SNE SIMULATION NOTES EUROPE

Journal on Developments and Trends in Modelling and Simulation

EUROSIM Scientific Membership Journal

Vol. 35 No.4, December 2025 ISSN Online 2306-0271 DOI 10.11128/sne.35.4.1075

Letter of the EUROSIM President

Dear Members, Colleagues and Friends of the EUROSIM Federation,

It is with a heavy heart that I write to you today, following the passing in September of our dear friend and distinguished colleague Agostino Bruzzone of the University of Genoa. His leadership, his commitment to our community of modelling & simulation scholars and practitioners and his personal warmth and generosity will be profoundly missed. Under his presidency the Federation maintained the high standards of scientific and professional excellence for which we are known and his legacy in terms of vision and friendship will always remain with us.
In the spirit of continuity and renewal, I have accepted the honour of serving as President of EUROSIM for the period until September 2026, when the Congress of EUROSIM will be held in Genoa from 21-23 September 2026. I am deeply aware of the trust the Board and our member societies have placed in me and I pledge to devote myself to our service and to our shared mission.

Looking ahead, my purpose is to build on the firm foundations laid by Agostino and his predecessors, while seeking to strengthen EUROSIM's role as Europe's premier forum for modelling and simulation in industry, research and development. To that end, I will try to enhance collaboration and connectivity across our network of member societies so that ideas, tools and experience flow more freely between countries and between academia and industry.
I will also try to promote stronger links between industrial users, service providers and academic developers (e.g. identification of possible joint research projects opportunities) thereby increasing visibility for the practical value of simulation in decision-making, product development and systems engineering. Together with Agostino's group, we will prepare for the 2026 Congress in Genoa, ensuring that it becomes a fruitful event rich in technical content, vibrant in networking, and forward-looking in topics (especially around digital twins, AI-enabled simulation, hybrid human-machine modelling) so that EUROSIM remains at the forefront of change.
I also think that keeping working on strengthening publications and dissemination, in particular via our journal Simulation Notes Europe (as already done by my predecessors), is something relevant so that our community's work not only appears in proceedings but leads to broader uptake and recognition in science and industry.
I believe that by aligning these priorities with the contributions of our member societies, we will not only honour Agostino's memory, but advance EUROSIM to the next stage of its evolution — a Federation that is more connected, more visible and more impactful than ever.

Your involvement matters.
In that spirit, I warmly invite you — as members, colleagues, partners — to share your ideas, to volunteer your time and expertise, and to help shape our shared agenda. Whether this means proposing special sessions for the 2026 congress, nominating emerging talent for awards, or initiating industry-academia collaboration platforms, your contribution is invaluable.

It is a privilege to serve as President of EUROSIM during this transition period. Together, let us honour our past, live our present with commitment and purpose, and embrace the future with confidence and curiosity.

With sincere thanks for your support,
Francesco Longo, President of EUROSIM
Until September 2026 !

Francesco Longo, PhD
Full Professor of Industrial Systems Engineering
Director of Modeling & Simulation Center - Laboratory of Enterprise Solutions
DIMEG, University of Calabria
Via Ponte Pietro Bucci, Cubo 45C - Third floor, Rende (CS) - Italy
T +39 0984 494891 | f.longo@unical.it

Editorial

Dear Readers, *This last issue of SNE Volume 35, SNE 35(4), again presents as first publication a Benchmark solution: ARGESIM Benchmark C22 ‘Non-standard Queuing Policies’ with STROBOSCOPE, and we hope for more solutions with this simulation environment, and with other tools and alternative approaches.*

And we continue the post-conference publications from ASIM Symposium 2024 (Munich), and we start with the post-conference publications from ASIM’s Workshop 2025 at DLR (April 2025). The topics of the contributions show the broad range of modelling and simulation: a drilling force model in multibody system, the system identification of an omnidirectional test vehicle, an evaluation method for simulation results, a procedural model for selecting decision support methods in production and logistics, and a clustering-based approach to identify similar driving sections. For 2026 we are planning also post-conference publications from MATHMOD 2025 (Vienna), ASIM Production and Logistics 2025 (Dresden), and from other European simulation conferences.

We are happy that Liophant, the Italian member society of EUROSIM, will organize the next EUROSIM Congress in Italy: the 12th EUROSIM Congress ***EUROSIM 2026****, September 21-23, 2026, Genova, Italy (together with I3M 2026. But we also have to announce sadly, that Agostino Bruzzone, well-known European simulationist, Liophant president and EUROSIM president, has suddenly passed away – see the obituary in this issue. Francesco Longo from Liophant and Univ. of Calabria has taken over EUROSIM presidency and will organize EUROSIM 2026 in memoriam of Agostino Bruzzone (see also his Letter of the President).*

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 the years 2025 and 2026: WinterSim 2025 December in Seattle, further conferences of the EUROSIM societies in 2026, e.g. SIMS 2026 in September in Eskilstuna, Sweden, and the EUROSIM Congress 2026 and I3M 2026 in September in Genova.

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

Contents SNE 35(4)

e-SNE 35(4), DOI 10.11128/sne.35.4.1075

www.sne-journal.org/sne-volumes/volume-35

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)

STROBOSCOPE Models for ARGESIM Benchmark C22 ’Non-standard Queuing Policies’.
P. G. Ioannou, V. Likhitruangsilp 171

A Drilling Force Model for Use in Multibody System Simulation Environments. *R. Reiser* 179

System Identification of an Omnidirectional Test Vehicle for Model-Based Function Design.
B. Carstens, M. Göllner, T. Li, X. Liu-Henke 187

Development of an Evaluation Method for Simulation Results in the Context of Industrial Symbiosis: A Target System for Increasing Resource Efficiency and Resilience.
B. Vollack, M. Boelcke, T. Schmidt 195

Conceptualization of a Procedural Model for Selecting Decision Support Methods in Production and Logistics.
K. Langenbach 199

Segmentation of Bus Driving Data: A Clustering-based Approach to Identify Similar Driving Sections.
A. Schniertshauer, S. Angerer, A. Grabow, M. Schlick 203

EUROSIM Societies & ARGESIM/SNE Short Info . N1 – N3

Obituary Agostino Bruzzone.. N4

Conferences EUROSIM / ASIM .... Covers

Front Cover: Map of Europe with EUROSIM Congress sites

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 (documentation of ARGESIM Benchmarks on modelling approaches and simulation implementations); modelling and simulation in and for education, simulation-based e-learning; society information and membership information for EUROSIM members (Federation of European Simulation Societies and Groups).

Editor-in-Chief: Felix Breitenecker, 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 - 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):

STROBOSCOPE Models for ARGESIM Benchmark C22 ’Non-standard Queuing Policies’

Photios G. Ioannou[1*], Veerasak Likhitruangsilp[2]

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

[2]Center of Digital Asset Management for Sustainable Development (CDAM), Dept of Civil Engineering,
Faculty of Engineering, Chulalongkorn University, Bangkok, Thailand

SNE 35(4), 2025, 171-177, DOI: 10.11128/sne.35.bn22.10751
Submitted: 2025-09-23
Received Improved: 2025-11-13; Accepted: 2025-11-15
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 a queueing system with four servers, each with its own dynamically changing queue, where entities are initially assigned to each of four queues based on minimum queue length. In addition to the *base* case, alternatives examined include *jockeying* from one queue to a shorter queue, *reneging* (leaving the system) if the waiting time exceeds a limit, and serving entities based on *classes*.

Introduction

ARGESIM Benchmark C22 [1] investigates how to model FIFO queues that include additional dynamic behaviour. In the base system, arriving entities can be served by one of four servers, each with its own queue, where entities choose the shortest queue to join upon arrival. In a jockeying system, queued entities observe the dynamic size of all queues and switch to a shorter queue when possible. In a reneging system, queued entities grow impatient if their waiting time exceeds a set limit and depart from the system without being served. In a classing system, arriving entities are assigned to different classes and are served in groups according to their class.

These four queueing systems were modelled in STROBOSCOPE [2] (an acronym for State and Resource-Based Simulation of Construction Processes). STROBOSCOPE is a free-to-use general-purpose discrete-event simulation language and system co-developed by the first author. Its simulation models use graphical networks (similar to *activity cycle diagrams*) to communicate easily to others the main modelling elements (queues, activities, links, etc.).

The complete models are defined in text files with statements written in the STROBOSCOPE language. These statements define the resources and their properties, the attributes and behaviour of the network modelling elements, the logic by which resources and network elements interact, the definition of auxiliary objects (such as for the collection of custom statistics), the initialization of queues with resources, the overall control of the simulation, etc.

The design of STROBOSCOPE is based on three-phase activity scanning that can easily model the complex resource interactions that characterize *cyclic* operations. In STROBOSCOPE, there is no distinction between resources that serve (servers or scarce resources) and those served (customers or moving entities).

Resources can be

(a) generic resources (i.e., without attributes),
(b) characterized resources that are objects that belong to types (general classes) and to specific subtypes (subclasses with specific properties), and
(c) compound resources that have types (general classes) and which can be *assembled* by combining any number of other resources.

For example, the resources “bus” and its “passengers” can be assembled into one compound resource object called a “loaded bus” that can flow through queues and activities as one object. Each characterized and compound resource can have its own dynamic properties.

During simulation, resources spend time in queues or in activity instances. An arriving entity, for example, enters a queue where it waits conditionally until it can be served. When the entity reaches the front of the queue, and when the associated server becomes free, a new instance of the service activity can be created and draw both the entity and the server from their respective queues. The duration of the service activity instance is the required service time.

The relative order in which activities create instances and draw resources from preceding queues (if they could start at the *same* simulation time) is controlled by their *priorities* which are evaluated dynamically while the simulation runs. Similarly, the order in which resources are arranged in a queue is controlled by a user-defined *discipline* that is evaluated dynamically each time a new resource enters a queue. The *order*, *filter*, and *number* of resources drawn from a queue by a link are also dynamic.

The following models make heavy use of STROBOSCOPE *filters*. Filters are powerful objects that can be attached to queues or activities and that define the criteria for creating *dynamic subsets* of the resources that are currently in a queue or an activity instance. Moreover, two filters can call each other and create complex constructs that filter two queues at the same time.

For example, two coupled filters working together can create a subset of all *entities* in a single queue (irrespective of which actual physical queue they reside in) that can be served by the subset of all currently idle *servers*.

The entities for all the models presented below wait in a single queue node called *Q* (instead of four separate queue nodes, *Q*1, *Q*2, *Q*3, *Q*4). Similarly, when idle, the four servers wait in a single queue node called SQ (instead of four separate queue nodes, SQ1, SQ2, SQ3, and *SQ*4). All models below utilize coupled filters to create dynamic subsets and match waiting entities with their corresponding servers.

There are several possible models for the C22 benchmark. The models below use a network with the fewest nodes and links, and as a result, require the most complex filters. These models are available from the authors.

1 Basic Queuing System

The STROBOSCOPE network for all simulation models (except for reneging) is shown in Fig. 1. This network includes only one queue *Q* for all waiting entities, one queue *SQ* where all idle servers reside, and one *Service* activity (instead of including four of each). The model uses coupled *filters* to match entities to their servers.

In the deterministic model of the base system, entities arrive every t_A=1, select the shortest queue from among four FIFO server queues, and store it in a *SaveProp*.

Each queue is served by a single server with service time t_S=4.5. Simulation stops after n_E=100 entities arrive.

The model defines two compound resource types:

```
COMPTYPE  Entity;    COMPTYPE  Server;
SAVEPROPS Entity ServerID;
```

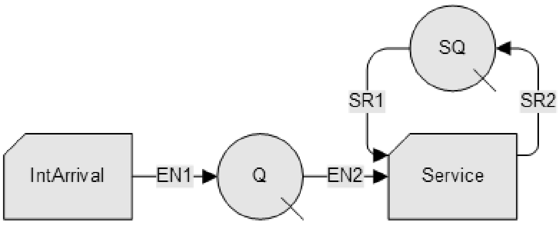


Figure 1: STRBOSCOPE Simulation Model Network.

The *SaveProp ServerID* of the resource *Entity* is assigned at runtime and stores the *ResNum* property (i.e., 1,2,3,4) of the *Entity’s* assigned *Server*.

The relative order of *events* is controlled by the priorities of activities. Combi activity *Service* has a higher priority to start and create an instance than *IntArrival*.

Thus, the order of events is as follows (important when ending and starting activity events occur at the same time).

1. *Service* ends: The idle *Server* is returned to *SQ*, and the *Entity* that was in *Service* is terminated.
2. *IntArrival* ends: An *Entity* is generated and is assigned a *Server* when it flows through link *EN*1 to queue *Q*.
3. *Service* starts: If any *Entities* in *Q* can be *matched* with an idle *Server* in *SQ*, then a new *Service* instance starts and draws the first matched Entity and its *Server*.
4. *IntArrival* starts: A new *IntArrival* instance is created.

The assignment of a *Server* to an arriving *Entity* in (2) considers *both* the number of *Entities* in *Q* already assigned to each *Server*, as well as the *Entity* being served by that *Server*. The counting of the *Entities* in *Q* that have already been assigned to each *Server* is done with filters.

The *matching* of *Entities* in *Q* with their *Servers* in (3) is done by the following two coupled filters as follows.

```
FILTER MatchedServers Server 1; /forward def
FILTER MatchedEntities Entity
   SQ.MatchedServers.Count;
VARIABLE ChoiceOfCursoredEntity
  'MatchedEntities.HasCursor?
   MatchedEntities.ServerID : EN2.ServerID';
FILTEREXP MatchedServers
   'ResNum==ChoiceOfCursoredEntity';
```

An instance of *Service* can start when the *ENOUGH* attributes of its incoming links *SR*1 and *EN*2 return the value *true* (i.e., any number greater than zero). Link *SR*2 has the default *ENOUGH,* which returns *true* when the preceding queue *SQ* is not empty.

The more important *ENOUGH* is that of link *EN*2 (shown below) that applies filter *MatchedEntities* to queue *Q* to create the subset of *Entities* that have a free *Server* in *SQ*. If that subset is not empty, then a new instance of *Service* is created.

```
ENOUGH EN2 'Q.MatchedEntities.Count';
```

The new instance of *Service* then draws resources through its incoming links. First, it draws through link *EN*2 the first *Entity* for which its *Server* is currently idle in queue *SQ*. And then it draws through link *SR*1 the matching *Server* (i.e., the one whose *ResNum* is the same as the *SaveProp ServerID* of the *Entity* already drawn).

```
DRAWWHERE EN2 'SQ.MatchedServers.Count';
DRAWWHERE SR1
    'ResNum==Service.Entity.ServerID';
```

1.1 Deterministic Basic Model Results

STROBOSCOPE can dynamically write data about the state of the simulation to files. Figure 2 shows an example Excel graph of the data in such a file with the *ID*s (i.e., *ResNum*) of the last 20 *Entities* vs. simulation time. STROBOSCOPE also has an *add-on* that can create Excel graphs of the contents of selected queues vs. simulation time. Such a graph for queue *Q* is shown in Figure 2.

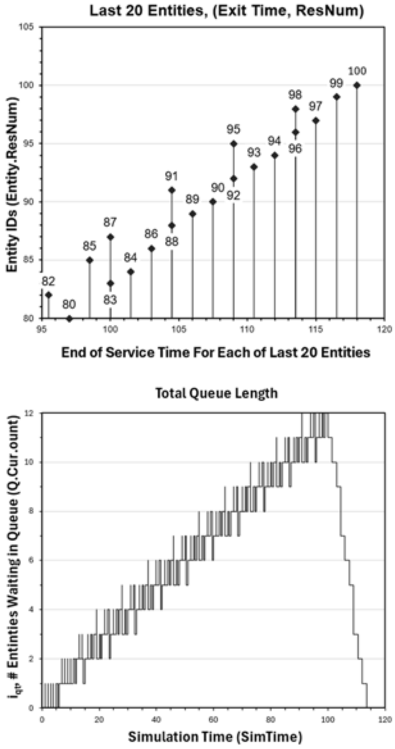


Figure 2: Outgoing *Entity IDs* (*ResNum*) and total length of queue *Q* for the base model.

Table 1 shows statistics about the length l_{qt} and the waiting time $t_{q,i}$ in *Q*, for the basic deterministic model.

	l_{qt}		$t_{q,i}$	
SimTime	**Avg**	**Max**	**Avg**	**Max**
118	5.68	13.00	6.70	13.50

Table 1: Basic Deterministic Model Statistics.

1.2 Stochastic Basic Model Results

For the stochastic version of the base model, the number of arriving *Entities* before simulation ends, and the durations of activities were as follows:

```
VARIABLE NE 500; /max arriving Entities
DURATION IntArrival Exponential[1]; /TA
DURATION Service Triangular[2.5,4.5,6.5];/TS
```

Table 2 shows the resulting statistics about the length l_{qt} and the waiting time $t_{q,i}$ in queue *Q*, from three runs.

		l_{qt}		$t_{q,i}$	
Run	**SimTime**	**Avg**	**Max**	**Avg**	**Max**
1	577.578	27.34	64	31.58	78.19
2	581.334	26.71	46	31.06	54.64
3	572.973	33.97	69	38.93	79.32

Table 2: Basic Stochastic Model Statistics.

1.3 Variant Order of Concurrent Events

In the deterministic model, several concurrent events can occur at the same time, and their relative order produces different simulation results.

For example, the following order of concurrent events occurs when activity *IntArrival* has a higher priority than activity *Service*:

1. *IntArrival* ends: An *Entity* is generated and is assigned a *Server* when it flows through link *EN*1 to queue *Q*.
2. *Service* ends: The idle *Server* is returned to queue *SQ*, and the *Entity* that was in *Service* is terminated.
3. *IntArrival* starts: A new instance is created.
4. *Service* starts: If any *Entities* in *Q* can be *matched* with an idle *Server* in *SQ*, then a new instance of *Service* starts and draws the first matched *Entity* and its *Server*.

