:

- Avoidance of flattening
- On demand generation of meta-information
- More variables on stack than on heap

For small models, comparing the memory usage hardly makes sense because we compare the memory usage of a dedicated simulator (LIED) with the one of a whole modeling and simulation environment (Modelica Tool).

The two orders of magnitude for small models is thus not surprising. What is surprising that this gap does not significantly close for larger models. Peak memory consumption seems to appear in the models at model translation. `zimsim` has been written with efficient use of memory in mind: Meta information is generated only on-demand and also the object-oriented formulation enables to do more computation on the steak than on the heap.

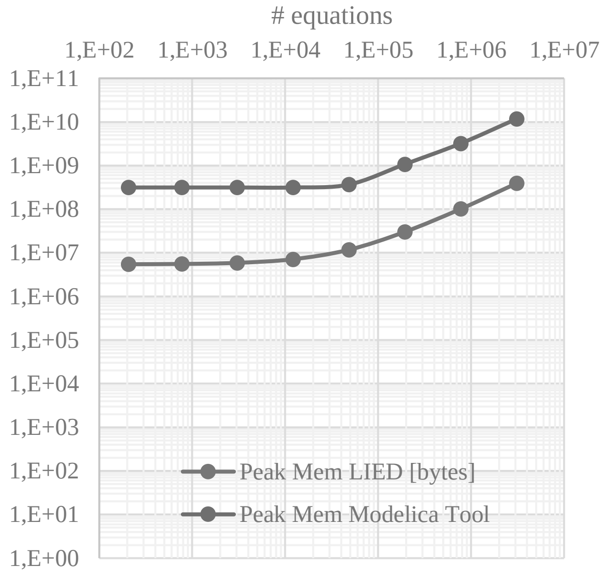


Figure 6: Memory usage in bytes.

3.3 Scaling results in terms of performance

Simulation speed is up to an order of magnitude slower using `zimsim`. The likely reasons are:

- More overhead in `zimsim` due to interface variables and many function calls.
- Numerical solution of linear equation system in `zimsim` instead of symbolical transformation.

Pure simulation time is compared here. Time for translation, compilation and instantiation is ignored. Generation of output has been reduced to a negligible amount.
One has to consider that the code generation of the Modelica tool underwent decades of optimization whereas zimsim is still prototypical. A reduction of 50% in computational effort might be achievable for `zimsim`, however a certain gap will always remain.

We can observe that for larger systems, the performance penalty is smaller, the reason is unknown but is probably due the higher memory efficiency in `zimsim`.

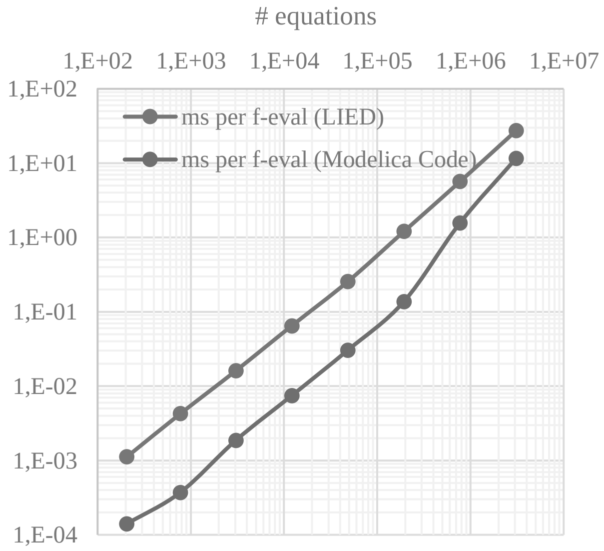


Figure 7: Time for single model evaluation in ms.

A final remark: results for even larger systems could not be attained for `zimsim` as well as for the Modelica tool, however for completely different reasons. The Modelica tool started to hit the limits of memory leading to excessive translation time due to disk swapping. `zimsim` had no such problems but the instantiation has been implemented with a low-performing algorithm, requiring too much time. This is however a pure implementation issue and will be improved.

4 Outlook

The prototypical simulator demonstrates that LIED system can actually be coded directly in C++ and executed with acceptable efficiency and while using comparably little memory. The complexity of the software is thereby significantly lower than of any Modelica environment. The scaling capabilities are promising and can be improved by going to code dedicated for GPUs.

Although the modeling in C++ turned out to be much more natural than originally conceived, it is still not a desirable target. Instead, the existing C++ modelling libraries shall be used as an inspiration for Modelica compilers. Especially open compilers such as the OMC [2] could be modified to compile from Modelica libraries into a set of pre-compiled components.

The main, albeit preliminary, conclusion is that LIED systems not only form an interesting class for robust modelling but also an interesting class for large scale system simulation, worthy of further investigation.

Publication Remark

This contribution is the revised version of the conference version published in

Tagungsband Langbeiträge ASIM SST 2024,
ARGESIM Report AR 47,
e-ISBN 978-3-903347-65-6,
vol. DOI 10.11128/arep.47,
article DOI 10.11128/arep.47.a4735, p 89-93.

References

[1] Fritzson P. *Principles of Object-Oriented Modeling and Simulation with Modelica 3.3*, 2nd ed., IEEE Press, Piscataway, New Jersey, 2014, pp. 1256.

[2] Fritzson PA, et al. The OpenModelica Integrated Modeling, Simulation, and Optimization Environment. 2019. In *Proceedings of The American Modelica Conference 2018*, October 9-10, USA.

[3] Jørgensen JB. Kristensen MR, Grove P. A Family of ESDIRK Integration Methods. 2018. *arXiv Numerical Analysis* eprint :1803.01613.

[4] Junglas P. Implementing Thermodynamic Cyclic Processes Using the DLR Thermofluid Stream Library. 2023. Simulation Notes Europe SNE 33(4), 2023, 175-182. DOI 10.11128/sne.33.sw.10665.

[5] Mattsson SE, Söderlind G. Index Reduction in Differential-Algebraic Equations Using Dummy Derivatives. 1993. In: *SIAM Journal on Scientific Computing* 1993 14:3, 677-692.

[6] Pantelides C. The consistent initialization of differential-algebraic systems. 1988. SIAM J. Sci. Statist. Comput., 9, 213–231.

[7] Zimmer D. Robust Object-Oriented Formu-lation of Directed Thermofluid Stream Networks. 2020. *Mathematical and Computer Modelling of Dynamic Systems*, Vol 26, Issue 3.

[8] Zimmer D, Weber N, Meißner M. The DLR ThermoFluid Stream Library. 2022. *MDPI Electronics -Special Issue*.

[9] Zimmer D, Oldemeyer C. Introducing Dialectic Mechanics. (2023). *Proceedings of the 15th International Modelica Conference*, Aachen.

[10] Zimmer D. Object-Oriented Modeling of Classic Physical Systems using Linear Implicit Equilibrium Dynamics. 2024. *Preprints* 2024, 2024031139.

STROBOSCOPE Model for ARGESIM Benchmark C6 'Emergency Department - Follow-Up Treatment'

Photios G. Ioannou[1], Constantine A. Ioannou[2]

[1]Dept. of Civil and Environmental Engineering, University of Michigan, Ann Arbor, MI, USA, *photios@umich.edu*

[2]School of Information and Computer Sciences, University of California, Irvine, CA, USA

SNE 36(1), 2026, 61-64, DOI: 10.11128/sne.36.bn06.10769
Submitted: 2025-12-20
Received Improved: 2026-01-30; Accepted: 2026-02-02
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna
ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. The STROBOSCOPE simulation system is used to model the follow-up treatment received by four types of patients in a hospital emergency department with two examining rooms, staffed by two doctors of different experience. Depending on their case, patients may be sent for an X-ray and to a plaster room one or two times, and may need to be seen again by the doctors. The effectiveness of three operational alternatives is compared.

Introduction

ARGESIM benchmark C6 [1] investigates how to model the follow-up treatment received by patients in a hospital emergency department, which included the following:

- One registration area (one person) that assigned 60% and 40% of patients to Casualty Wards 1 or 2.
- Casualty Ward 1 (CW1) with two experienced doctors.
- Casualty Ward 2 (CW2) with two inexperienced doctors (for simpler cases).
- One waiting area for CW1 and one for CW2.
- Two X-ray units with a common waiting queue.
- One room that applied or removed plaster.

The four patient types, and their ratios, were as follows:

1. X-ray patients (35%). Patients are seen in CW*k* before being sent for X-rays. Patients return to CW*k*, X-rays are examined, and patients leave.
2. Plaster removal (20%). Patients enter CW*k*, go for plaster removal, and then leave.
3. Plaster renewal (5%). Patients enter CW*k*, are sent for an X-ray, receive a new plaster, return for another X-ray, return to CW*k*, and leave.
4. Changing wound dressings. Patients are seen in CW*k* and then leave.

The time between patient arrivals is distributed exponentially with a mean of 0.3 minutes.

The durations of the various activities are distributed Triangular(*minimum*, *mode*, *maximum*) as follows:

Registration:	Triangular(0.2, 0.5, 1.0) minutes
CW1:	Triangular(1.5, 3.2, 5.0) minutes
CW2:	Triangular(2.8, 4.1, 6.3) minutes
X-ray:	Triangular(2.0, 2.8, 4.1) minutes
Plaster:	Triangular(3.0, 3.8, 4.7) minutes

1 STROBOSCOPE

The STROBOSCOPE simulation system [2] is based on three-phase activity scanning, which is well-suited to modelling cyclic operations and controlled routing of resources, such as those in ARGESIM Benchmark C6. Simulation models are networks of nodes (such as activities and queues) where resources spend time, connected by links that control both when activities can start and the flow of resources (from preceding to succeeding nodes). A graphical user interface (GUI) facilitates the development of simulation model networks (Figure 1).