In this variant order of concurrent events, an arriving *Entity* is assigned a *Server before* the departing *Entity* in the terminating instance of *Service* is released.

Thus, the departing *Entity* is counted as still being in *Service* at that *Server* (i.e., incorrectly) and changes the assignment of a *Server* to the arriving *Entity*. The results are shown in Figure 3. The statistics are identical to those in Table 2.

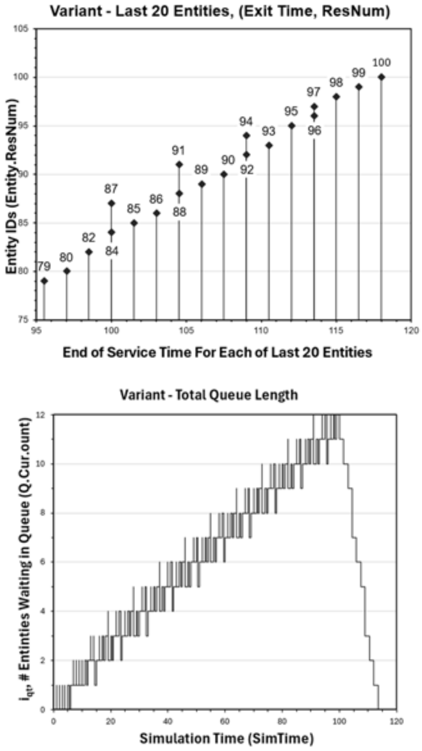


Figure 3: Outgoing Entity IDs (*ResNum*) and total queue *Q* length for the variant basic model.

1.4 Large Version with 40 Queues

STROBOSCOPE has preprocessing statement capabilities that enable scaling simulation models to any number of queues, activities, links, resources, etc.

Here, the *stochastic* model was scaled to a larger model as follows:

$n_Q = 40$, t_A = Exponential[0.1], $n_E = 5000$

The simulation results are shown in Table 3.

	l_{qt}		$t_{q,i}$	
SimTime	**Avg**	**Max**	**Avg**	**Max**
572.13	267.71	548	30.63	69.29

Table 3: Large Basic Stochastic Model Statistics

2 Jockeying Queues

2.1 Model Description

In a system with multiple queues, jockeying occurs when an entity leaves its current queue to join a shorter queue.

In these simulation models, all *Entities* wait in queue *Q*. Thus, jockeying simply requires changing their *SaveProp ServerID* from the currently assigned *Server* (with the longer queue) to another *Server* (with a shorter queue).

This is accomplished as follows, using *filters* that create the appropriate *logical subsets* of *Entities* in *Q* to make decisions and act (without the need to remove and reinsert any *Entity* objects in *Q*).

1. Jockeying may occur at the end of an instance of activity *Service* when a *Server* returns to queue *SQ*. This reduces the *Entities* assigned to that *Server* by one.
2. At that point, the last *Entity* assigned to each of the four *Servers* evaluates the minimum queue length for the other three *Servers*, and if it is shorter than its own queue by two *Entities*, then it changes its *SaveProp ServerID* to the target *Server* with the shorter queue.
3. In case of ties, the chosen target *Server* is the one with the smallest *Server ID* (i.e., 1,2,3,4).
4. The comparisons and assignments in step (2) are done in reverse order of *Server ID*s, starting with the logical queue (subset) for *Server* 4 and finishing with the logical queue (subset) for *Server* 1.

2.2 Deterministic Jockeying Model Results

The result graphs for the jockeying model are shown in Figure 4.

The first five and last five jockeying events are shown in Table 4. They are identical to the results shown in [3].

2.3 Stochastic Jockeying Model Results

The corresponding stochastic jockeying model was obtained by changing the duration of activities *IntArrival* and *Service*, like in the base model.

Table 5 shows results from three runs that were obtained automatically as part of the standard STROBOSCOPE reports.

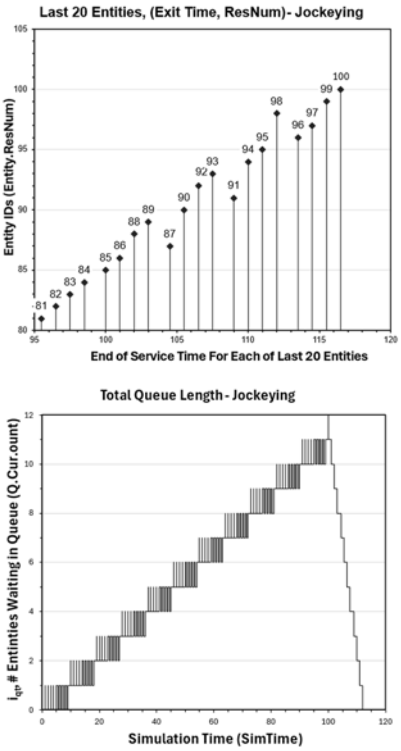


Figure 4: Outgoing ids and total queue length for the jockeying model.

SimTime, *t*	Entity ID	Source Q	Destin. Q
6.5	6	1	2
7.5	7	1	3
8.5	8	1	4
11.0	10	1	2
12.0	11	1	3
…	…	…	…
89.5	89	2	4
93.0	92	2	3
94.0	93	2	4
98.5	98	3	4
103.0	100	1	4

Table 4: First five and last five jockeying events.

		l_{qt}		$t_{q,i}$	
Run	**SimTime**	**Avg**	**Max**	**Avg**	**Max**
1	565.915	22.03	46	24.93	54.98
2	576.728	11.42	33	13.17	40.87
3	579.266	19.13	63	22.16	74.73

Table 5: Jockeying Stochastic Model Statistics.

3 Reneging Queues

3.1 Model Description

Reneging occurs when the waiting time for an *Entity* in *Q* exceeds its waiting time *tolerance* (e.g., $t_R = 9$), and it chooses to leave the system without being served.

The STROBOSCOPE network for the reneging model is shown in Figure 5. In this model, each terminating instance of *IntArrival* generates two resources, an *Entity* and a matching *Twin* with the same *ResNum*.

Twin is a new compound resource type that makes reneging easy to model. *Twin* is released to an instance of activity *MaxWaitTime*, with duration $t_R = 9$, and then to queue *TQ*. Combi activity *Reneg* can then *always* start and draw the *Twin* from *TQ*. *Reneg* also attempts to draw the matching *Entity* with the same *ResNum* that might renege from *Q*. Drawing the matching *Entity* can only occur if that *Entity* has not been served yet and is still in *Q*. Otherwise, no *Entity* is drawn.

For this reason, combi activity *Reneg* is assigned less priority than activity *Service*. Only *Twins* for the reneged *Entities* are collected in queue *RenTn*.

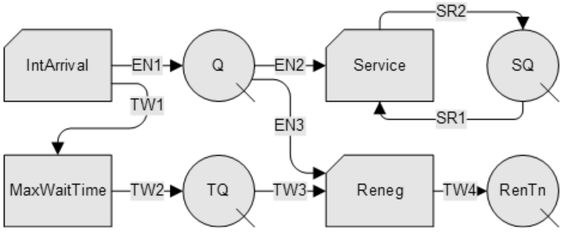


Figure 5: STRBOSCOPE Simulation Model Network.

3.2 Deterministic Reneging Model Results

Figure 6 shows a graph of the *ID*s (*ResNum*) of the last 40 outgoing *Entities* and the total length of *Q* vs *Simtime*. There were four reneging *Entitie*s, and they are shown by orange circles in Figure 6. They are also listed in Table 6.

SimTime, *t*	Entity ID	Server ID
77	68	1
86	77	2
89	80	1
104	95	1

Table 6: All four reneging events for deterministic model.

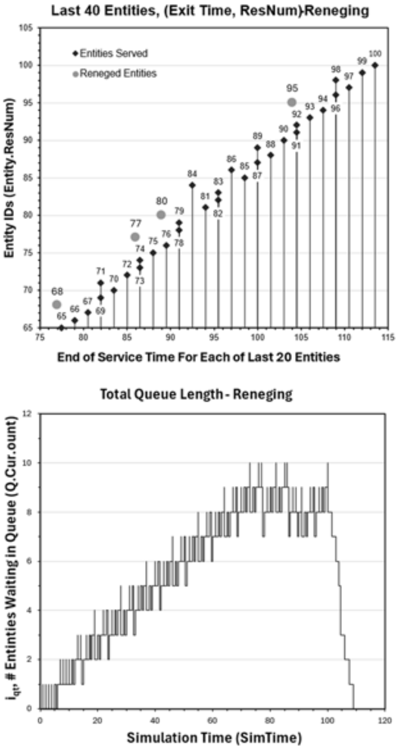


Figure 6: Outgoing *ID*s and total length of queue *Q* for the reneging model.

SimTime, *t*	Entity ID	Server ID
77	68	1
86	77	2
89	80	1
104	95	1

Table 6: All four reneging events for deterministic model.

3.3 Stochastic Reneging Model Results

The statistics results from three runs of the stochastic reneging model are shown in Table 7.

		l_{qt}		$t_{q,i}$	
Run	SimTime	Avg	Max	Avg	Max
1	544.171	4.26	14	4.64	9
2	482.019	5.83	16	5.62	9
3	517.067	4.94	15	5.10	9

Table 7: Reneging Stochastic Model Statistics.

4 Classing Queues

4.1 Model Description

In the deterministic classing queue model, an *Entity* is generated at the end of an instance of activity *IntArrival* and flows through link *EN*1 where it chooses the shortest *Server* queue (which it stores in its *SaveProp ServerID*), and an assigned *class* (i.e., a number from 1 to 5, which it stores in a new *SaveProp* named *Class*). Thus, the first *Entity* is assigned to *Class* 1, the second *Entity* to *Class* 2, etc. This assignment order repeats with *Entity* 6, which is again assigned to *Class* 1, etc.

Arriving *Entities* enter queue *Q* but are not served right away. Instead, service at all *Servers* starts when *SimTime* reaches 10. The order in which *Entities* are served is controlled by dynamic filters and the *SaveValue CurClass*, which is the number of the class called by an *operator*. Only the queued *Entities* whose *Class* equals *CurClass* can pass the filters and be served in FIFO order.

The value of *CurClass* starts at 5 and is decremented by 1 whenever there are no more queued *Entities* whose *Class* equals *CurClass*. Eventually, the value *CurClass* decreases to 1, and after that, the process is repeated by setting *CurClass* back to 5, etc.

Activity *Service* can start and create an instance whenever there are any *matched Entities* in queue *Q*. *Matched Entities* are those whose *Class* equals *CurClass* and which belong to a *logical queue* (i.e., a *filtered subset* of *Q*) that currently has a free *Server*. Each new instance of *Service* draws first the *matched Entity* from *Q* that is currently at the front of its logical queue and then draws the matching idle *Server* from queue *SQ*.

Each time a *matched Entity* is drawn from *Q* to a new instance of *Service*, the model counts the current number of *Entities* still in *Q* whose *Class* equals *CurClass*. When that number becomes zero, the value of *CurClass* is decremented by 1. When *CurClass* reaches the value 1, and there are no more *Entities* in queue *Q* with *Class* equal to 1, *CurClass* starts again at 5.

STROBOSCOPE supports user-defined *collectors* that can collect any calculated data and produce statistics. Five collectors were defined in this model to produce waiting time statistics for each of the five classes. Each time a *matched Entity* was drawn from *Q* to a new instance of *Service*, the model calculated its waiting time in *Q* and sent it to the statistics collector for that *Class*.

4.2 Deterministic Classing Model Results

Figure 7 shows a graph of the *ID*s (*ResNum*) of the last 20 outgoing *Entities* and the total length of *Q* vs *SimTime*.

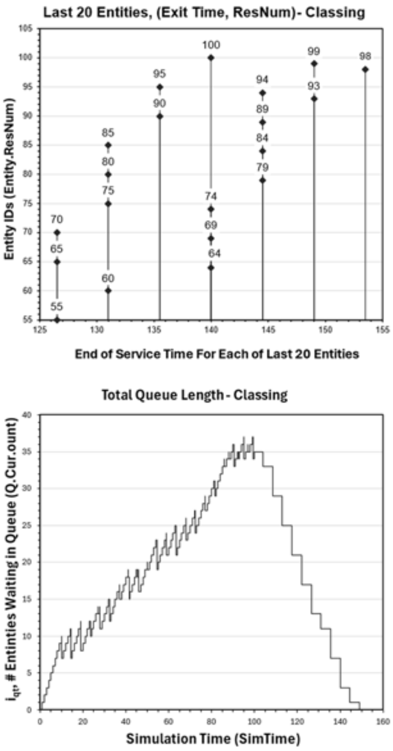


Figure 7: Outgoing ids and total queue length for the classing model.

Table 8 shows the average and maximum queue waiting times for the *Entities* in each of the five *Classes* that were compiled by the user-defined statistics collectors.

Class:	1	2	3	4	5
Avg:	30.05	26.45	20.85	32.55	30.52
Max:	62.50	60.50	51.50	71.50	67.00

Table 8: Queue waiting times per class (deterministic classing model).

4.3 Stochastic Classing Model Results

As in the base model, the stochastic classing model was produced from the deterministic model by changing the durations of the activities *IntArrival* and *Service* and increasing the number of generated *Entities* to 500. For the stochastic version, the *Class* of each arriving *Entity* was chosen with equal probability from 1 to 5.

Table 9 shows statistics from three runs about the length and the waiting times in queue *Q*.

		l_{qt}		$t_{q,i}$	
Run	**SimTime**	**Avg**	**Max**	**Avg**	**Max**
1	733.365	84.49	179.00	123.92	350.00
2	706.638	60.05	133.00	84.87	259.83
3	702.241	91.04	184.00	127.86	322.06

Table 9: Stochastic Classing Model Statistics.

Table 10 shows the average and maximum queue waiting times in *Q* for the *Entities* in each of the five *Classes* from three runs of the stochastic classing model.

Run	Class:	1	2	3	4	5
1	**Avg:**	146.2	151.3	124.2	58.1	131.2
	Max:	350.0	326.9	281.9	278.8	342.9
2	**Avg:**	64.1	78.1	80.4	112.6	91.8
	Max:	214.5	227.4	246.4	259.8	249.7
3	**Avg:**	142.9	150.4	103.5	96.0	142.3
	Max:	322.1	299.0	238.4	260.7	312.8

Table 10: Queue waiting times per class (stochastic classing model).

5 Conclusion

The STROBOSCOPE graphical and statistical results are very close to those produced by GPSS in [3]. The versatility of STROBOSCOPE queues, activities, and especially filters, makes it straightforward to model the non-standard queueing systems in ARGESIM Benchmark C22 by using compact simulation networks that are suitable for education.

References

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

[2] STROBOSCOPE Simulation System Software. Retrieved from www.stroboscope.org. Sept. 20, 2025.

[3] Junglas P, Pawletta Th. Solving ARGESIM Benchmark C22 'Non-standard Queuing Policies' with MatlabGPSS. Simulation Notes Europe SNE. 2019; 29(4): 199-205. DOI 10.11128/sne.29.bn22.10496

A Drilling Force Model for Use in Multibody System Simulation Environments

Robert Reiser*

Institute of Robotics and Mechatronics, German Aerospace Center (DLR),
Münchener Straße 20, 82234 Weßling, Germany; *firstname.lastname@dlr.de

SNE 35(4), 2025, 179-186, DOI: 10.11128/sne.35.tn.10752
Selected ASIM WS2025 Postconf. Publication: 2025-09-10
Rec. Revised: 2025-11-12; Accepted: 2025-11-14
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna
ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. In this work, a drilling-force model for the use in multibody system simulations is presented. The model is based on drilling-force calculations, collision detection, a compliant contact model, and a method for applying the forces to a multibody environment. The model can be used in two ways. First, it can be used to calculate drilling forces based on a given velocity and angular velocity of the drill. Second, it can be integrated into a model to interact as a body in a multibody scenario and be subjected to an external force and angular velocity.

Introduction

Drilling processes are widely used and occur in many different areas, such as manufacturing or construction. Several models exist to calculate the forces during a drilling process. However, they usually cannot be used for body interaction in a multibody context (e.g. if the drilling forces are to be applied to the robot in order to optimize the robot-assisted drilling process).

Therefore, in this work, a drilling-force model is presented that can be used in multibody simulations. The goal is to use it in system simulation environments. This allows the creation of multi-domain models. For example, the drilling model can be combined with the dynamics model of a robot and other models such as the control system or the motors of the robot. This can then be used to analyze the influence of the drilling process on the robot dynamics or energy consumption. The object-oriented and multi-domain modeling language *Modelica* [7] is used as system simulation environment.

Models are built based on the multibody components [9] from the *Modelica Standard Library* (MSL) [8].

The developed drilling-force model is based on drilling-force calculations, collision detection, a compliant contact model, and a method for applying the forces to the multibody environment.

In the next section, the state of the art is introduced. In Section 2, the developed drilling-force model is presented. Examples are given in Section 3. Finally, the results are discussed and future developments for the model are considered in Section 4.

1 State-of-the-Art

In this section, the basics are introduced. These include the forces in the drilling process and collision detection and response for multibody simulation environments.

1.1 Forces in the drilling process

Drilling is a machining process. Because the geometry of the cutting edges is known, it is classified as a *machining process with a geometrically defined cutting edge* (similar to turning and milling) [5]. There are several models for calculating the forces in the drilling process. Examples are the works of Dietrich (2016) [2] and Fritz and Schulze (2015) [5].

The most suitable parameters for a drilling process are often determined using table values [3]. In this work the model of Dietrich (2016) is applied [2]. It is used to calculate the cutting force F_c and the feed force F_f from a given rotational frequency n and feed per rotation f.