2 Simulation Model

Figure 1 illustrates the STROBOSCOPE simulation model network for the ARGESIM benchmark C6, which was created using the custom STROBOSCOPE Graphical User Interface (GUI) in Microsoft Visio.

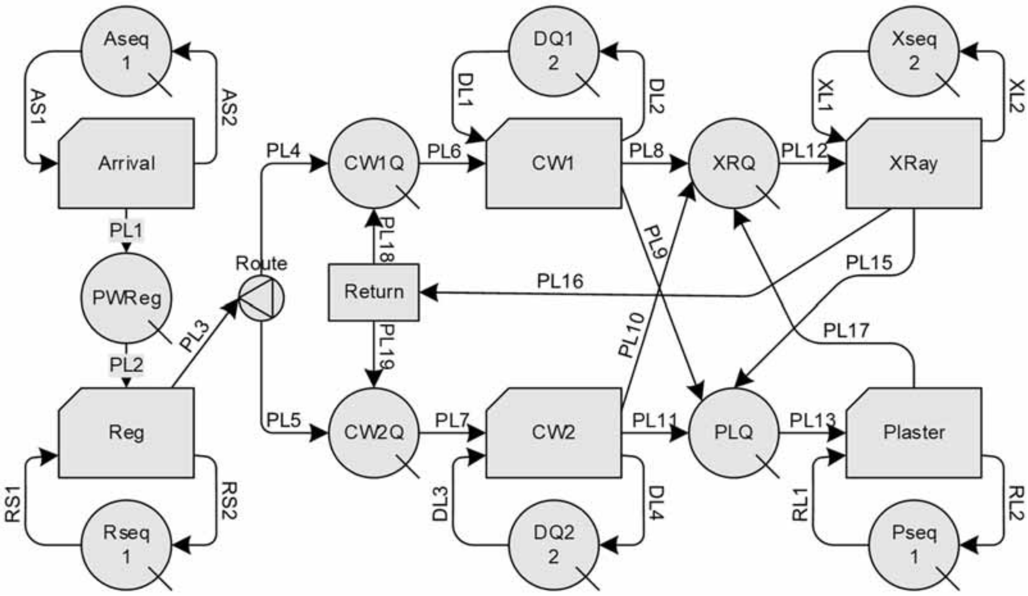


Figure 1: STROBOSCOPE Simulation Model Network.

Patients were modelled as characterized resources of type *Patient* (a class) with four subtypes: *P*1, *P*2, *P*3, and *P*4. Doctors were modelled as characterized resources of type *Doctor* with two subtypes: *D*1 (experienced) and *D*2 (inexperienced). Both *Patients* and *Doctors* have static and dynamic properties that depend on their subtype.

The queues in Figure 1 were initialized with the number of resources indicated inside each node. For example, queues *DQ*1 and *DQ*2 were initialized with 2 *Doctors* each, 2 *D*1 in *DQ*1 and 2 *D*2 in *DQ*2.

The four queues in Figure 1 whose names end in "*seq*" were initialized with 1 or 2 units of the generic resource *Sequence* to start the serial instantiations of their successor conditional activities (combis).

For example, the *Sequence* resource in queue *Aseq* allowed the combi activity *Arrival* to create serial instances that model the arrival of patients. When each *Arrival* instance ended, it generated a *Patient* whose subtype was determined by random sampling, with probabilities *P*1: 35%, *P*2: 20%, *P*3: 5%, and *P*4: 40%.

Each arriving *Patient* then waited in queue *PWReg* until it could register, *Reg*. It was then sent to the fork *Route*, which directed 60% of *Patients* to queue *CW*1*Q* and 40% to queue *CW*2*Q*, where they waited for a doctor in *DQ*k, which enabled treatment to start in activity *CW*k.

When a *Patient* flowed through link *PL*4 or *PL*5, its Casualty Ward (1 or 2) was stored in its *SaveValue WN*.

The number of X-rays (0, 1, 2) that a *Patient* had was stored in its *SaveValue nXRays*. This was increased by 1 each time the *Patient* was drawn by link *PL*12.

After the first examination by a *Doctor* in combi activities *CW*1 or *CW*2, each *Patient* was released to the appropriate sequence of links depending on its subtype.

The appropriate routing of *Patients* was done by the *ReleaseWhere* attribute of each *releasing* link (from a combi to a queue) as follows:

- Links *PL*8 and *PL*10 allowed a *Patient* to flow through *only* if:
  (a) Its subtype was *P*1 and its *nXRays* was 0, or
  (b) Its subtype was *P*3 and *nXRays* was less than 2.
- Links *PL*9 and *PL*11 allowed a *Patient* to be released *only* if their subtype was *P*2.
- Link *PL*15 allowed a *Patient* to be released *only* if its subtype was *P*3 and its *nXRays* equaled 1.
- Link *PL*16 allowed a *Patient* to be released only if:
  (a) Its subtype was *P*1 and its *nXRays* equaled 1, or
  (b) Its subtype was *P*3 and its *nXRays* equaled 2.
- Link *PL*17 allowed a *Patient* to be released only if its subtype was *P*3.
- Links *PL*18 and *PL*19 allowed a *Patient* to be released only if its savevalue *WN* equaled 1 or 2, respectively.

A *Patient* that was not allowed to be released through any of the outgoing links was destroyed at the termination of the combi instance in which it resided.

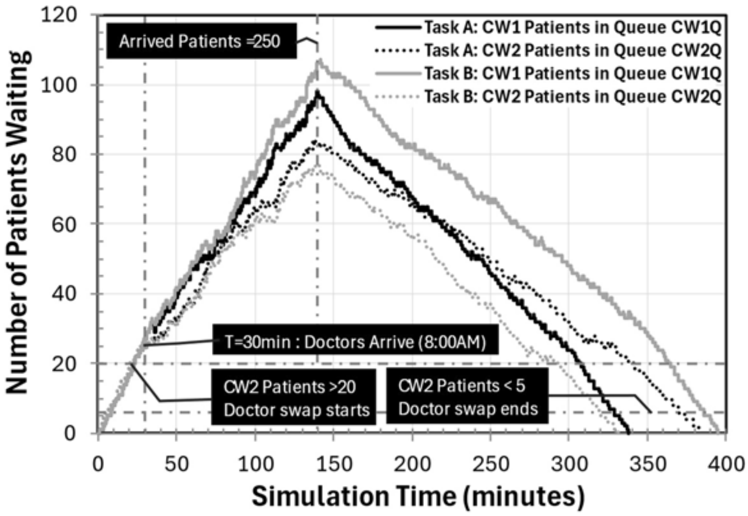


Figure 2: Number of *Patients* waiting in queues *CW*1*Q* and *CW*2*Q* on a typical day for both Task-A and Task-B.

For example, a *Patient* of subtype *P*4 in an instance of *CW*1 or *CW*2 was destroyed along with that instance.

The fact that *Patients* started arriving at 7:30AM while *Doctors* started work at 8:00AM was modelled by the *Semaphores* of combis *CW*1 and *CW*2 that prevented them from starting when *SimTime* < 30 minutes.

3 Task A

Task A collected statistics on the average time needed to treat one *Patient* of each type, by running the simulation until 250 *Patients* had arrived (which represented 1 day).

Patient	Task A	Task B	Task C
All types	174	184	161
Type 1	237	252	164
Type 2	136	144	162
Type 3	252	266	175
Type 4	127	136	156
Avg. Std. Dev.	82	95	75
Closing time	13:48	14:30	13:52

Table 1: Averages for daily patient treatment time (min), daily standard deviation (min), and closing time.

The statistics in Table 1 are based on 100 replications (i.e., 100 days), each day allowing 250 *Patient* arrivals. These results show the average treatment time for all *Patients* and for each *Patient* type. Also shown is the average (over 100 days) of the daily standard deviation of all treatment times and the department average closing time.

4 Task B

Task B is an investigation into the effects of swapping one experienced *Doctor* (*D*1) from *DQ*1 with an inexperienced *Doctor* (*D*2) in *DQ*2. The swap would start when the queue for Ward 2, *CW*2*Q*, exceeded 20 *Patient*s and end when the queue *CW*2*Q* had fewer than 5 *Patients*. The times of *Doctor D*2 when working in *CW*l were increased by 20% due to the more complex cases.

Insights into the effect of this swap are provided by Figure 2, which shows the number of *Patients* waiting in the Casualty Ward queues *CW*1*Q* and *CW*2*Q* on a typical day for both ***Task A*** and ***Task B***.

Figure 2 shows that by 8:00AM (i.e., 30 minutes after the simulation started), the number of *Patients* arriving in *CW*2*Q* would already exceed 20 (for both ***Task A*** and ***Task B***).

Figure 2 also shows that in all cases, the number of *Patients* in *CW*2*Q* would drop to below 5 only at the end of the day.

This means that the doctor swap would start at 8:00AM, as soon as the *Doctors* arrived for work in the morning, and would last practically the entire day.

Table 1 shows that all statistics resulting from the swap in ***Task B***, i.e., the average daily patient treatment time, the average daily standard deviation, and the average closing time, were worse than those for ***Task A***.

This was not unexpected because, as shown in Figure 2, queue *CW*1*Q* in Task A already had more *Patients* waiting than *CW*2*Q*. Thus, moving an experienced *Doctor D*1 from *CW*1 (the Ward with the larger queue) to *CW*2 (the Ward with the smaller queue) in ***Task B*** was not a good idea. It could only make performance worse.

5 Task C

Task C investigated whether giving higher priority for service to patients who would return (a second time) to the same queues they had previously visited would reduce the daily standard deviation of treatment times.