First, the feed per rotation per cutting edge f_z as well as the stress thickness h, the stress width b, and the stress cross-section A are calculated.
Inputs the are number of cutting edges z_E, the tip angle of the drill σ, and the drill diameter d (see Figure 1) [2].

$$\begin{aligned} f_z &= \frac{f}{z_E} && (1)\\ h &= f_z \cdot \sin(\frac{\sigma}{2}) && (2)\\ b &= \frac{d}{2 \cdot \sin(\frac{\sigma}{2})} && (3)\\ A &= h \cdot b && (4) \end{aligned}$$

The specific cutting force k_c depends on the stress thickness h and the material dependent parameters z and $k_{c1.1}$ [2]. The latter is the specific cutting force of a given material for $h = 1$ mm. There are also correction factors which are not considered in this work.

$$k_c = \frac{0.001^z}{(f_z \cdot \sin(\frac{\sigma}{2}))^z} \cdot k_{c1.1} \qquad (5)$$

The cutting force per cutting edge F_{cz}, the cutting force (for the entire drill) F_c, and the feed force F_f are then calculated as follows [2].

$$\begin{aligned} F_{cz} &= \frac{d \cdot f_z}{2} \cdot k_c && (6)\\ F_c &= z_E \cdot \frac{d \cdot f_z}{2} \cdot k_c && (7)\\ F_f &= z_E \cdot F_{cz} \cdot \sin(\sigma) && (8) \end{aligned}$$

1.2 Collision detection

The basis for calculating drilling forces in a multibody environment is the interaction between the bodies. This requires collision detection. Common algorithms for collision detection are the Gilbert-Johnson-Keerthi distance algorithm (GJK) [6] and the Minkowski Portal Refinement algorithm (MPR) [12]. Both algorithms provide the penetration depth of two colliding bodies.

There have been several approaches to enabling collision detection for the system simulation environment *Modelica*. Most of them extend *Modelica* with an external library for collision detection. An overview can be found in Reiser and Reiner (2023) [11].

1.3 Multibody contacts

Based on the collision detection, the next step is to prevent the bodies from intersecting. This is also known as collision response. There are two main approaches to handling multibody contacts: compliant (penalty-based) and non-smooth (constraint-based).

In this work, only compliant contacts are considered. Contact forces are calculated based on the penetration depth. [4]

A common model is the Kelvin-Voigt contact force model. It uses a spring-damper element to calculate the normal force F_N based on the penetration depth s [4]:

$$F_N = k \cdot s + d \cdot \dot{s} \qquad (9)$$

where k is the spring stiffness and d the damping factor. In addition, a friction force dependent on F_N acts in the tangential direction (not considered in this work).

Multiple works have dealt with compliant contacts between bodies in *Modelica* multibody environments. One example is Buse et al. (2023) [1]. Further works are listed in Reiser and Reiner (2023) [11].

2 A Drilling-force Model for Multibody Environments

The developed model for drilling forces in multibody simulation environments is presented in this section.

2.1 Overview

A key component of the model is collision detection. It is used to determine the drill diameter and the penetration depth of the drill. The model has two parts.

In the **kinematic model**, the rotational frequency and the feed per rotation are provided and the model calculates the drilling forces. These are not used for body interaction. However, they can be used in a simulation to check if certain forces are exceeded.

In contrast, the drilling forces in the **dynamics model** are applied to the bodies in the multibody environment. This allows, for example, force-controlled robot-assisted drilling processes to be simulated together with robot dynamics models.

2.2 Determination of the drilling diameters

The first step is to determine the drilling diameters during the drilling process. If the drill does not come out of the material on the opposite side during drilling, only the outer diameter d_O is relevant. Otherwise, the inner diameter d_I is also important.

Collision detection is used to determine the drilling diameters during the process.

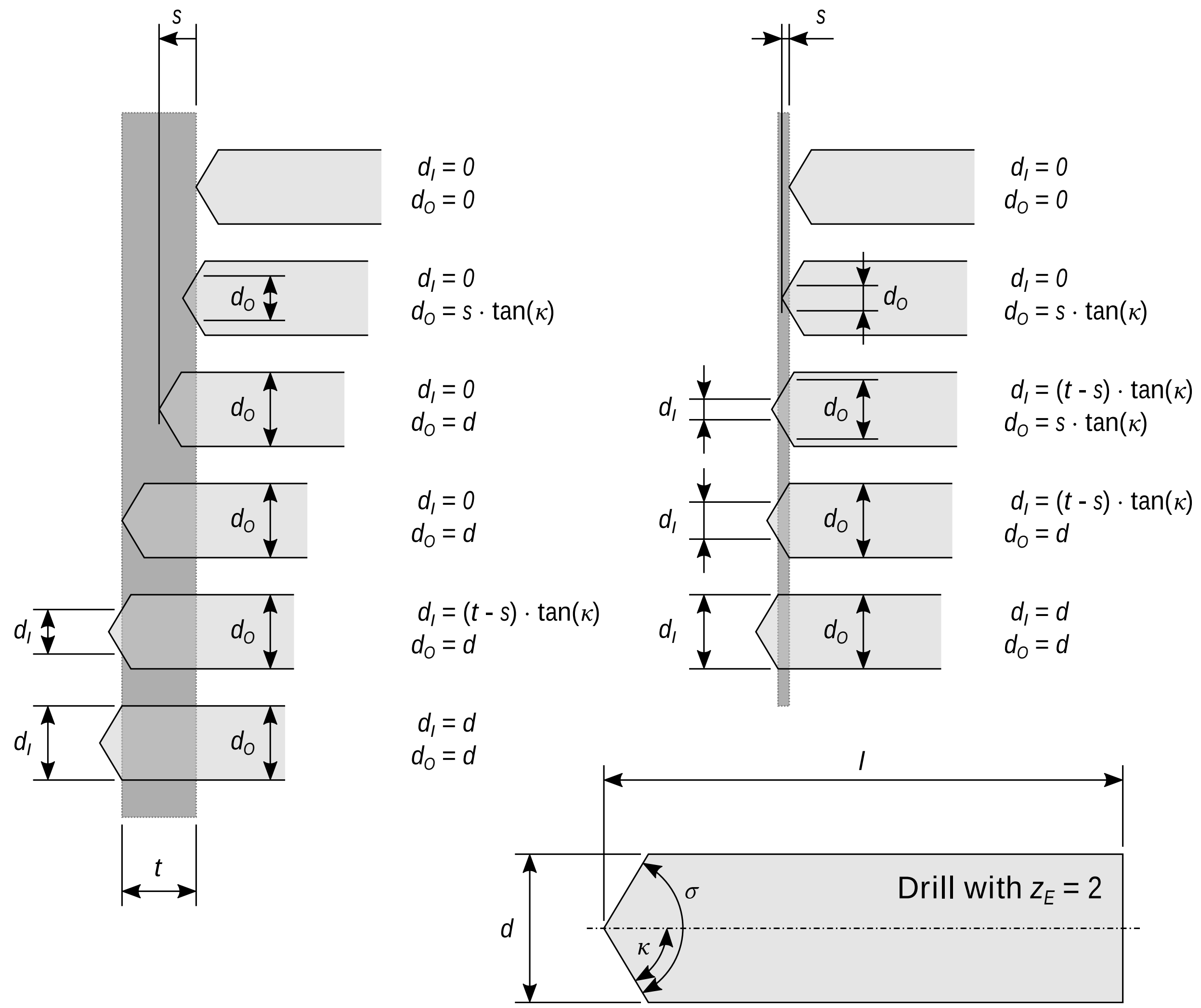


Figure 1: Drilling diameters for several states during the drilling process. A distinction is made as to whether the material is thicker than the height of the drill tip (left) or not (right). A drill is described by its length l, diameter d, and number of cutting edges z_E (shown in the bottom right), while the material is described by its thickness t.

The MPR algorithm is used in the developed model. It is connected to the multibody model in *Modelica* via an external library, similar to the method in Buse et al. (2023) [1].

Figure 1 shows the resulting drilling diameters for several states during the process. Collision detection is used to determine both the penetration depth of the drill and if the drill cone is penetrating the workpiece, is within the workpiece, or is leaving the workpiece. Depending on the state, different calculations are used to determine the diameters (see Figure 1). In addition, it is necessary to distinguish if the material is thicker than the height of the drill cone.

2.3 Kinematic model

The kinematic model uses the calculated drilling diameters from Section 2.2. Taking into account the drilling diameters, the cutting force per cutting edge F_{cz} is calculated with a modification of Equation (6) [2]:

$$F_{cz} = \frac{(d_O - d_I) \cdot f_z}{2} \cdot k_c \qquad (10)$$

Based on the cutting force, the feed force F_f is calculated using Equation (8). The kinematic part of the developed drilling-force model has already been used in the *MFlex 2025* project [10].

2.4 Dynamics model with applied forces

A different approach is used for the dynamics model. The drilling forces are not applied directly to the bodies in the multibody model. Instead, a compliant contact model is applied to the drill body. This allows the drill to be pressed against the table even when it is not rotating. This also increases the numerical stability of the model. The collision detection provides the penetration depth s for the contact model.

The idea now is to make the feed force for the drilling F_f equal to the contact normal force F_N (see Equation (9)). The cutting force per cutting edge F_{cz} can then be calculated from this:

$$\begin{aligned} F_f &= F_N && (11) \\ \Rightarrow z_E \cdot F_{cz} \cdot \sin(\sigma) &= k \cdot s + d \cdot \dot{s} && (12) \\ \Rightarrow F_{cz} &= \frac{k \cdot s + d \cdot \dot{s}}{z_E \cdot \sin(\sigma)} && (13) \end{aligned}$$

Next, the feed per cutting edge f_z is calculated from the cutting force per cutting edge. This is done by combining Equation (10) and Equation (5):

$$\begin{aligned} F_{cz} &= \frac{(d_O - d_I) \cdot f_z \cdot 0.001^z}{2 \cdot (f_z \cdot \sin(\frac{\sigma}{2}))^z} \cdot k_{c1.1} && (14) \\ \Rightarrow F_{cz} &= \frac{(d_O - d_I) \cdot f_z^{1-z} \cdot 0.001^z}{2 \cdot \sin(\frac{\sigma}{2})^z} \cdot k_{c1.1} && (15) \\ \Rightarrow f_z &= \left(\frac{F_{cz} \cdot 2 \cdot \sin(\frac{\sigma}{2})^z}{(d_O - d_I) \cdot 0.001^z \cdot k_{c1.1}}\right)^{\frac{1}{1-z}} && (16) \end{aligned}$$

The feed per rotation f is then calculated based on f_z and the number of cutting edges z_E:

$$f = z_E \cdot f_z \qquad (17)$$

This feed per rotation is now used to calculate the target velocity v_{tar} of the drill, based on the angular velocity ω of the drill, which is derived from the rotational frequency n of the drill:

$$v_{tar} = f \cdot n = f \cdot \frac{\omega}{2 \cdot \pi} \qquad (18)$$

Finally, the target drill depth s_{tar} can be calculated by integrating the target velocity v_{target}:

$$s_{tar} = \int v_{tar} \, \mathrm{d}t \qquad (19)$$

This target drill depth is used as the reference depth for the calculation of the contact force. Equation (9) is modified to calculate the contact normal force F_N:

$$\begin{aligned} F_N &= k \cdot \Delta s + d \cdot \Delta \dot{s} && (20) \\ \Rightarrow F_N &= k \cdot (s - s_{tar}) + d \cdot v && (21) \end{aligned}$$

The penetration depth s of the drill and the drill velocity v are still used, but now the contact surface of the workpiece moves based on the drilling process. Its position is defined by s_{tar}.

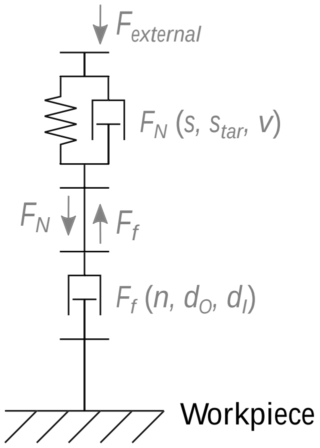


Figure 2: A substitute model of the drilling-force model. An external force is applied to the spring-damper element representing the compliant contact model. The contact force is equal to the feed force of the drilling-force model, here represented by a damper element, with the damping force dependent on the rotational frequency and the drill diameters.

A substitute model of the developed drilling-force model is shown in Figure 2. It can be seen as a series connection of a spring-damper element and a damper element.

The former describes the contact normal force and the latter the movement of the drill as a function of the rotational frequency and the drilling diameters.
Figure 3 shows a more detailed representation of the drilling-force model. It consists of three parts:

- A *DrillGeometry* model to calculate the drilling diameters and the penetration depth of the drill based on an external collision detection library (see Section 1.2 and Section 2.2).

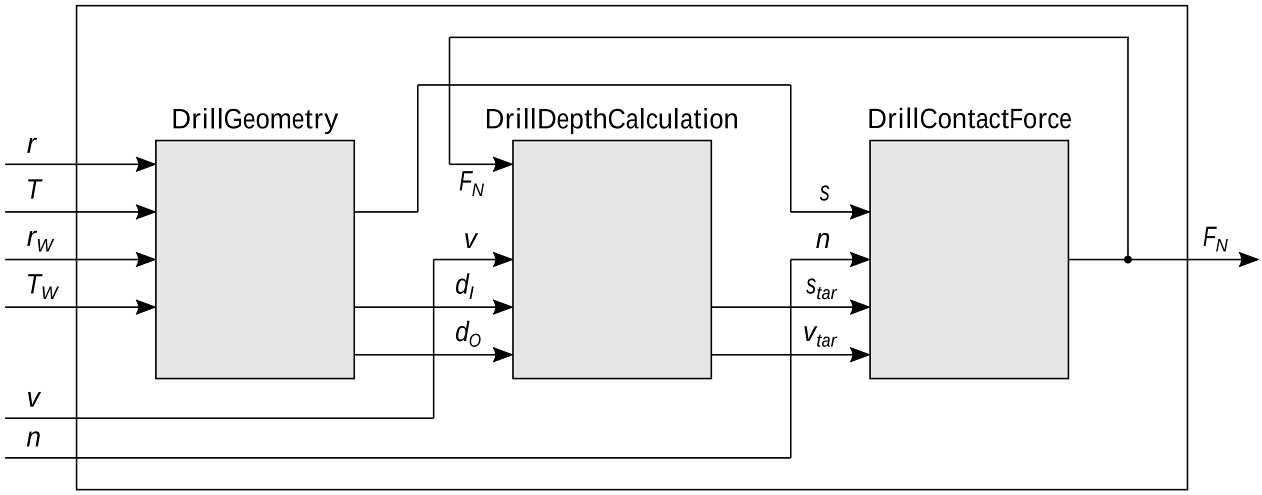


Figure 3: The developed drilling-force model with all models included. The drilling diameters and the drill penetration are calculated in the *DrillGeometry* model. The *DrillDepthCalculation* model provides the target position for the drill. The contact force is calculated in the *DrillContactForce* model based on the actual and target position of the drill.

- A *DrillDepthCalculation* model to calculate the target position of the drill. The calculation is based on Equations (11) to (19).
- A *DrillContactForce* model to calculate the compliant contact force based on the actual and target position and the velocity of the drill (see Equation (20) and (21)).

As mentioned above, the inputs to the drilling-force model are the velocity v and the rotational frequency n of the drill.

In addition, the position r and orientation T of the drill are required for the collision detection in the *DrillGeometry* model.

The same applies to the workpiece, described by r_W and T_W. Collision detection also requires the geometry of the drill and the workpiece.

3 Applications

In this section, two drilling processes are shown. One of them consists of drilling in a thin plate.

The kinematic model is used to calculate the forces involved. In the other process, the dynamics model is applied for a drilling process. This involves drilling into a material with a high material thickness.

3.1 Kinematic drilling into a thin plate

In the first example, the kinematic model is used for the drilling process of a thin plate. The plate is thinner than the height of the drill cone.

The material *34 CrMo4* is used for the workpiece (see [2, p. 19]):

$$k_{c1.1} = 2240\,\frac{\mathrm{N}}{\mathrm{mm}^2} \qquad (22)$$

$$z = 0.21 \qquad (23)$$

A rotational frequency of $n = 9.28\,\frac{1}{\mathrm{s}}$ and a velocity of $v = 1.672\,\frac{\mathrm{mm}}{\mathrm{s}}$ are used. The drill diameter is $d = 16$ mm with an angle of 118 degrees.

From this, the feed per cutting edge per rotation f_z can be calculated, resulting in a specific cutting force k_c of:

$$f_z = \frac{f}{z_E} = \frac{v}{z_E \cdot n} = \frac{1.672\,\frac{\mathrm{mm}}{\mathrm{s}}}{2 \cdot 9.28\,\frac{1}{\mathrm{s}}} \approx 0.090\ \mathrm{mm} \qquad (24)$$

$$\Rightarrow k_c = \frac{0.001^z}{(f_z \cdot \sin(\frac{\sigma}{2}))^z} \cdot k_{c1.1} \approx 3836\,\frac{\mathrm{N}}{\mathrm{mm}^2} \qquad (25)$$

The calculated maximum cutting force is:

$$F_{c,\,max} = z_E \cdot \frac{d \cdot f_z}{2} \cdot k_c \approx 5524\ \mathrm{N} \qquad (26)$$

However, this value is not reached because the difference between the inner and outer drilling diameter is much smaller. The results are presented in Figure 4. The cutting force, the feed force, and both drilling diameters are shown. The maximum cutting force is $F_c = 986$ N and the maximum feed force is $F_f = 2917$ N (see Figure 4).

The simulation was run on a desktop computer with an Intel Core i7-11700K processor. A *Rkfix2* fixed-step solver with a fixed step size of 0.001 seconds was used. The computation time was 0.11 seconds for a simulation time of 10 seconds (real-time factor of 0.011). The simulation is real-time capable.

3.2 Drilling process using the dynamics model

The second example shows the dynamics model.

A *Modelica* model has been created for a drilling process. As in Section 3.1, the material used is *34 CrMo4*. The model is shown in Figure 5. An external force and angular velocity are generated with source blocks from the MSL. The drilling-force model is integrated into one block. This block is connected to a revolute joint, which in turn is connected to a prismatic joint. This allows a linear motion and a rotation. The drilling forces are applied directly to the frame of the model.