In STROBOSCOPE, this change in *Patient* priority was implemented in a straightforward manner by changing the *Discipline* attribute for the queues for the Casualty Wards, for X-rays, and for plaster work (i.e., *CW*1*Q*, *CW*2*Q*, *XRQ*, *PLQ*), from the default *TimeIn* (i.e., *FIFO* based on the most recent times that *Patients* entered each queue) to *ResNum* (i.e., where *Patients* with a smaller *resource number*, i.e., those that arrived to the *system* earlier, would be sent to the front of the queue).

Table 1 shows that, as expected, the change in *Patient* priority reduced the average treatment time for *Patients P*1 and *P*3 (who made repeated visits to queues) and increased the average treatment time for *Patients P*2 and *P*3 (the easier cases) who visited the queues once.

The overall objective of ***Task C*** was achieved. The daily standard deviation of treatment times was reduced from 82 to 75 minutes.

6 Conclusion

The STROBOSCOPE simulation results in Table 1 are very close to previous solutions, e.g., [3] and [4]. Figure 2 is new and provides additional insights into the system.

The simulation model presented here requires only the basic modelling capabilities of STROBOSCOPE and is thus suitable for introductory educational purposes.

The network in Figure 1 used the standard drag-and-drop GUI elements implemented in Visio. Right-clicking the nodes and links in the GUI would activate dialog boxes (provided by a C++ add-on) that allowed the definition of node and link attributes as needed.

The GUI was used to create all the required STROBOSCOPE simulation model statements (text), which were then sent to STROBOSCOPE to perform the simulations.

References

[1] Breitenecker F. Comparison 6: Emergency Department-Follow-up Treatment - Definition. Simulation Notes Europe SNE. 1992; 2(3): 30.

[2] STROBOSCOPE Simulation System Software. Retrieved from www.STROBOSCOPE.org. Sept. 20, 2025.

[3] Ölsböck K, Noya1 M, Einzinger P. An Agent-based Approach to ARGESIM Benchmark C6 ‘Emergency Department – Follow-up Treatment’ using Anylogic. Simulation Notes Europe SNE. 2013; 23(3-4): 193-194.

[4] Pllana S, Fahringer T. An OO Programmed Approach to ARGESIM Comparison C6 ‘Emergency Department’ with CSIM. Simulation Notes Europe SNE. 2004; SNE 14(40), 2004, 33.

EUROSIM Contact 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 modelling and simulation in industry, research, and development – by publications and conferences.

EUROSIM members are national simulation societies and regional or international societies and groups dealing with modelling and simulation.

Full Members are ASIM, CEA-SMSG, CSSS, DBSS, KA-SIM, LIOPHANT, LSS, PTSK, NSSM, SIMS, SLOSIM, UKSIM. Observer Members are ALBSIM and ROMSIM. Former Members (societies in re-organisation) are: CROSSIM, FRANCOSIM, HSS, ISCS.

EUROSIM is governed by a Board consisting of one representative of each member society, president, past president, and **SNE** representative.

Each year a major **EUROSIM** event takes place, as the **EUROSIM CONGRESS** organised by a member society, **SIMS EUROSIM Conference**, and **MATHMOD Vienna Conference (ASIM)**.

The 12th EUROSIM Congress 2026 will be organized by **LIOPHANT** in Genova, September 2026.

Furthermore, **EUROSIM Societies** organize local conferences, and **EUROSIM** co-operates with the organizers of **I3M Conference** and **WinterSim Conference Series.**

Contact Information

www.eurosim.info

President:
Francesco Longo (LIOPHANT)
francesco.longo@unical.it

SNE – Simulation Notes Europe is EUROSIM's membership journal with peer-reviewed scientific contributions about all areas of modelling and simulation, including new trends as big data, cyber-physical systems, etc. The EUROSIM societies distribute **e-SNE** in full version to their members as official membership journal.

SNE has also become a post-conference publication journal for the EUROSIM societies, publishing revised, improved or extended versions of the conference publications.

The basic version of **e-SNE** is available with open access (Creative Commons license CC BY). Publishers are **ASIM, ARGESIM** and **EUROSIM.**

www.sne-journal.org

SNE-Editor:
Felix Breitenecker (ASIM)
eic@sne-journal.org

EUROSIM Member Societies

ASIM - German Simulation Society
Arbeitsgemeinschaft Simulation

ASIM is the association for simulation in the German speaking area, servicing mainly Germany, Switzerland and Austria.

President
Oliver Rose, *oliver.rose@unibw.de*

Contact Information

www.asim-gi.org

info@asim-gi.org

ASIM – Office Germany, Univ. Bundeswehr Munich, Inst. für Technische Informatik, Tobias Uhlig, Werner-Heisenberg Weg 39, 85577 Neubiberg, Germany

ASIM – Office Austria, dwh Simulation Services,
F. Breitenecker, N. Popper, Neustiftgasse 57-59,
1070, Wien, Austria

CEA-SMSG – Spanish Modelling and Simulation Group

CEA is the Spanish Society on Automation and Control. The association is divided into national thematic groups, one of which is centered on Modeling, Simulation and Optimization (**CEA-SMSG**).

Representative
Emilio Jiménez *emilio.jimenez@unirioja.es*

Contact Information

www.ceautomatica.es/
modelado-simulacion-y-optimizacionl

simulacion@ceautomatica.es

CSSS – Czech and Slovak Simulation Society

CSSS is the Simulation Society with members from the two countries: Czech Republic and Slovakia.

President
Michal Štepanovský
michal.stepanovsky@fit.cvut.cz

Contact Information

cssim.cz

michal.stepanovsky@fit.cvut.cz

CSSS – Český a Slovenský spolek pro simulaci systémů, Novotného lávka 200/5,
11000 Praha 1, Česká republika

DBSS – Dutch Benelux Simulation Society

DBSS was founded in July 1986 in order to create an organisation of simulation professionals within the Dutch language area.

President
M. Mujica Mota, *m.mujica.mota@hva.nl*

Contact Information

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

KA-SIM Kosovo Simulation Society

The Kosova Association for Modeling and Simulation (**KA-SIM**) is closely connected to the University for Business and Technology (UBT) in Kosovo.

President
Edmond Hajrizi, *ehajrizi@ubt-uni.net*

Contact Information

www.ubt-uni.net

ehajrizi@ubt-uni.net

Dr. Edmond Hajrizi, Univ. for Business and Technology (UBT), Lagjja Kalabria p.n., 10000 Prishtina, Kosovo

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.

President
Marina Massei, *massei@itim.unige.it*

Contact Information

www.liophant.org
info@liophant.org

LIOPHANT Simulation, c/o Marina Massei,
DIPTEM University of Genoa, Savona Campus,
via Cadorna 2, 17100 Savona, Italy.

LSS – 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.

President
Artis Teilans, *Artis.Teilans@rta.lv*

Contact Information

www.itl.rtu.lv/imb/

Artis.Teilans@rta.lv, Egils.Ginters@rtu.lv

LSS, Dept. of Modelling and Simulation, Riga Technical University, Kalku street 1, Riga, LV-1658, Latvia

NSSM – National Society for Simulation Modelling (Russia)

NSSM (Национальное Общество Имитационного Моделирования – НОИМ) was officially registered in Russia in 2011.

President
R. M. Yusupov, *yusupov@iias.spb.su*

Contact Information

www.simulation.su

yusupov@iias.spb.su

NSSM / R. M. Yusupov, St. Petersburg Institute of Informatics and Automation RAS,
199178, St. Petersburg, Russia

PTSK – Polish Society for Computer Simulation

PTSK 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.

President
Tadeusz Nowicki,
Tadeusz.Nowicki@wat.edu.pl

Contact Information

www.ptsk.pl

leon@ibib.waw.pl

PSCS, ul. Gen. Witolda Urbanowicza 2, pok. 222,
00-908 Warszawa 49, Poland

SIMS – Scandinavian Simulation Society

SIMS is the Scandinavian Simulation Society with members from the five Nordic countries Denmark, Finland, Norway, Sweden and Iceland.

President
Tiina Komulainen,
tiina.komulainen@oslomet.no

Contact Information

www.scansims.org

vadime@wolfram.com

Vadim Engelson, Wolfram MathCore AB,Teknikringen 1E, 58330, Linköping, Sweden

SLOSIM – Slovenian Society for Simulation and Modelling

The Slovenian Society for Simulation and Modelling was established in 1994. It promotes modelling and simulation approaches to problem solving in industrial and in academic environments by establishing communication and cooperation among corresponding teams.

President
Goran Andonovski,
goran.andonovski@fe.uni-lj.si

Contact Information

www.slosim.si

slosim@fe.uni-lj.si, vito.logar@fe.uni-lj.si

SLOSIM, Fakulteta za elektrotehniko, Tržaška 25,
1000, Ljubljana, Slovenija

UKSIM - United Kingdom Simulation Society

The UK Modelling & Simulation Society (**UKSim**) is the national UK society for all aspects of modelling and simulation, including continuous, discrete event, software and hardware.

President
David Al-Dabass,
david.al-dabass@ntu.ac.uk

Contact Information

uksim.info

david.al-dabass@ntu.ac.uk

UKSIM / Prof. David Al-Dabass, Computing & Informatics, Nottingham Trent University, Clifton lane, Nottingham, NG11 8NS, United Kingdom

Observer Members

ROMSIM – Romanian Modelling and Simulation Society

Contact Information

florin_h2004@yahoo.com
ROMSIM / Florin Hartescu, National Institute for Research in Informatics, Averescu Av. 8 – 10,
011455 Bucharest, Romania

ALBSIM – Albanian Simulation Society

Contact Information

kozeta.sevrani@unitir.edu.al

Albanian Simulation Goup, attn. Kozeta Sevrani, University of Tirana, Faculty of Economy ,
rr. Elbasanit, Tirana 355, Albania

ASIM Books – ASIM Book Series – ASIM Buchreihen

Monographs

Simulation-based Optimization: Industrial Practice in Production and Logistics
Lothar März, Markus Rabe, Oliver Rose (Eds.); to appear; ASIM Mitteilung 191