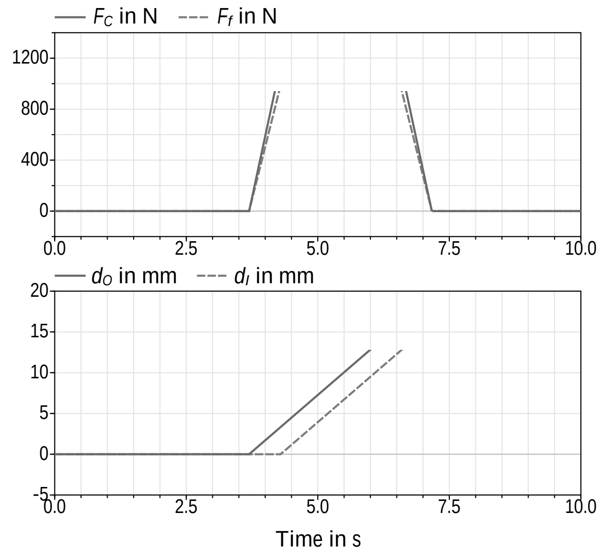


Figure 4: Results for the drilling process of a thin plate. The cutting force and the feed force are shown above. The drilling diameters are shown at the bottom.

The workpiece in the form of a plate is represented by a second block. Both the *DrillGeometry* model of the drilling-force model and this workpiece are modeled with components from a contact detection library. This allows multiple workpieces to be drilled in a multibody scenario without the need for predefined contact pairs.

The results are shown in Figure 6. The variable-step solver *Dassl* was used with a tolerance of 1e-8. It took 0.33 seconds to run the 40 seconds simulation.

As in Section 3.1, a desktop computer with an Intel Core i7-11700K processor was used. Fixed-step solvers are not suitable because the contact between the drill and the workpiece is very stiff and therefore very small step sizes are required at some points.

The actual and target position of the drill fit together well. The feed force and cutting force results match the manually calculated values. The specific cutting force is not defined for the full range of feeds (see Equation (5)). It is therefore limited.

When the feed is less than 1 mm, the specific cutting force does not increase any further. This increases the stability of the model.

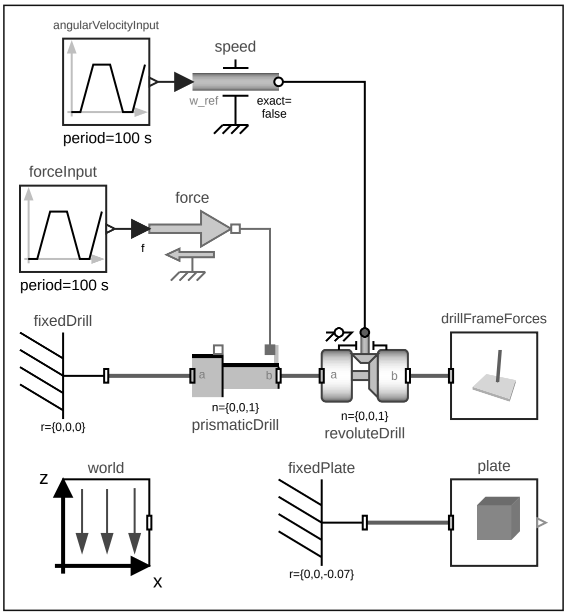


Figure 5: The *Modelica* model of the drilling process based on the dynamics model. The drilling model applies the forces directly to the frame. An external force and angular velocity are also applied to the frame.

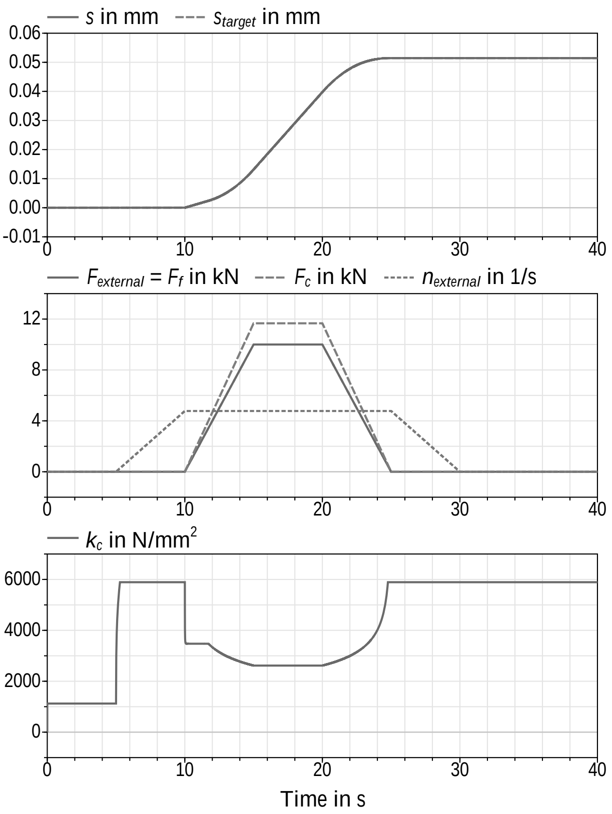


Figure 6: The results for the drilling process based on the dynamics model. The image shows the actual and target position for the drill (top), the external force, the cutting force and the rotational frequency (middle), and the specific cutting force (bottom).

4 Conclusion

In this work, a drilling-force model has been developed. It consists of two parts. A kinematic model can be used to calculate the forces of the drilling process without applying them. The drilling forces in the dynamics model are applied to the bodies in the multibody environment and can be used for body interaction.

The drilling-force model is based on drilling-force calculations, collision detection, a compliant contact model, and a method for applying the forces to the multibody environment. It can be used in system simulation environments such as *Modelica*. The capability of the developed model has been successfully demonstrated in two example drilling processes. However, the model has some limitations: Holes must be drilled at a 90 degree angle to the workpiece without tangential motion.

And friction forces are not taken into account (for both the contact force and the drilling forces).

Possible further developments include the integration of a friction model. This would allow forces in the tangential direction to be considered for the contact model. Friction forces could also be included in the drilling-force calculation. Torques from the drilling process could also be considered, for example to model the drive of the drill spindle. In addition, the combination of the drilling-force model with force controlled robot models could be investigated.

Acknowledgement

I would like to thank my colleagues Fabian Buse, Bernhard Thiele, Tobias Treichl, and Dirk Zimmer for fruitful discussions regarding the drilling-force model. Additionally, I would like to thank Anja Reiser for her linguistic advice.

Parts of this model have been developed in the *MFlex 2025* project funded by the German Federal Ministry for Economic Affairs and Climate Action (Förderkennzeichen 20X1720B).

Publication Remark

This contribution is the revised version of the workshop version published in
ASIM Workshop GMMS/STS 2025 Tagungsband
ARGESIM Report AR 48, p 87-93
ISBN ebook: 978-3-903347-66-3
volume DOI 10.11128/arep.48

References

[1] Buse, Fabian, Antoine Pignède, and Stefan Barthelmes (2023). “A Modelica Library to Add Contact Dynamics and Terramechanics to Multi-Body Mechanics”. In: *Proceedings of the 15th International Modelica Conference*, pp. 433–442. DOI: 10.3384/ecp204433.

[2] Dietrich, Jochen (2016). *Praxis der Zerspantechnik. Verfahren, Werkzeuge, Berechnung.* 12th ed. Wiesbaden: Springer Fachmedien Wiesbaden. ISBN: 978-3-658-14052-6.

[3] Fischer, Ulrich et al. (2008). *Tabellenbuch Metall.* 44th ed. Haan-Gruiten: Verlag Europa-Lehrmittel. ISBN: 978-3-8085-1724-6.

[4] Flores, Paulo (2022). “Contact mechanics for dynamical systems: a comprehensive review”. In: *Multibody System Dynamics* 54.2, pp. 127–177. DOI: 10.1007/s11044-021-09803-y.

[5] Fritz, A. Herbert and Günter Schulze (2015). *Fertigungstechnik.* 11th ed. Berlin and Heidelberg: Springer Vieweg. ISBN: 978-3-662-46554-7.

[6] Gilbert, Elmer G., Daniel W. Johnson, and S. Sathiya Keerthi (1988). "A fast procedure for computing the distance between complex objects in three-dimensional space". In: *IEEE Journal on Robotics and Automation 4.2*, pp. 193–203. DOI: 10.1109/56.2083.

[7] Modelica Association (2017). *Modelica - A Unified Object-Oriented Language for Systems Modeling. Language Specification Version 3.4. Tech. Rep.* Linköping: Modelica Association. URL: https://modelica.org/documents/ModelicaSpec34.pdf

[8] Modelica Association (2020). *Modelica Standard Library. Version 4.0.0.* Modelica Association. URL: https://github.com/modelica/ModelicaStandardLibrary

[9] Otter, Martin, Hilding Elmqvist, and Sven Erik Mattsson (2003). "The New Modelica MultiBody Library". In: *Proceedings of the 3th International Modelica Conference*, pp. 311–330.

[10] Reiser, Robert et al. (2022). "Real-time simulation and virtual commissioning of a modular robot system with OPC UA". In: *ISR Europe 2022. 54th International Symposium on Robotics (Munich)*. Munich: VDE Verlag. ISBN: 978-3-8007-5891-3.

[11] Reiser, Robert and Matthias J. Reiner (2023). "Modeling and simulation of dynamically constrained objects for limited structurally variable systems in Modelica". In: *Proceedings of the 15th Int. Modelica Conference*, pp. 151–158. DOI: 10.3384/ecp204151.

[12] Snethen, Gary (2008). "Xenocollide: Complex collision made simple". In: *Game programming gems 7*. Ed. by Scott Jacobs. Boston, MA: Charles River Media, pp. 165–178. ISBN: 978-1-58450-527-3.

System Identification of an Omnidirectional Test Vehicle for Model-Based Function Design

Björn Carstens*, Marian Göllner, Taihao Li, Xiaobo Liu-Henke

Control Engineering and Vehicle Mechatronics Group, Ostfalia University of Applied Sciences,
Salzdahlumer Str. 46/48, 38302 Wolfenbüttel, Germany; *bj.carstens@ostfalia.de

SNE 35(4), 2025, 187-193, DOI: 10.11128/sne.35.tn.10753
Selected ASIM SST 2024 Postconf. Publication: 2024-12-10
Ext. English version: 2025-08-01; Accepted: 2025-08-15 SNE - Simulation Notes Europe, ARGESIM Publisher Vienna ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. System identification is an essential method in control engineering that enables the creation and validation of mathematical models of dynamic systems. This paper provides a detailed overview of the principles and methods of system identification, with a particular focus on validating the transfer function of an omnidirectional test vehicle with a Killough chassis. Of particular interest is the identification of the coupling structures of the overactuated system. The identified model forms the basis for model-based function design.

Introduction

The multidisciplinary field of system identification, which deals with the development of mathematical models for dynamic systems based on experimentally obtained data, plays a central role in engineering science. This process is used in many application areas, such as control engineering, signal processing, and mechanical engineering [1]. The ability to develop precise models enables engineers and scientists to understand, analyze, and control the behavior of complex systems.

The fundamental idea of system identification is to find a model that describes the behavior of a system based on input and output data. A dynamic system can be represented by various mathematical models, including differential equations, transfer functions, and state-space representations.

These mathematical models enable the description of the temporal or frequency-dependent behavior of a system and the making of predictions about its future states.

The acquisition of relevant data is the first and one of the most important steps in the system identification process. Data can be collected through experimental procedures, which include targeted experiments to generate meaningful datasets, or through passive observation, where existing data from the operation of the system is utilized. The aim of this work is to present the fundamentals of system identification and to develop a methodology for validating models based on experimental data.

Here, a methodology presented below is used and successively applied to a test vehicle described in Chapter 4.1 in order to perform the verification and validation of the results on a real system in combination with an iteratively built model.

1 Methodology

The consistently model-based and verification-oriented function design and verification of networked mechatronic systems according to [2] has proven to be time and cost-efficient in numerous applications in research and industry. Figure 1 shows the model-based mechatronic development cycle, which includes Model-in-the-Loop (MiL), Software-in-the-Loop (SiL), and Hardware-in-the-Loop (HiL) simulations as well as real-time realization through prototypes for verification.

The design process begins with modeling based on the real system, which is reduced or simplified according to requirements, initially resulting in a physical model. This is converted into a mathematical model using physical laws, which in turn can be represented in the computer, for example in the form of signal flow diagrams, and simulated using CAE tools and appropriate numerics. The modeling process includes measurements on the real system. The parameters of the mathematical model are identified and the simulation is validated.

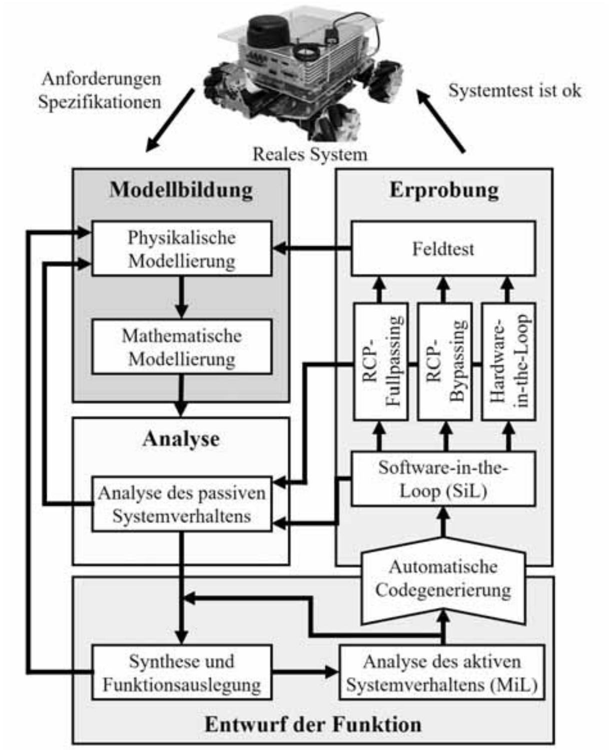

Anforderungen
Spezifikationen
Systemtest ist ok
Reales System
Modellbildung
Physikalische Modellierung
Mathematische Modellierung
Analyse
Analyse des passiven Systemverhaltens
Erprobung
Feldtest
RCP-Fullpassing
RCP-Bypassing
Hardware-in-the-Loop
Software-in-the-Loop (SiL)
Automatische Codegenerierung
Synthese und Funktionsauslegung
Analyse des aktiven Systemverhaltens (MiL)
Entwurf der Funktion

Figure 1: Mechatronic development cycle.

A subsequent analysis of the simulation results allows conclusions about the fundamental static and dynamic properties of the real system, on the basis of which the conception of functions takes place.

Both the function architecture and approaches for its design and optimization are determined. Complex functions are also broken down into hierarchical subfunctions according to the generalized cascade principle to reduce function complexity and thus make the design process manageable.

For this purpose, hardware and software requirements are considered at an early stage of development and interfaces for cross-functional communication are defined.

The testing process is carried out in parallel with development. When a subfunction has been developed, it is tested and analyzed.

In this work, requirements are first d etermined using this method. Through the analysis of the physical model as well as the mathematics, the function is modeled. Each part of the function is tested accordingly.

2 State of Knowledge

System identification refers to the creation of a model for a dynamic system that describes the relationship between input and output data. This paper covers the theoretical fundamentals of system identification, with particular focus on identification using linear models. Methods for determining non-parametric models are presented, such as the determination of transfer functions from step responses and the determination of the frequency response. For parametric models, parameter estimation is an essential component of system identification.

The theoretical fundamentals of parameter estimation are explained in detail in [3, 4]. Excitation signals play a central role here, as they reveal the behavior of the system to be controlled. In control engineering, it is particularly important that the excitation signal covers the relevant frequency spectrum in order to develop accurate models and ensure effective control. Various types of excitation signals, such as impulses, steps, sinusoidal oscillations, and pseudo-random binary signals (PRBS), are used to comprehensively characterize system behavior. Precise model building based on these signals enables engineers to optimize the control behavior of systems and improve the stability and performance of control loops. [4]

Frequency response analysis is a central method for the identification and validation of models in control engineering. It includes the analysis of the system response to sinusoidal input signals at different frequencies. This method describes how the system responds to different harmonic input variables, whereby amplitude and phase shifts are measured and analyzed. A Bode diagram graphically represents the results of frequency response analysis. It displays the logarithmic amplitude and phase shift as functions of frequency. This not only allows the stability and dynamics of a system to be assessed, but also weaknesses in the model to be identified. The practical steps of a frequency response analysis include:

1. Applying input signals to the system.
2. Measuring the respective output signals.
3. Calculating amplitude ratios and phase shifts between input and output signals.
4. Presenting the results in Bode diagrams.
5. Comparing experimental frequency responses with theoretical models.

Through the systematic comparison of experimental and theoretical data sets, the accuracy of the models can be validated and possible modeling errors can be identified. [5, 6, 7]

3 Conception

The process of system identification begins with modeling, in which the behavior of the system is described by physical laws and equations [8]. Subsequently, input and output signals of the system are measured, which can be generated by suitable excitation signals, see Chapter 2. Signals with a broad frequency spectrum are particularly suitable for vehicle systems [9]. Based on the measured data, a mathematical model is chosen, often in the form of AR (Autoregressive), MA (Moving Average), or ARMA (Autoregressive Moving Average) models.

The parameters of the model are optimized using methods such as the least squares method, the recursive least squares method, or the prediction error method. Validation is performed by comparing model predictions with independent data sets to ensure that the model adequately represents system behavior.

Challenges such as measurement errors, nonlinearities, and temporal variations of the system require a balanced relationship between model complexity and manageability. Experimental methods are based on measuring system responses to defined test signals, while theoretical approaches are based on modeling using physical laws.

An important step in system identification is frequency response measurement, which allows precise validation of the transfer function.

In our case, the transfer function $G(s)$, which describes the ratio of the Laplace transforms of the output to input signals, was established as follows:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_0 + b_1 s + \cdots + b_m s^m}{a_0 + a_1 s + \cdots + a_n s^n}$$

To verify the transfer function, the system was exposed to harmonic input signals of different frequencies, and the respective output signals were measured. The frequency response measurement was performed with an Abacus 901, which precisely analyzes amplitude and phase changes.

Validation consists of comparing the experimental frequency response with the model prediction. If the measured behavior matches the modeled behavior, the model is considered confirmed. Deviations may indicate nonlinear effects, measurement errors, or insufficient model assumptions.

3.1 Kinematics

The Mecanum wheel shown in Figure 2 displays the quantities necessary for describing the kinematics. The angular velocity ω is shown, which is often also represented by the rotational speed n in the further course. In addition, the speed of the wheel in the locking direction is at the inclination angle α of the rolling element. The vehicle velocity V can be calculated via the wheel radius r and the angular velocity ω.

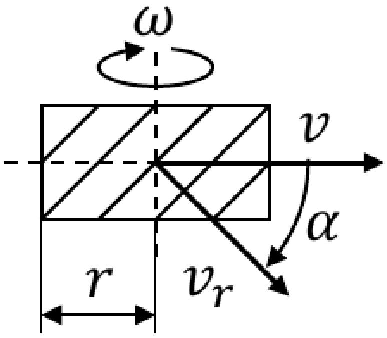


Figure 2: Kinematics of a Mecanum wheel [10].

The goal is to establish the kinematics matrix shown in Equation (1). This describes the wheel speeds n as a function of the total velocity of the test vehicle in its three degrees of freedom v_{fx}, v_{fy}, and $\dot{\omega}_f$.

$$\begin{bmatrix} n_1 \\ n_2 \\ n_3 \\ n_4 \end{bmatrix} = \underline{\underline{K}} \cdot \begin{bmatrix} v_{fx} \\ v_{fy} \\ \dot{\omega}_f \end{bmatrix} \tag{1}$$

3.2 Decoupling

The decoupling of mechatronic systems is a central concept in control engineering, which aims to minimize or eliminate the interactions between different components of a system.

This enables independent control of the individual components and improves the overall performance and efficiency of the system.

In mechatronic systems, which often consist of mechanical, electrical, and software-controlled components, decoupling can be particularly challenging since these components are closely interconnected. [11]

The decoupling is performed using the kinematics matrix $\underline{\underline{K}}$. For a vehicle with four Mecanum wheels, the result according to [12] is:

$$\underline{\underline{K}} = \begin{bmatrix} \frac{\cos(\delta_1)}{\sin(\alpha_1)} & \frac{-\sin(\delta_1)}{\sin(\alpha_1)} & \frac{-\sin(\varepsilon_1)}{\sin(\alpha_1)} \cdot r_1 \\ \frac{\cos(\delta_2)}{\sin(\alpha_2)} & \frac{-\sin(\delta_2)}{\sin(\alpha_2)} & \frac{-\sin(\varepsilon_2)}{\sin(\alpha_2)} \cdot r_2 \\ \frac{\cos(\delta_3)}{\sin(\alpha_3)} & \frac{-\sin(\delta_3)}{\sin(\alpha_3)} & \frac{-\sin(\varepsilon_3)}{\sin(\alpha_3)} \cdot r_3 \\ \frac{\cos(\delta_4)}{\sin(\alpha_4)} & \frac{-\sin(\delta_4)}{\sin(\alpha_4)} & \frac{-\sin(\varepsilon_4)}{\sin(\alpha_4)} \cdot r_4 \end{bmatrix} \tag{2}$$

With the barrel angle of the Mecanum wheel α_i, the position angle of the wheel axis β_i to the vehicle's coordinate system, and the orientation angle γ_i.

Together with the absolute distance of the origin coordinate system of the Mecanum wheel to the origin coordinate system of the vehicle r_i, the polar coordinates result starting from the coordinate origin.

$$\alpha_i = (-1)^i \cdot 45^\circ \tag{3}$$

$$\beta_i = \arcsin\left(\frac{b_i}{r_i}\right) - (i-1) \cdot 90^\circ \tag{4}$$

$$\gamma_i = \arcsin\left(\frac{b_i}{r_i}\right) + (i-1) \cdot 90^\circ \tag{5}$$

$$r_1 = \sqrt{{b_i^2}^2 + {b_n}^2}, r_{i>1} = \sqrt{b_{i-1}^2 + b_i^2} \tag{6}$$

The construction angles $\delta_i = \alpha_i - \beta_i - \gamma_i$ and $\varepsilon_i = \alpha_i - \beta_i$ result from these. For the test vehicle used, the kinematics matrix $\underline{\underline{K}}$ is thus:

$$\underline{\underline{K}} = \begin{bmatrix} 1 & -1 & -\frac{l_x+l_y}{2} \\ 1 & 1 & -\frac{l_x+l_y}{2} \\ 1 & -1 & \frac{l_x+l_y}{2} \\ 1 & 1 & \frac{l_x+l_y}{2} \end{bmatrix} \cdot \frac{1}{2 \cdot \pi \cdot r} \tag{7}$$

4 Test Setup Infrastructure

The identification of a simulation model requires, as already mentioned, additional hardware and software as well as corresponding adaptations in the test vehicle. All necessary interventions are presented below. The transfer function of the test vehicle's velocity is to be determined.

4.1 Test Vehicle Setup

The test vehicle offers numerous functions, whereby only the systems relevant to this work are described here. Further details can be found in [12].

One relevant system is the microcontroller. The test vehicle used is equipped with an STM32H743 board, which has a dual-core architecture. The microcontroller can receive instructions and transmit status information via a LAN interface. The drive system of the test vehicle can consist of swerve, holonomic, or Mecanum systems.

Mecanum wheels, developed by Bengt Ilon, enable omnidirectional movement through special rollers on the circumference of the wheel, which allow independent control of the direction of movement in all axes. Swerve wheels combine rotational movements and lateral displacements, allowing the vehicle to also be steered in all directions, while holonomic wheels offer similar freedom of movement, but in a completely different way. Mecanum, swerve, and holonomic wheels offer advantages such as space savings, lower construction and maintenance costs, high precision, and easy replacement of rollers.

Mecanum wheels consist of obliquely mounted rollers that allow the vehicle to move without directional restrictions. Swerve wheels are special steering and chassis combinations that enable complete freedom of movement in all directions by independently controlling the steering and speed of each wheel. Holonomic wheels refer to systems that enable a vehicle to move in all directions without restrictions on the direction of movement. They are ideal for intralogistics, automated guided vehicle systems, and mobile robotics [13]. The test vehicle is scaled down; the scaling law will be taken into account during evaluation and adaptation.

In a next step, a distance controller is to be implemented on the test vehicle's Raspberry Pi. This controller continuously records the distance to a preceding object and calculates an appropriate target velocity based on this.

The calculated velocity values are then transmitted to the vehicle's microcontroller, which takes over the control of the drive unit.

4.2 Measurement Test System

Figure 3 shows the measurement test system. A host PC with DP900 software from Data Physics is connected to the Abacus 901 via an Ethernet connection. The excitation signal as output, as well as the two inputs, are connected to the test vehicle.

The Abacus 901 generates analog excitation signals, which are converted into digital signals by the AD converters of the dSPACE real-time hardware. This digital excitation signal is immediately fed back through a DA converter to the Abacus. This signal later serves as a reference signal.

This makes it possible to compensate for the possible interfering influences resulting from the AD and DA conversion. The chosen signal path was deliberately designed so that the dead times that occur due to signal and information conversions can be recorded.

The digital excitation signal is received via the Ethernet interface of the microcontroller as a target velocity signal [12].

The microcontroller now regulates the target velocity using a velocity controller implemented on the microcontroller. The drive motors are equipped with rotary encoders.

With the help of these encoders, the microcontroller can determine the current wheel speed and, using the inverse kinematics from Eq. (7), the velocity. The velocity is then sent back via the Ethernet interface of the microcontroller to the dSPACE real-time hardware, which transmits this velocity as an analog signal to the Abacus 901.

This signal path is to be identified, since later the Raspberry Pi will also send the target velocity to the microcontroller via the Ethernet interface.

The setup shown in Figure 3 represents the transmission of measurements in the software. When forming the transfer function, the excitation signal of the Abacus is not used as the system input, but rather the signal influenced by the real-time hardware.

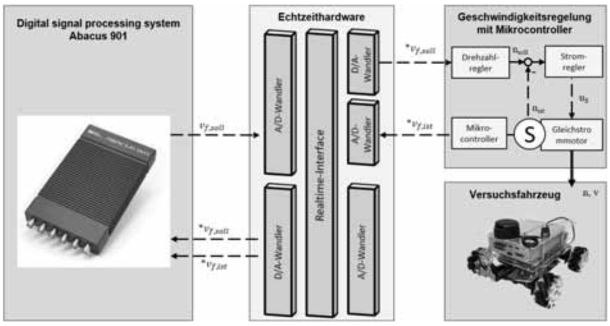


Figure 3: Transmission of excitation in the software.

5 Results

The results of the frequency response measurements were analyzed and compared with the mathematical model. Through parameter adaptation, the parameters of the model were modified and optimized with respect to the deviation from the measurements.

Thus, the transfer matrix $\underline{\underline{G}}$ of the system could be found.

$$\underline{\underline{G}} = \begin{bmatrix} G_{v;x,x} & G_{v;x,y} & G_{v;x,\psi} \\ G_{v;y,x} & G_{v;v;y,y} & G_{v;y,\psi} \\ G_{v;\psi,x} & G_{v;\psi,y} & G_{v;\psi,\psi,} \end{bmatrix} \tag{8}$$

The main diagonal describes the direct effect of the excitation signal set for the respective degree of freedom via the kinematics. The transfer functions of the main diagonal were excited with the help of the transformation matrix shown in Eq. (7) via the velocity controller and related to the degree of freedom to be investigated by direct backward transformation of the measurement signals.

The Bode diagram shown in Figure 4 represents the transfer function of velocity in the x-direction with excitation in the x-direction. The diagram shows the characteristic of a PT1 system.

The transfer function $G_{v;x,x}$ thus results as a first-order time delay element with a time constant of $\omega = 1.93 \cdot 10^1 \rightarrow T = \frac{1}{19.3} \approx 0.05$ and a gain factor $K = 1$.

The off-diagonal elements of the transfer matrix represent the coupling of the system and would be disruptive to the desired system dynamics with a relatively high gain factor.

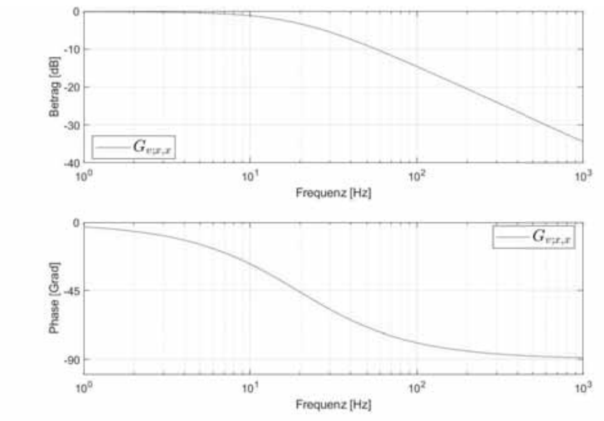


Figure 4: Transfer function of velocity from MATLAB/Simulink.

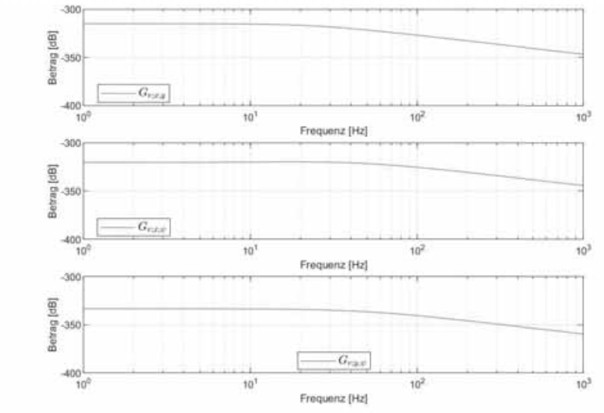


Figure 5: Transfer functions of some off-diagonal elements of velocity from MATLAB/Simulink.

The Bode diagrams shown in Figure 5 represent the transfer functions of the off-diagonal elements. The upper Bode diagram is the transfer function for velocity in the y-direction with excitation in the x-direction. The middle diagram shows excitation in the x-direction and the system response in ψ, the vertical axis of the vehicle. The last diagram shows excitation in the y-direction and again the system response in ψ around the vertical axis of the vehicle.

What all responses have in common is that the gain factor of the identified PT1 systems is at -320 dB $= 20\log_{10}(K) \rightarrow K \approx 10^{-16}$ and thus the influence of the coupling in relation to the desired system dynamics of the main diagonal is very low.

Based on this modified model, the controller synthesis and optimization can now be carried out using the mechatronic development process.

For this purpose, a reference model of the same order and structure with free parameters is built. The identification of the optimal parameters is performed by minimizing the squared error between an assumed transfer element of the reference model as the feedforward transfer function of the separated transfer path within the transfer matrix and the measured frequency response.

The entirety of the transfer elements is thus optimally fitted to the measured frequency response in their course.

6 Summary and Outlook

To derive a model of the dynamic behavior of an omnidirectional intralogistics vehicle on a reduced scale, the presented methods of system identification were applied. The omnidirectionality of the vehicle is realized here through the use of a Killough chassis with four Mecanum wheels.

In the field of kinematics, the Mecanum wheel describes the movement of a vehicle based on angular velocity and vehicle velocity. Decoupling is a central concept for minimizing interactions between system components.

The test vehicle is also equipped with a microcontroller which is responsible for the local control of the chassis. The hardware additionally required for measurement includes a connection to the host PC and the Abacus 901 signal analyzer; necessary additions in the vehicle's software include the processing and retransmission of setpoint and actual values to these. The results include the representation of transfer functions for various degrees of freedom and show the system responses to different excitations.

In the further course, the same method for identifying the transfer function will be carried out on a larger test vehicle. The test platform AURONA serves as a platform for current research projects, therefore it is necessary to identify basic driving dynamics models for this vehicle and, based on this, to further develop various autonomous driving functions that were first implemented with the help of the small test vehicles.

Acknowledgement

Funded by the Lower Saxony Ministry of Science and Culture under funding number ZN4068 in the Niedersächsisches Vorab of the VolkswagenStiftung.

VolkswagenStiftung Niedersächsisches Ministerium für Wissenschaft und Kultur

Publication Remark

This contribution is an extended English version of the workshop publication published in
ASIM Conference SST 2024 Langbeiträge
ARGESIM Report AR 47, p. 175-83
DOI DOI 10.11128/arep.47.a4723
ISBN ebook: 978-3-903347-65-6
Volume DOI 10.11128/arep.47

References

[1] Balakrishnan V. System identification: theory for the user (second edition). *Automatica*. 2002;38(2):375–378.

[2] Liu-Henke X. *Mechatronische Entwicklung der aktiven Feder-/Neigetechnik für das Schienenfahrzeug RailCab: Zugl.: Paderborn, Univ., Diss., 2005*, vol. 589 of *Fortschritt-Berichte VDI Reihe 12, Verkehrstechnik/Fahrzeugtechnik*. Düsseldorf: VDI-Verl., als ms. gedr ed. 2005.

[3] Bohn C. Systemidentifikation. In: *HÜTTE – Das Ingenieurwissen*, edited by Hennecke M, Skrotzki B, pp. 1–54. Berlin, Heidelberg: Springer Berlin Heidelberg. 2020;.

[4] Unbehauen H, ed. *Identifikation, Adaption, Optimierung: Mit 11 Tabellen*, vol. 3 of *Studium Technik*. Braunschweig: Vieweg, 6th ed. 2000.

[5] BLASS tH. Anwendung der Frequenzganganalyse beim praktischen Betrieb von Regelungseinrichtungen. *at - Automatisierungstechnik*. 1954;2(1-12):137–143.

[6] Pestel E, Kollmann E. Die Frequenzgangmethode. In: *Grundlagen der Regelungstechnik*, edited by Pestel E, Kollmann E, Regelungstechnik in Einzeldarstellungen, pp. 148–246. Wiesbaden and s.l.: Vieweg+Teubner Verlag. 1961;.

[7] Iyer SV. Frequenzganganalyse. In: *Digitales Filterdesign mit Python für Anwendungen in der Energietechnik*, edited by Iyer SV, pp. 125–161. Cham: Springer Vieweg. 2024;.

[8] Isermann R. *Identifikation dynamischer Systeme 1*. Berlin, Heidelberg: Springer Berlin Heidelberg. 1992.

[9] Isermann R. *Automotive Control*. Berlin, Heidelberg: Springer Berlin Heidelberg. 2022.

[10] Dr-Ing X Liu-Henke, Sören Scherler, Marian Göllner, Johannes Maisik, Matthias Fritsch. Simulationsgestützte Konzeption der Antriebstopologie eines fahrerlosen Transportfahrzeugs. In: *Tagungsband ASIM Workshop STS/GMMS 2018*. ARGESIM. 2018; .

[11] Labisch D, Konigorski U. Verkopplungsbasierte Methode zur Entkopplung nichtlinearer Deskriptorsysteme. *at - Automatisierungstechnik*. 2014; 62(7):475–486.

[12] Jacobitz S, Göllner M, Zhang J, Liu-Henke X. Modellbasierte Entwicklung des Antriebsmoduls für vernetzte fahrerlose Transportfahrzeuge in einem cyber-physischen Labortestfeld;.

[13] Aktuelle Meldungen der Nabtesco Precision Europe GmbH.
URL `https://j2n.eu/r/wfeEa`

[14] Max van Haren, Lennart Blanken, Tom Oomen. Frequency Domain Identification of Multirate Systems: A Lifted Local Polynomial Modeling Approach. *2022 IEEE 61st Conference on Decision and Control (CDC)*. 2022;pp. 2795–2800.