Energy-related Material Flow Simulation in Production and Logistics.
S. Wenzel, M. Rabe, S. Strassburger, C. von Viebahn (Eds.); Springer Cham 2023, print ISBN 978-3-031-34217-2, eISBN 978-3-031-34218-9, DOI 10.1007/978-3-031-34218-9, ASIM Mitteilung 182

Kostensimulation - Grundlagen, Forschungsansätze, Anwendungsbeispiele
T. Claus, F. Herrmann, E. Teich; Springer Gabler, Wiesbaden, 2019; Print ISBN 978-3-658-25167-3; Online ISBN 978-3-658-25168-0; DOI 10.1007/978-3-658-25168-0; ASIM Mitteilung 169

Proceedings*

Simulation in Produktion und Logistik 2025 - Tagungsband 21. ASIM-Fachtagung Simulation in Produktion und Logistik,
Dresden, 24. bis 26. September 2025, S. Rank, M. Kühn, T. Schmidt (Hrsg.)
ASIM Mitteilung 194, e-ISBN 978-3-86780-809-5, DOI 10.25368/2025.233, Technische Uni Dresden / Qucosa Publ., 2025

Tagungsband ASIM Workshop 2025 - GMMS/STS - ASIM Fachgruppenworkshop 2025, DLR Oberpfaffenhofen
W. Commerell, U. Durak, D. Zimmer (Hrgs.), ARGESIM Report 48; ASIM Mitteilung AM 193
ISBN ebook 978-3-903347-66-3, DOI 10.11128/arep.48, ARGESIM Verlag Wien, 2025

Tagungsband Kurzbeiträge ASIM SST 2024 -27. ASIM Symposium Simulationstechnik, Univ. Bundeswehr München, München/Neubiberg, Sept. 2024, O. Rose, T. Uhlig (Hrgs.), ARGESIM Report 46; ASIM Mitteilung AM 189
ISBN ebook 978-3-903347-64-9, DOI 10.11128/arep.46, ARGESIM Verlag Wien, 2024

Tagungsband Langbeiträge ASIM SST 2024 -27. ASIM Symposium Simulationstechnik, Univ. Bundeswehr München München/Neubiberg, Sept. 2024,O. Rose, T. Uhlig (Hrgs.), ARGESIM Report 47; ASIM Mitteilung AM 190
ISBN ebook 978-3-903347-65-6, DOI 10.11128/arep.47, ARGESIM Verlag Wien, 2024

Simulation in Production and Logistics 2023 – Tagungsband 20. ASIM Fachtagung Simulation in Produktion und Logistik
TU Ilmenau, September 2023; S. Bergmann, N. Feldkamp, R. Souren, S. Straßburger (Hrsg.);
ASIM Mitteilung 187; ISBN ebook 978-3-86360-276-5, DOI: 10.22032/dbt.57476, Universitätsverlag Ilmenau, 2023

Proceedings Langbeiträge ASIM Workshop 2023 - STS/GMMS/EDU - ASIM Fachgruppenworkshop 2023
Univ. Magdeburg, März 2023; C. Krull; W. Commerell, U. Durak, A. Körner, T. Pawletta (Hrsg.)
ARGESIM Report 21; ASIM Mitteilung 185; ISBN ebook 978-3-903347-61-8, DOI 10.11128/arep.21, ARGESIM Verlag, Wien, 2023

Kurzbeiträge & Abstract-Beiträge ASIM Workshop 2023 STS/GMMS/EDU - ASIM Fachgruppenworkshop 2023
Univ. Magdeburg, März 2023; C. Krull; W. Commerell, U. Durak, A. Körner, T. Pawletta (Hrsg.)
ARGESIM Report 22; ASIM Mitteilung 186; ISBN ebook 978-3-903347-62-5, DOI 10.11128/arep.22, ARGESIM Verlag, Wien, 2023

Proceedings Langbeiträge ASIM SST 2022 -26. ASIM Symposium Simulationstechnik, TU Wien, Juli 2022
F. Breitenecker, C. Deatcu, U. Durak, A. Körner, T. Pawletta (Hrsg.), ARGESIM Report 20; ASIM Mitteilung AM 180
ISBN ebook 978-3-901608-97-1, DOI 10.11128/arep.20, ARGESIM Verlag Wien, 2022

Book Series Fortschrittsberichte Simulation – Advances in Simulation *

Beschleunigung von Diskret-Ereignisorientierten Simulationsstudien unter Verwendung des DEVS-Formalismus auf HPC-Systemen. D. Jammer, FBS 42, e-ISBN 978-3-903347-43-4, DOI 10.11128/fbs.42, ARGESIM/ASIM Publ. Vienna, 2026.