[15] Tang W, Shi Z. Robust system identification of continuous-time model from frequency response function data. In: *2011 9th World Congress on Intelligent Control and Automation*. IEEE. 2011; pp. 661–665.

[16] Göllner M, Jacobitz S, Ferrara R, Liu-Henke X. Identifikation instabiler, unteraktuierter Systeme mit nicht-linearem dynamischen Verhalten. In: *ASIM SST 2024 Tagungsband Langbeiträge*, edited by Rose O, Uhlig T. ARGESIM Publisher Vienna. 4.– 6. September 2024; pp. 175–183.

Development of an Evaluation Method for Simulation Results in the Context of Industrial Symbiosis: A Target System for Increasing Resource Efficiency and Resilience

Björn Vollack*, Manuel Boelcke, Thorsten Schmidt

Chair of Logistics Engineering, TU Dresden, Münchner Platz 3, 01187 Dresden; *bjoern.vollack@tu-dresden.de

SNE 35(4), 2025, 195-198, DOI: 10.11128/sne.35.sn.10754
Selected ASIM SST 2024 Postconf. Publication: 2024-12-10
Rec. Impr. English Version: 2025-11-04; Accepted: 2025-11-15
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna
ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. The importance of energy and resource efficiency has increased significantly in recent years. Industrial symbiosis (IS) increases resource efficiency, among other things, with the help of energy and material exchange relationships between companies. In addition to efficiency, resilience is also becoming increasingly important. IS can contribute to avoiding production interruptions through a broad network of alternative relationships. The exchange relationships in an industrial cluster can be simulated. However, a standardised method for evaluating the simulation results is necessary in order to be able to compare different variants with each other. This article presents a target system that considers economic, ecological, symbiotic and resilient aspects. Different weightings of the target functions make it possible to set individual priorities. The method is demonstrated using an example with four companies and different simulation model variants.

Introduction

Recent years have impressively highlighted the importance of energy and resource efficiency. Industrial parks in particular can leverage potential through the application of industrial symbiosis (IS). IS describes a concept that increases resource efficiency through the exchange of water, energy, waste and by-products between companies, among other things [1], [2].

In addition to efficiency, resilience, i.e. the ability to respond to resource shortages, for example, is playing an increasingly important role for companies [3], [4]. In this respect, too, IS offers opportunities, on the one hand because not all material and energy requirements have to be covered externally – keyword: self-sufficiency – and on the other hand because, where applicable, the networking of companies provides alternatives to previous sources of supply or customers.

Currently, the design and operation of industrial parks are often not geared towards these internal exchange processes. The research project "Energy-efficient industrial cluster optimisation" (EnICO) aims to optimise energy and material exchange relationships in an industrial cluster using simulation. Simulation is used to determine overall optimal interaction relationships, taking into account various target functions.

This is based on a standardised method for evaluating the simulation results, which also takes into account different weighting preferences with regard to the target functions on the part of the users. To this end, a target system has been developed that combines several target functions and makes it possible to set different evaluation priorities.

1 State of Research

An industrial cluster is a collection of several neighbouring companies that engage in exchange relationships. This fulfils the basic requirement of spatial proximity, which Chertow identifies as an essential prerequisite for industrial symbiosis (IS), even though internal company and supraregional symbioses are now also the subject of research [1].

In the literature, such clusters are also referred to as Eco Industrial Parks (EIP) [1], [5]. One of the best-known examples is the EIP in Kalundborg, Denmark [6].

MAIWALD ET AL. have developed a model library in the SimulationX simulation environment (www.esi-group.com/products/simulationx) that enables the dynamic simulation of an EIP [7]. Automated evaluation or optimisation is not yet planned, but is considered by the authors to be an important further development of their simulation.

There are numerous key performance indicator systems for EIPs in the literature. These are often classified according to economic, ecological and social key performance indicator types [8].

VALENZUELA-VENEGAS ET AL. describe an important key performance indicator for the topic of resilience. This so-called resilience indicator has a value range of [0, 1], where the value 1 corresponds to the best possible resilience and the value 0 corresponds to no resilience. It is composed of the Network Connectivity Index NCI and the Flows Adaptability Index ϕ [9]:

$$\text{Resilience Indicator} = \frac{1}{2} * NCI + \frac{1}{2} * \phi \tag{1}$$

The Network Connectivity Index NCI of the industrial cluster considers the number of connections. For example, a company may have only one connection to another company or several connections to other companies. The more connections there are within the EIP, the higher the probability that, in the event of one company failing, there will be at least one other source of supply or purchase. The Flows Adaptability Index ϕ assesses the extent to which the lost resource flows can be compensated for if a company fails. This is because such a failure means that the company is no longer available as a buyer of its input products and an emitter of its output products. The proportion of non-compensable losses is determined for each company [9].

As the research project focuses less on social aspects, economic and ecological indicators in particular are incorporated into the assessment method. Due to its importance, the resilience indicator is also explicitly included.

2 Description of the Method

A standardised evaluation is necessary for the classification of simulation results. The steps underlying this article are presented below:

1. Selection of relevant indicators and conversion of these into suitable target functions.
2. Weighting of the target functions under different objectives.
3. Consideration of decision-maker preferences with regard to the weighting of these objectives.

The key figures are selected on the basis of a literature review and a comparison with the information available in the simulation model. Key figures with only limited significance are not taken into account in order to keep complexity to a minimum and thus increase traceability.

The result is a target system with six target functions:

- Z_1: Minimise connection costs
- Z_2: Minimise external electricity procurement
- Z_3: Minimise external heat procurement
- Z_4: Minimise external resource procurement
- Z_5: Maximise the proportion of symbiotic relationships
- Z_6: Maximise resilience indicator

The objective function Z_1 takes into account both the investment and the operating costs of a symbiotic relationship (e.g., district heating pipeline, truck transport), because not every theoretically possible connection is also economically viable.

The objective functions Z_2 to Z_4 address the requirement that, as far as possible, all electricity, heat and resources generated within an EIP should be reused internally. Z_5 also takes this into account by maximising the proportion of symbiotic relationships between individual companies.

The resilience indicator in Z_6 is based on the considerations of Valenzuela-Venegas et al. and has been adapted to the SimulationX environment [9].

	economic	ecological	symbiotic	resilient
Z_1	0.091	0.042	0.043	0.039
Z_2	0.187	0.172	0.098	0.093
Z_3	0.187	0.196	0.098	0.093
Z_4	0.442	0.456	0.098	0.093
Z_5	0.058	0.076	0.374	0.230
Z_6	0.035	0.058	0.289	0.452

Table 1: Weightings of the target functions among the various objectives.

In the second step, the weightings of the objective functions for different objectives (economic, ecological, symbiotic, resilient) are determined using the Analytic Hierarchy Process (AHP) [10]. The objective functions are compared in pairs in terms of their importance for the individual objectives. In total, the weightings (value range [0, 1]) for each objective add up to 1. This step was carried out as part of the project using findings from the literature review (see Table 1). In future, it is planned to involve companies in the evaluation in order to validate the results and adjust them if necessary.

Finally, the point allocation method is used to take the decision-maker's preference into account, i.e. in the final application, a decision can be made as to which weighting should be given to the individual objectives [11]. To do this, 100 points are distributed among the four objectives and a total weighting is calculated according to the objective preference.

3 Application of the Method

The procedure for applying the method is illustrated below using a simple and fictitious example. An industrial cluster with four companies is considered: a steel producer (I1_STEE), a combined heat and power plant (I2_CHP), a paper manufacturer (I3_PAMI) and a plastics manufacturer (I4_CHEM). Each company manufactures a main product, the production of which generates by-products.

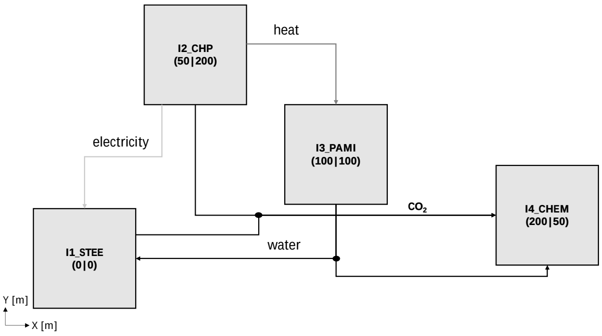


Figure 1: Schematic representation of the symbiotic exchange relationships between the companies (variants 1 and 2).

Figure **1** shows a schematic representation of the exchange relationships between the companies. The black connections represent material connections, red connections represent heat connections and yellow connections represent electricity connections.

Four simulation model variants (SV) of the simulation model are created, which differ in the type of their exchange relationships and are evaluated using the method presented:

- SV 1: Multiple symbiotic relationships for material, heat and electricity flows.
- SV 2: The symbiotic connections are analogous to variant 1, but supplemented by the consideration of losses.
- SV 3: All required energy and material quantities are sourced externally; there is no symbiotic exchange.
- SV 4: Basic structure analogous to variant 2, but water and electricity are sourced externally.

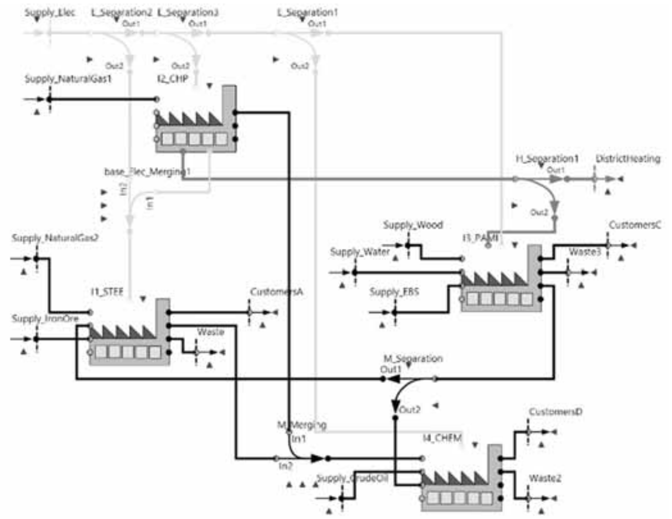


Figure 2: Complete simulation model SV 1.

Figure 2 shows the complete simulation model SV 1 with all exchange relationships taken into account. The simulation library by MAIWALD ET AL. is used, which has been expanded to include a loss component that enables, among other things, the consideration of transmission losses in district heating pipes [7].

Each variant of the simulation model is evaluated using this method. The weighting of the target functions according to Table 1 and the following decision-maker preferences are used: economic: 30; ecological: 30; symbiotic: 25; resilient: 15. These fictitious decision-maker preferences represent a balanced distribution with a slight focus on economic and ecological objectives.

The total utility values shown in Table 2 result in the following ranking of the simulation variants: $SV\,1 \succ SV\,2 \succ SV 4 \succ SV\,3$.

This means that the total utility value of variant SV 1 exceeds the total utility value of all other variants and, with the selected weighting and decision-maker preference, represents the overall optimal solution. It should be noted that the slight differences between SV 1 and SV 2 are solely due to the consideration of transmission losses. These are very small in the example model and can therefore only be shown in the sixth decimal place. This is more of a theoretical distinction with little practical relevance, as losses are to be expected in any case in reality.

Overall, it is clear that simulation variants SV 3 and SV 4 are rated significantly lower. This is because the symbiosis potential is not fully exploited here.

SV 1	SV 2	SV 3	SV 4
0.211665	0.211662	-0.522877	-0.019082

Table 2: Total utility values of the simulation model variants (SV) of the scenarios.

4 Summary

The method presented here for evaluating simulation results in the context of industrial symbiosis makes it possible to systematically evaluate and compare simulation results. Users can weight different objectives according to their preferences. This enables the evaluation of selected manually created simulation model variants, as in section 3. However, a particular strength of the method is that it can also be used to evaluate numerous automatically created simulation model variants and can be used in the context of automated optimisation.

The next step is to check the weightings of the AHP (see section 2). Here, it is advisable to have the pair comparisons carried out by company representatives, for example, in order to obtain valid weightings for possible practical use. The method can then be used in the context of practice-oriented simulative optimisation.

Funding Information

This research project was funded by the German Research Foundation (DFG) for a period from November 2020 to October 2023 (GEPRIS No.: 439187891; SCHM 2689/17-1; RO 2126/8-1).

Deutsche Forschungsgemeinschaft
German Research Foundation

Publication Remark

This contribution is the improved English version of the (German) conference version published in
Tagungsband Kurzbeiträge ASIM SST 2024, p 9-12
ARGESIM Report AR 46
ISBN ebook: 978-3-903347-64-9
volume DOI 10.11128/arep.46

References

[1] Chertow MR. Industrial symbiosis. Literature and taxonomy. *Annual Review of Energy and the Environment*, 2000, 25, 313–337.

[2] VDI Centre for Resource Efficiency. *Industrial symbiosis.* https://www.ressource-deutschland.de/themen/kreislaufwirtschaft/kreislauffuehrung-im-verarbeitenden-gewerbe/unternehmenskooperationen/, accessed on 07.06.2024.

[3] Gabler Business Dictionary. *Resilience.* https://wirtschaftslexikon.gabler.de/definition/resilienz-52429/version-275567, accessed on 07.06.2024.

[4] Kleemann FC, Frühbeis R. Resilient supply chains in the VUCA world. Supply chain management for Corona, Brexit & Co., 2021, Springer Gabler, Wiesbaden.

[5] Martin M, Svensson N, Eklund M. Who gets the benefits? An approach for assessing the environmental performance of industrial symbiosis. *Journal of Cleaner Production*, 2015, Vol. 98, 263–271.

[6] Jacobsen NB. Industrial Symbiosis in Kalundborg, Denmark. A Quantitative Assessment of Economic and Environmental Aspects. *Journal of Industrial Ecology*, 2006, Vol. 10, No. 1-2, pp. 239-255.

[7] Maiwald M, Kosmol L, Pieper C, Schmidt T. ESProNet: A Model Library for the Dynamic Simulation of Industrial Symbiosis. *International Journal of Modelling and Optimisation*, 2020, Vol. 10, No. 1, pp. 1-7.

[8] Fraccascia L, Giannoccaro I. What, where, and how measuring industrial symbiosis: A reasoned taxonomy of relevant indicators. *Resources, Conservation & Recycling*, 2020, Vol. 157, 1-11.

[9] Valenzuela-Venegas G, Henríquez-Henríquez F, Boix M, Montastruc L, Arenas-Araya F, Miranda-Pérez J, Díaz-Alvarad FA. A resilience indicator for Eco-Industrial Parks. *Journal of Cleaner Production*, 2018, 174, 807–820.

[10] Saaty TL. Decision making - the Analytic Hierarchy and Network Processes (AHP/ANP). *Journal of Systems Science and Systems Engineering*, 2004, 13 (1), 1-35.

[11] Khan JA, Ur Rehman I, Hayat Khan Y, Javed Khan I, Rashid S. Comparison of Requirement Prioritisation Techniques to Find Best Prioritisation Technique. *International Journal of Modern Education and Computer Science*, 2015, 7 (11), 53–59.

Conceptualization of a Procedural Model for Selecting Decision Support Methods in Production and Logistics

Katharina Langenbach

Department of IT in Production and Logistics, Technical University of Dortmund, Leonhard-Euler-Str. 5,
44227 Dortmund, Germany; *katharina.langenbach@tu-dortmund.de*

SNE 35(4), 2025, 199-202, DOI: 10.11128/sne.35.sn.10755
Selected ASIM SST 2024 Postconf. Publication: 2024-12-10
Rec. Rev. English Version: 2025-10-28; Accepted: 2025-11-15
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna
ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. The complexity of modern production and logistics results in recurring decisions being made with the help of decision support methods (DSM). However, choosing an appropriate DSM is complex, especially when considering the available data. A procedural model assists practitioners in this selection by providing a structured, comprehensible, and repeatable approach. This paper proposes a procedural model that categorizes questions arising in production and logistics while supporting the selection of suitable DSMs as well as techniques for aggregating and disaggregating data.

Introduction

The processes of planning, implementation, and control within production and logistics necessitate numerous decision-making activities. These decisions vary according to factors such as time horizons and system complexity [1]. The advent of digital transformation has notably increased system complexity [2].
Given the intricate nature of decisions pertaining to complex systems, it is prudent to employ technical support for decision-making.

A variety of methods can be utilized for decision support, each with specific requirements to yield meaningful outcomes. Consequently, the selection of appropriate methods should be tailored to the specific decision support question at hand [1]. This selection process is both challenging and essential [3].

All decision support methods (DSM) are dependent on data. However, this data is often not available at an appropriate level of aggregation, being either excessively granular or overly condensed. Therefore, data aggregation, disaggregation, or in general data transformation (DT) may be necessary.

This paper presents a foundational approach to the selection and implementation of DSMs, while addressing the provision of data at suitable aggregation levels.

1 Foundations

Problem-solving typically involves addressing specific questions. The formulation of questions allows for a clear delineation of problem statements. Depending on the domain or management level, these questions can vary significantly. In the context of production and logistics, typical inquiries include evaluating changes in production capacities or planning impacts [4].

DSMs are particularly beneficial when there are conflicting objectives or when analyzing the impact of individual decisions on other decisions [5]. Three exemplary types of DSMs are outlined below:

Spreadsheet tools or simple mathematical models are frequently employed in production and logistics for efficient quick calculations that capture system dependencies [6]. Simplifications, such as static averages, are often utilized in this context.

For more complex systems involving randomness and interactions, which are critical factors in real-world scenarios, simulation can be employed [7]. However, it is essential to ensure that a question is "simulation-worthy" [4].

Not all simulation characteristics, such as temporal or stochastic elements, are necessary for every query within production and logistics domains [8].

For instance, heuristic or exact approaches may suffice for transport planning tasks [9].