An Architecture for Model Behavior Generation for Multiple Simulators. H. Folkerts, FBS 42
e-ISBN 978-3-903347-42-7, DOI 10.11128/fbs.32, ARGESIM Publ. Vienna, 2024

Das Verhalten von Transuranelementen in Erdböden - Theorie, Beprobung und radiochemische Analysen. K. Breitenecker, FBS 41; ISBN ebook 978-3-903347-41-0, DOI 10.11128/fbs.41, 2024; ISBN print 978-3-901608-99-5, 2010/2024; ARGESIM Publ. Vienna

Aufgabenorientierte Multi-Robotersteuerungen auf Basis des SBC-Frameworks und DEVS. B. Freymann, FBS 40
ISBN ebook2020_978-3-903347-40-3, DOI 10.11128/fbs.40, ARGESIM Publ. Vienna, 2022

Cooperative and Multirate Simulation: Analysis, Classification and New Hierarchical Approaches. I. Hafner, FBS 39
ISBN ebook978-3-903347-39-7, DOI 10.11128/fbs.39, ARGESIM Publ. Vienna,2022

Die Bedeutung der Risikoanalyse für den Rechtsschutz bei automatisierten Verwaltungsstrafverfahren. T. Preiß, FBS 38
ISBN ebook 978-3-903347-38-0, DOI 10.11128/fbs.38, ARGESIM Publ. Vienna,2020

Methods for Hybrid Modeling and Simulation-Based Optimization in Energy-Aware Production Planning. B. Heinzl, FBS 37
ISBN ebook 978-3-903347-37-3, DOI 10.11128/fbs.37, ARGESIM Publ. Vienna,2020;

Konforme Abbildungen zur Simulation von Modellen mit verteilten Parametern. Martin Holzinger, FBS 36
ISBN ebook 978-3-903347-36-6, DOI 10.11128/fbs.36, ARGESIM Publ. Vienna, 2020

Fractional Diffusion by Random Walks on Hierarchical and Fractal Topological Structures. G. Schneckenreither, FBS 35
ISBN ebook 978-3-903347-35-9, DOI 10.11128/fbs.35, ARGESIM Publ. Vienna, 2024

A Framework Including Artificial Neural Networks in Modelling Hybrid Dynamical Systems. Stefanie Winkler, FBS 34
ISBN ebook 978-3-903347-34-2, DOI 10.11128/fbs.34, ARGESIM Publ. Vienna, 2020

Modelling Synthesis of Lattice Gas Cellular Automata and Random Walk and Application to Gluing of Bulk Material. C.Rößler, FBS 33
ISBN ebook 978-3-903347-33-5, DOI 10.11128/fbs.33, ARGESIM Publ. Vienna, 2021

Combined Models of Pulse Wave and ECG Analysis for Risk Prediction in End-stage Renal Desease Patients. S. Hagmair, FBS 32
ISBN ebook 978-3-903347-32-8, DOI 10.11128/fbs.32, ARGESIM Publ. Vienna, 2024

Mathematical Models for Pulse Wave Analysis Considering Ventriculo-arterial Coupling in Systolic Heart Failure. S. Parragh, FBS 31
ISBN ebook 978-3-903347-31-1, DOI 10.11128/fbs.31, ARGESIM Publ. Vienna, 2024

Variantenmanagement in der Modellbildung und Simulation unter Verwendung des SES/MB Frameworks. A. Schmidt, FBS 30; ISBN ebook 978-3-903347-30-4, DOI 10.11128/fbs.30, ARGESIM Verlag, Wien 2019

12th EUROSIM Congress
September 21-23, 2026
Genova, Italy

The 12th edition of the EUROSIM Congress will be held in Genova, Italy, organized by the Liophant Simulation Group. This edition's congress is dedicated to the theme of

M&S & AI: Opportunities and Challenges for Present & Future

Genova has a rich maritime heritage and dynamic role as a hub for logistics, ICT, art, finance, business and fashion.

Authors are welcome to submit their contributions to the following, but not limited to areas such as: Agent-Based Simulation | System Dynamics Simulation | Discrete Event Simulation | Supply Chain Management, Logistics & Transportation | Industrial Case Studies Complex, Intelligent, Adaptive and Autonomous Systems | Data Science for Simulation | Environment and Sustainability Applications | Artificial Intelligence | Agentic AI and Simulation | Military Application | Hybrid Simulation | Manufacturing Applications | Simulation-Optimization | Tutorials / workshops from academics and industry, PhD Track and many more

Special ASIM Tracks and reduced fee for ASIM members

Researchers and practitioners are encouraged to submit their original contributions as:

Discussing paper (2-page extended abstract); peer-reviewed; accepted for presentation

Full paper (max 8 pages); peer-reviewed; accepted for presentation; published in the Congress proceedings; considered for journal publication.

Deadlines

- Special Session/Open Track Proposal Closure: April 10th, 2026
- Draft Paper Submission Deadline: April 15th, 2026
- Notification of Acceptance & Review Reports: May 15th, 2026
- Camera-ready Paper Submission Deadline: June 15th, 2026

www.msc-les.org/eurosim2026/

www.sne-journal.org

www.argesim.org