To mitigate the limitations inherent in individual methods, hybrid approaches that combine multiple techniques can be advantageous, particularly during optimization tasks. Optimization strictly requires complete information availability (deterministic models). However, uncertainties are ubiquitous, leading to oversimplified representations of real systems [10]. Therefore, combining DSMs, such as simulation-based optimization, can be beneficial.

Not all DSMs require identical forms or structures of input data; hybrid applications further compound these variations by necessitating tailored adjustments across datasets beforehand. Data quality significantly influences the trustworthiness of outcomes generated by any employed method [11], with suitability, including proper granularity, playing pivotal roles. Techniques facilitating preprocessing transformations serve critical functions in adapting datasets accordingly before deployment stages, thereby enabling valid outputs to consistently emerge [12].

The interdependencies between the question under consideration, the selection of the DSM, and the available data that can be transformed underscores the need for a procedural model to ensure a structured, reproducible, and reliable approach.

2 Conceptual Approach

To address complex questions, a structured, comprehensible, and repeatable approach is essential. The specifics of the questions and the existing boundary conditions must be considered. Nonetheless, a general procedure should be provided to systematically support especially industry users. In the following, a suitable procedural model is proposed for this purpose.

2.1 Decision Process

When a decision is required in the field of production and logistics, it is associated with one or more questions (see Section 1) of varying difficulty. These questions are always addressed while considering boundary conditions such as the time horizon in question or available resources. Based on these questions and boundary conditions, a DSM is selected to facilitate an informed and comprehensible decision.

The implementation of any DSM necessitates data (see Section 1).

At this stage, various challenges may arise. Data is not available in the necessary quantity or often exists at an unsuitable level of aggregation. If the chosen DSM were applied using such data, it would lead to unreliable or even incorrect results. Therefore, the data must be adjusted according to the specific question at hand. If the data is appropriately formatted after this adjustment, or if it was already available in a suitable form, the DSM can be applied, and insights can be derived from analyzing its results. Based on these insights, decisions can then be made.

2.2 Input Variables

As outlined in Section 2.1, several input variables play a significant role both in the selection of DSM and in determining an appropriate DT. The input variables for both subprocesses exhibit similarities and are briefly described below.

The foundation for selecting both the DSM and the extent of required DT lies in the underlying question. This question defines the focus areas for answering inquiries, needed information, provides guidance on prioritizing objectives amidst conflicts, and determines the required level of detail for results, like a yes/no answer or a detailed forecast. However, the question cannot be considered in isolation; it is always tied to boundary conditions such as the available timeframe, accessible resources, and existing knowledge bases. Accordingly, the selection of DSM is always based on both the question itself and its associated boundary conditions.

To implement the chosen method successfully, data is required. The structure and level of aggregation of this data determine whether a DSM can be applied effectively. The type of data fundamentally required is defined by the question at hand. Specifically, the information needed to answer this question, dictates at what level of aggregation this data must exist. Furthermore, DSMs influence structural requirements for data as well as adjustments needed prior to application. Additionally, other boundary conditions, such as resources available for implementing various transformation techniques, must also be considered during this process.

2.3 Grouping

The selection of an appropriate DSMs and DT necessitates a comprehensive analysis of the prevailing circumstances. The complexity of potential questions and their answers vary significantly.

To streamline the DSM selection process, it is advantageous to categorize questions for example by their answer complexity and application domain while considering the associated boundary conditions.

Recommendations regarding the appropriate DSM can be derived by comparing the attributes of the questions with the requirements and strengths of individual DSMs. This process incorporates boundary conditions and their resulting groupings, thereby integrating expert knowledge.

The early selection of a DSM is crucial, as each method imposes specific requirements on the necessary data, particularly concerning data structure and aggregation levels. Boundary conditions (see Section 2.2), especially those related to the underlying question, provide key indicators for determining suitable aggregation levels for data.

Selecting an appropriate DT-method presents significant challenges for users. By creating groups, it becomes possible to select a suitable method in a simplified and comprehensible manner. These groups are established based on content-driven criteria such as data aggregation levels dictated by both questions and boundary conditions of DSMs, consideration of available data, and alignment with the requirements and potentials of various DT-methods.

2.4 Workflow

Based on the considerations outlined in Sections 2.1 to 2.3, a fundamental procedural model emerges, as illustrated in Figure 1. It becomes evident that the selection of a DSM and its corresponding DT-method based on the clients question involves a complex and multi-layered approach. This approach aims to address the underlying question at hand. The selection process is preceded by categorizing input variables (see Section 2.2) into groups (see Section 2.3), ensuring that both the uniqueness of the question and its associated boundary conditions are adequately addressed before proceeding further with implementation.

Only after this step the DSM can be implemented effectively. Following implementation and analysis of results, different scenarios may arise: The question may be fully answered; further refinement may become necessary; or entirely new questions may emerge during this iterative process. This is represented by the dashed arrow in Figure 1. The whole process can therefore be seen as iterative.

The structured workflow depicted in Figure 1 supports transparent and reproducible decision-making for selecting and implementing DSMs. This enables industry practitioners to tackle complex questions within production and logistics effectively.

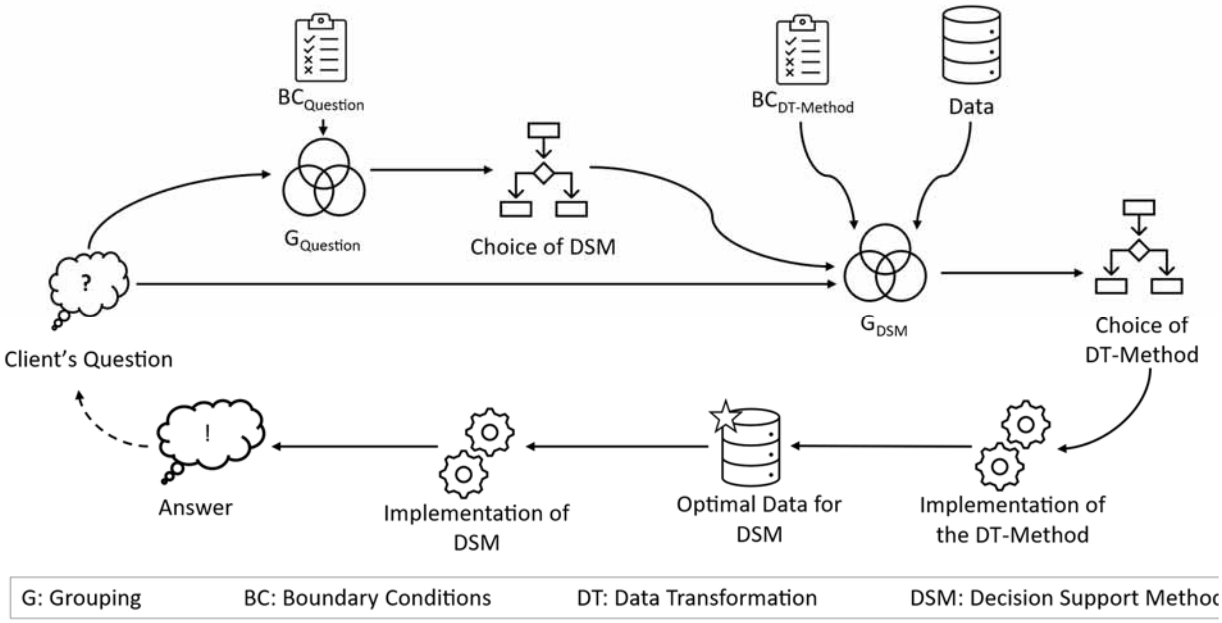


Figure 1: Schematic representation of the procedural model.

3 Outlook

The methodology delineated in Section 2 constitutes an initial framework for the development of a procedural model, thereby highlighting research imperatives that facilitate the comprehensive elaboration and practical implementation of the model.

Initially, it is imperative to analyze the questions to be addressed, classifying their complexity while taking into account boundary conditions. This classification serves as the basis for selecting an appropriate DSM. In this regard, it is also essential to collect and scrutinize the requirements of individual DSMs.

Such analysis enables the alignment of the requirements and potentials of DSMs with the demands of each specific question, thereby aiding in the selection of a suitable method. Existing methodologies for selecting DSMs in production, logistics, and related fields should be considered.

Furthermore, DT-methods must be provided to facilitate the application of selected DSMs. The requirements, strengths, and weaknesses of individual DT-methods must also be evaluated.

To implement the proposed approach in practice, it is crucial to define the individual phases, phase outcomes, and roles within the procedural model. Additionally, automated verification and validation of phase outcomes should be prioritized to enhance the credibility of the generated results and bolster trust in the reliability of this newly developed procedural model.

Publication Remark

This contribution is the revised English version of the (German) conference version published in

Tagungsband Kurzbeiträge ASIM SST 2024, p 33-36
ARGESIM Report AR 46
ISBN ebook: 978-3-903347-64-9
volume DOI 10.11128/arep.46

References

[1] Arnold D, Isermann H, Kuhn A, Tempelmeier H, Furmans K. Handbuch Logistik. Berlin, Heidelberg: Springer 2008.

[2] Lang S, Schenk M, Reggelin T. Towards Learning- and Knowledge-Based Methods of Artificial Intelligence for Short-Term Operative Planning Tasks in Production and Logistics: Research Idea and Framework. IFAC-PapersOnLine 52 (2019) 13, p. 2716–2721.

[3] Renzi C, Leali F, Di Angelo L. A Review on Decision-Making Methods in Engineering Design for the Automotive Industry. Journal of Engineering Design 28 (2017) 2, p. 118–143.

[4] Bicalho-Hoch AL, Ozkul F, Wittine N, Wenzel S. A Tool-Based Approach to Assess Simulation Worthiness and Specify Sponsor Needs for SMEs. In: Feng B, Pedrielli G, Peng Y, Shashaani S, Song E, Corlu CG, Lee LH, et al. (eds.): Proc.2022 Winter Simulation Conference (WSC), p. 1818–1829, Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

[5] Triantaphyllou E. Multi-Criteria Decision Making Methods. In: Pardalos PM, Hearn D, Triantaphyllou E (eds.): Multi-Criteria Decision Making Methods: A Comparative Study. Boston, Massachusetts: Springer US 2000, p. 5–21.

[6] VDI 3633 Blatt 9: Simulation von Logistik-, Materialfluss- und Produktionssystemen: Tabellenkalkulation (TK) im Umfeld der Simulation. Berlin: Beuth, 2019.

[7] VDI 3633 Blatt 1: Simulation von Logistik-, Materialfluss und Produktionssystemen: Grundlagen. Berlin: Beuth, 2014.

[8] Gutenschwager K, Rabe M, Spieckermann S, Wenzel S. Simulation in Produktion und Logistik. Berlin, Heidelberg: Springer 2017.

[9] Günther H-O, Tempelmeier H. Supply Chain Analytics: Operations Management und Logistik. Norderstedt: Books on Demand 2020.

[10] Juan AA, Faulin J, Grasman SE, Rabe M, Figueira G. A Review of Simheuristics: Extending Metaheuristics to Deal with Stochastic Combinatorial Optimization Problems. Operations Research Perspectives 2 (2015), p. 62–72.

[11] Rabe M, Spieckermann S, Wenzel S. Verifikation und Validierung für die Simulation in Produktion und Logistik: Vorgehensmodelle und Techniken. Berlin, Heidelberg: Springer 2008.

[12] García S, Luengo J, Herrera F. Data Preprocessing in Data Mining. Cham: Springer International Publishing 2015.

Segmentation of Bus Driving Data: A Clustering-based Approach to Identify Similar Driving Sections

Anna Schniertshauer[1*], Sven Angerer[2], Andreas Grabow[1], Michael Schlick[1]

[1] Institute for Automotive Systems Engineering, Technische Hochschule Ulm, Prittwitzstraße 10, 89075, Ulm; *anna.schniertshauer@thu.de

[2] Ulm University, Helmholtzstraße 16, 89081 Ulm, Germany

SNE 35(4), 2025, 203-210, DOI: 10.11128/sne.35.tn.10756
Selected ASIM WS2025 Postconf. Publication: 2025-09-10
Rec. Revised: 2025-11-09; Accepted: 2025-11-15
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna
ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. Driving cycles are required for a variety of applications including longitudinal dynamics simulations. For the generation of representative driving cycles, a driving data analysis is indispensable. This paper proposes a method to efficiently segmenting data and subsequently identifying typical trip sections. A first cluster analysis is performed on individual data points using the kmeans++ algorithm. Based on the results, the consecutive data points are segmented into microsegments. Subsequently, these microsegments are being clustered in a second cluster analysis. The results obtained reveal patterns of cluster formations that are similar to those observed in the cluster analysis of individual data points.

Another segmentation, based on the minimum duration of standstill times between two driving sections, enables the identification of typical trips of longer durations. This is achieved by taking the proportions of the microsegments assigned to the same cluster as input variables for the third cluster analysis. Thereby, groups of similar trips can be identified with the typical distribution of microsegment proportions.

Thus, the developed method yields representative trip sections for a driving dataset and thereby forms a basis for generating representative driving cycles both in the research and in the development of simulation-based technologies.

Introduction

To reduce greenhouse gas emissions related to public transit vehicles, such as buses, the transition from conventional internal combustion drive trains to alternative drive trains is necessary. When developing new drive trains, longitudinal dynamics simulations play an important role. In order to fully exploit the capabilities of these simulations, it is essential to use representative driving cycles for different application scenarios as stimuli. This facilitates the evaluation of the vehicle's ability to meet not only the requirements of standardized driving cycles for buses, such as the Standardized On-Road Tests (SORT), but also the requirements of real customer use. To obtain these representative driving cycles, it is necessary to identify typical operating trips by analyzing driving data from buses already operating in the application area. To this end, this paper proposes an approach to effectively segmenting the dataset and finding groups of similar trips using a clustering algorithm. This approach provides a basis for generating representative driving cycles, which can subsequently be used in longitudinal dynamics simulations and support the development of new drive trains.

There are several approaches for analyzing driving data, whereby the dataset is segmented into brief driving sections, hereby referred to as microsegments. One of these approaches involves the establishment of a fixed duration for the segments. Montazeri-Gh et al. use a duration of 150 s [1] whereas Brady and O'Mahony use a duration of 30 s [2]. The disadvantage of a fixed duration is that a single microsegment contains several different driving scenarios. Therefore, in this work, the driving data is segmented into microsegments of variable length which start or end when the driving conditions change.

A similar approach is employed by Langner et al. to extract representative driving scenarios from real-world driving data [3]. The segmentation is based on significant changes in features such as speed limits, street types and curviness [3]. However, there is no detailed description of how these significant changes in the features are identified.

The approach in this paper is to realize the segmentation by clustering individual data points and creating microsegments from consecutive data points that are in the same cluster. Chetouane et al. also perform a cluster analysis of individual data points to identify similar driving episodes from an autonomous driving dataset [4]. However, the post-processing differs from the one chosen in this paper and the identified driving episodes are not used for further characterization of longer driving segments [4].

The methodology employed for the segmentation of driving data and the identification of groups of similar trips is examined first. Then a description of the dataset utilized to develop the method follows. The implementation of the approach is divided into two steps. Initially, data points are segmented into microsegments, which are then clustered. Subsequently, trips are segmented based on a minimum standstill time, followed by a cluster analysis. Finally, a discussion of the results precedes the conclusion of this work.

1 Methodology

With the aim of identifying similar trips from a driving dataset, the method shown in Figure 1 is developed. The raw dataset undergoes a series of preprocessing steps to obtain the desired data quality prior to segmentation. Two segmentation levels are employed for this purpose. On the first level, the data is segmented into microsegments, which are characterized by similar driving conditions. A new segment is created when the considered variables change significantly. To achieve this segmentation, all data points are categorized with a first cluster analysis. For all cluster analyses in this paper the kmeans++ algorithm by Arthur and Vassilvitskii is used [5]. It is an augmentation of the well-known cluster algorithm kmeans, which was developed by Lloyd [6].

Kmeans is a partitioning cluster algorithm and is commonly used to cluster driving data [1, 2, 4, 7]. The algorithm takes the number of clusters k as an input and chooses k cluster centers randomly from the dataset. In the next step the Euclidean distances from each point to the cluster centers are calculated and the data points are assigned to the cluster center with the minimal distance to the point. New cluster centers are calculated as the average of all assigned points and the procedure is repeated. Kmeans++ chooses the initial cluster centers based on the distribution of the input data, which leads to better results and a faster convergence [5]. To choose the number of clusters k, three metrics are considered. The Silhouette score compares the difference of the average distances between a given point and the points within the same cluster and the average distance between that point and the points in the nearest cluster with the maximum of these two values [8]. The values are in a range of -1 and 1, where a higher value indicates a better cluster result [8]. The Davies-Bouldin index compares the size of the clusters with the distance between them. Better partition and separation are achieved with a lower Davies-Bouldin index [9]. The third index used is the Calinski-Harabasz index, which validates the cluster validity by calculating the proportion obtained by dividing the total dispersion between clusters by the total dispersion within clusters across all clusters [10]. A higher index indicates better results [10]. All three metrics are also used by Chetouane and Wotawa to evaluate the results of clustering driving data [11].

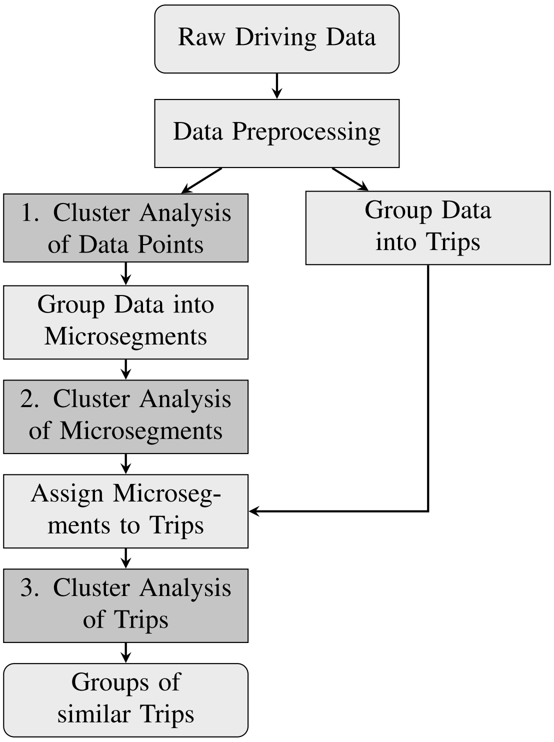

Raw Driving Data
Data Preprocessing
1. Cluster Analysis of Data Points
Group Data into Trips
Group Data into Microsegments
2. Cluster Analysis of Microsegments
Assign Microsegments to Trips
3. Cluster Analysis of Trips
Groups of similar Trips

Figure 1: Segmentation and clustering process.

The first cluster analysis assigns to each data point a cluster index, ci_p. It is imperative to ensure that the occurrence of single data points, possessing a different cluster index compared to their surrounding data points that share the same cluster index, does not result in the initiation of a new microsegment. To this end, a smoothing method is developed. Furthermore, driving sections are identified that consist of data points with highly divergent cluster indices. They are grouped as self-contained microsegments. The values of the individual data points are aggregated so that each microsegment has one value for each cluster variable. The start and end times of each microsegment are taken from the first and the last data point of this microsegment. Subsequently, a second cluster analysis is performed on the aggregated microsegments, which assigns to each microsegment a cluster index ci_m.

On the second level of segmentation, the preprocessed dataset is segmented into trips based on a defined threshold for standstill time. If the vehicle stands still for a duration exceeding the threshold, the current trip ends and a new trip starts as soon as the vehicle moves off again. This results in larger segments than those extracted from the individual data point clustering. Therefore, each trip consists of several microsegments. The start and end time of each trip and microsegment facilitates the temporal assignment of the microsegments to the trips. For each trip, the proportions of time spent by the vehicle in a microsegment of the clusters ci_m are calculated and taken as new variables for the third cluster analysis at the trip level.

The quantity of cluster variables at this level is defined by the cluster number that is derived from the second cluster analysis conducted at the microsegment level. The analysis yields groups of trips with similar proportions of microsegments assigned to the same cluster index ci_m.

2 Data Source and Preparation

The data source used for this paper is the Zurich Transit Bus dataset [12]. It contains driving data from two electric city buses over a period of 3.5 years. To prepare the data for the cluster analysis and to ensure sufficient data quality, a variety of filtering and data preprocessing steps are carried out preceding the clustering. The signals used from the dataset are the time of recording, the latitude and longitude of the vehicle and the velocity of the vehicle.

To filter out incomplete data, the dataset is split into individual days. A day is only included in the analysis if the positional information *or* the velocity signal is not missing for more than 60 s. If the positional information *and* the velocity signal is missing, the day is included since this corresponds to a break. This results in a dataset of 7.5 million data points with a sampling rate of less than 10 s for 99.2% of the data points and a sampling rate of exactly one second for 93.1% of the data points. If the positional information is missing, it is interpolated linearly.

As described by Widmer et al., the dataset contains the raw measurements of the sensors [12]. To counteract on inaccuracies of the latitude and longitude signal introduced by the GNSS sensor, the Valhalla Map Matching API [13] is used to map the coordinates onto the road.

The altitude information for the matched coordinates is retrieved from NASA's publicly available SRTM data [14]. To smooth the altitude signal, the average altitude of the 15 seconds preceding and succeeding each data point is calculated and used for further processing. The velocity of the vehicle is calculated by Widmer et al. based on the signal of rotational speed sensors mounted on the motor shafts of the vehicle and then multiplied with a transmission ratio γ [12]. To counteract slight deviations introduced through this estimation, all values with negative velocity values are set to zero.

To calculate the slope, the difference in altitude is divided by the distance traveled between two data points. For the slope of one data point, the average slope of the preceding and succeeding data point is used. To avoid dividing by a small value of the traveled distance, the slope is set to zero if the velocity of a data point or the preceding or succeeding data point is less than $0.2\ \frac{m}{s}$.

3 Segmentation and Clustering of Microsegments

In order to identify similar microsegments, individual data points are clustered. For this, the slope and velocity of the vehicle are used, as these are typical input variables for longitudinal dynamics simulations. To filter out any outliers, the interpercentile range (ipr) between the 5 and 95 percentile is calculated. Values smaller than $p_5 - 1.5 \cdot ipr$ or larger than $p_{95} + 1.5 \cdot ipr$ are neglected. This filters out 0.2% of the values corresponding to 14 385 data points.

The data is then scaled using a Standard Scaler to remove the mean and scaling it to unit variance [15]. Each data point is clustered using the kmeans implementation by Scikit learn [15] with the kmeans++ algorithm for choosing the initial clusters [5]. This results in each data point being assigned to one of k clusters.

To determine the number of clusters, the scores in Table 1 are considered. As the Silhouette score and Calinski-Harabasz index indicate, the best results are achieved with $k = 4$. The cluster index a data point is assigned to is called ci_p. As can be seen in Figure 2, the clusters with $ci_p = 2$ and $ci_p = 4$ contain data points with moderate slope but differ in velocity. The clusters with $ci_p = 1$ and $ci_p = 3$ contain data points with moderate to high absolute slope values regardless of the velocity.

k	Silhouette	Davies–Bouldin	Calinski–Harabasz
2	0.370	1.099	398668
3	0.374	0.929	435960
4	0.405	0.876	470390
5	0.356	0.885	459099
6	0.360	0.866	447477

Table 1: Scores to determine the number of clusters.

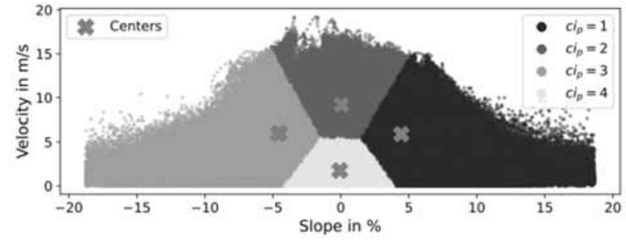


Figure 2: Clustering result of individual data points

In the next step, consecutive data points with the same cluster indices ci_p are grouped together into microsegments.

As can be seen in Figure 3(a), there are data points with a different cluster index than the cluster index that is dominating in this driving section. To address this issue a smoothing process is introduced.

For this, each data point is first transformed into a binary vector representation v, where only the value at the index of the cluster (ci_p) is set to one.

$$v = \begin{bmatrix} v_1 \\ \vdots \\ v_i \\ \vdots \\ v_k \end{bmatrix} \qquad v_i = \begin{cases} 1 & \text{if } i = ci_p, \\ 0 & \text{otherwise.} \end{cases}$$

Since each data point can only be assigned to one cluster, each vector always only has one entry not equal to zero, which means that all cluster indices have the same Euclidean distance to each other and the smoothing can be performed. The average is calculated row-wise in a rolling window. The row with the highest average is then set to one and all other values are set to zero. In a last step the vector can be transformed back into the cluster assignment. An example of this process is demonstrated in Table 2 with a fixed window size of 3 data points. The actual implementation of the algorithm uses a window size of 9 s centered around the current data point. Since the majority of data points have a sampling rate of 1 s, this corresponds to a window size of 9 data points. As shown in Figure 3(b), this method successfully assigns the dominating cluster index to individual points that differ from the dominating cluster index. Depending on the size of the rolling window, even multiple deviating points can be adjusted this way. An important aspect which this method also fulfills is that an already clear transition between microsegments remains intact.

2	2	1	1	3	1	1

$$\downarrow \text{Transformation in } \{0,1\}^k \downarrow$$

$$\begin{bmatrix}0\\1\\0\end{bmatrix} \begin{bmatrix}0\\1\\0\end{bmatrix} \begin{bmatrix}1\\0\\0\end{bmatrix} \begin{bmatrix}1\\0\\0\end{bmatrix} \begin{bmatrix}0\\0\\1\end{bmatrix} \begin{bmatrix}1\\0\\0\end{bmatrix} \begin{bmatrix}1\\0\\0\end{bmatrix}$$

$$\downarrow \text{Rolling Average} \downarrow$$

$$\begin{bmatrix}0\\1\\0\end{bmatrix} \begin{bmatrix}\frac{1}{3}\\\frac{2}{3}\\0\end{bmatrix} \begin{bmatrix}\frac{2}{3}\\\frac{1}{3}\\0\end{bmatrix} \begin{bmatrix}\frac{2}{3}\\0\\\frac{1}{3}\end{bmatrix} \begin{bmatrix}\frac{2}{3}\\0\\\frac{1}{3}\end{bmatrix} \begin{bmatrix}\frac{2}{3}\\0\\\frac{1}{3}\end{bmatrix} \begin{bmatrix}1\\0\\0\end{bmatrix}$$

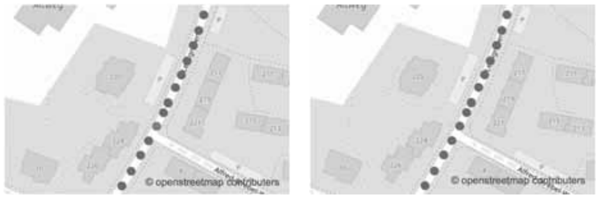
$$\downarrow \text{Assignment} \downarrow$$

$$\begin{bmatrix}0\\1\\0\end{bmatrix} \begin{bmatrix}0\\1\\0\end{bmatrix} \begin{bmatrix}1\\0\\0\end{bmatrix} \begin{bmatrix}1\\0\\0\end{bmatrix} \begin{bmatrix}1\\0\\0\end{bmatrix} \begin{bmatrix}1\\0\\0\end{bmatrix} \begin{bmatrix}1\\0\\0\end{bmatrix}$$

$$\downarrow \text{Transforming to Cluster} \downarrow$$

2	2	1	1	1	1	1

Table 2: Example on how to smooth the cluster result with a rolling window of length 3.



(a) Before Smoothing (b) After Smoothing

Figure 3: Correcting missmatched points after clustering.

In a next step the driving sections are handeled in which a dominant cluster is not apparent. For this, the variance of the smooth clusters is calculated.

If in a window of 9 s the cluster index changes more than four times or the variance is smaller than 0.1 and the cluster index changes more than two times, the data point is labeled as diverse and assigned to an additional cluster index value. After the introduction of this additional cluster index value, the smoothing is performed again. If a microsegment has a duration of less than 5 s, it will be assigned to the microsegment closest to the start or end time of the current microsegment. If the time difference to the previous and next microsegment does not differ, the microsegment will be assigned to which ever has the shorter duration. The entire dataset now consists of 325 311 microsegments with a duration of at least 5 s. To uniquely identify each microsegment, a microsegment index is assigned to each microsegment. This index is employed to aggregate the data and calculate the metrics later used for clustering. The aggregated table also contains the start and end time of each microsegment, which is later used to match the microsegments to the corresponding trips.

To not only capture the average or median slope of a microsegment, the cumulative sum of altitude gain and altitude loss are calculated while aggregating. These variables are now used to calculate the altitude gain per distance and altitude loss per distance. Together with the median velocity of a microsegment these two variables are used to cluster the microsegments.

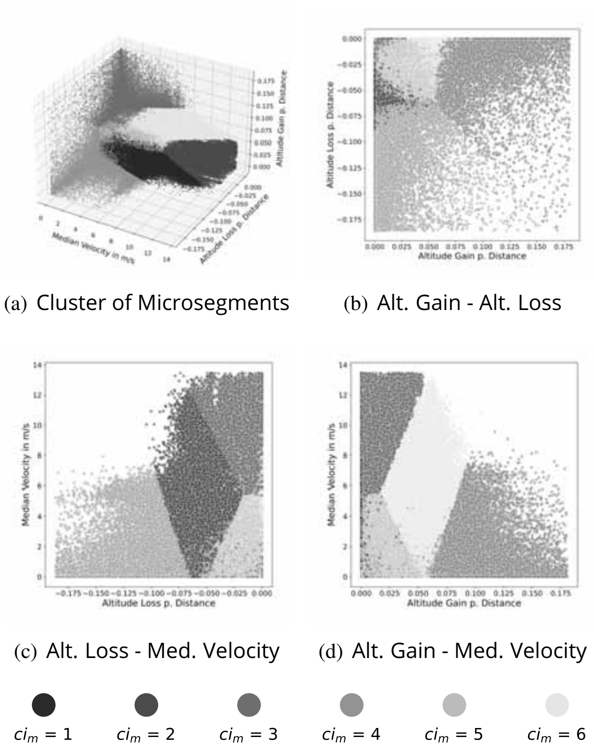
To filter out any outliers, microsegments in which the velocity falls within the bottom or top 5% of values or the altitude loss per distance or altitude gain per distance falls within the top or bottom 3% of values, are not considered. To determine the number of clusters, the metrics in Table 3 are considered. Based on that, a cluster number of $k = 6$ is chosen.

k	Silhouette	Davies–Bouldin	Calinski–Harabasz
2	0.324	1.292	139185
3	0.378	1.008	179274
4	0.385	0.909	207663
5	0.379	0.942	198766
6	0.404	0.876	213039
7	0.391	0.908	202768
8	0.390	0.922	200846

Table 3: Scores to determine the number of clusters.

As can be seen in Figure 4, the clusters are separated by the median velocity, as well as the altitude gain and altitude loss within a microsegment.

The cluster with $ci_m = 2$ contains microsegments with the highest velocities but only little to moderate changes in altitude gain and loss. The clusters with $ci_m = 1$ and $ci_m = 6$ contain microsegments with moderate velocities and are separated by altitude gain and altitude loss. While the cluster with $ci_m = 1$ contains microsegments with moderate altitude loss and almost no altitude gain, the cluster with $ci_m = 6$ contains microsegments with moderate altitude gain and only very little altitude loss. The cluster with $ci_m = 5$ contains the microsegments with low velocity and almost no altitude gain or loss. In contrast to that, the clusters with $ci_m = 3$ and $ci_m = 4$ contain microsegments with velocities up to 6.5 $\frac{m}{s}$. While the cluster with $ci_m = 3$ contains segments with moderate to high altitude gain and moderate altitude loss, the cluster with $ci_m = 4$ contains segments with moderate altitude gain and moderate to high altitude loss.



(a) Cluster of Microsegments (b) Alt. Gain - Alt. Loss

(c) Alt. Loss - Med. Velocity (d) Alt. Gain - Med. Velocity

Figure 4: Cluster Result Microsegments.

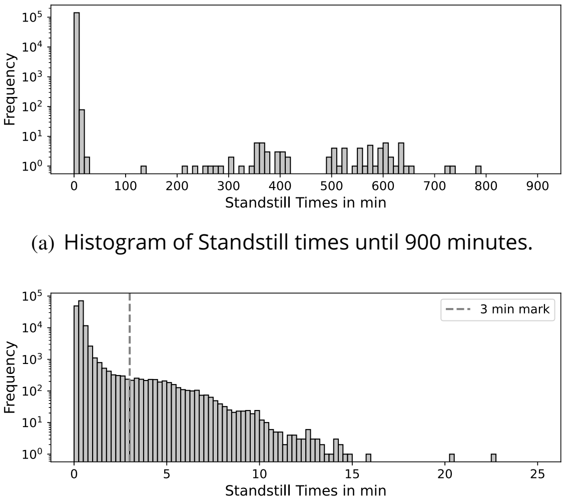
4 Segmentation and Clustering of Trips

In addition to segmenting the dataset based on changes of the velocity or the slope, segmentation is performed at a higher level based on a minimum time of standstill.

A trip ends when the velocity signal is below 0.5 $\frac{m}{s}$ for a certain amount of time. As soon as the velocity signal exceeds this value again, a new trip begins. Figure 5(a) shows a histogram of the standstill times up to a duration of 900 minutes.

This exposes that most standstill times are below 20 minutes, which correspond to short breaks and traffic flow interruptions during daily operation. The standstill times between 200 and 800 minutes most probably reflect the nighttime breaks. However, choosing a minimum standstill time of 200 minutes would result in a total of only 229 trips because only 0.16 % of standstills have a minimum duration of 200 minutes.

Consequently, the minimum standstill time is set at a lower threshold. As can be seen in Figure 5(b) a high proportion of standstill times are below a duration of three minutes. Standstills of less than three minutes are therefore attributed to normal standstill times during operation, for example at bus stops or traffic lights. Standstills that last longer than three minutes account for 2.44% of all standstills and are defined as a segmentation criterion for the trips. This results in a total number of 3 475 trips, 99.42% of which have a duration of less than 300 minutes.



(a) Histogram of Standstill times until 900 minutes.

(b) Histogram of Standstill times until 25 minutes.

Figure 5: Histograms of Standstill times.

A closer look reveals that 10.16% of the trips are shorter than 30 s and 11.08% are not longer than 5 minutes. However, a trip duration of less than 5 minutes is not reasonable. Therefore, all trips of less than 5 minutes are excluded from further analysis. This leaves 3 090 trips for the cluster analysis.

After segmentation, all 325 311 microsegments are temporally assigned to the trip that includes the microsegment. Since trips with a duration of less than 5 minutes are neglected, 0.97% of microsegments cannot be assigned to trips. For each trip, the temporal proportions of microsegments with cluster index ci_m are calculated. As the best result within the cluster analysis of the microsegments is achieved when $k = 6$ is set, it leads to six new input variables for the cluster analysis of the trips. Outliers are identified and excluded by neglecting data points if the value of any of the used variables fall within the 1% lowest or highest values. From 3 090 trips before filtering there are 2 866 left after filtering for the cluster analysis. To determine the number of clusters k with the best cluster results, the metrics in Table 4 are calculated for different values of k. As the Davies-Bouldin and Calinksi-Harabasz Index indicates best results for $k = 6$, a cluster number of six is chosen. In order to show the results of the cluster analysis with six input variables, a radar chart is plotted in Figure 6. The chart visualizes the centers of the six clusters with the cluster indices $ci_t = 1$ to $ci_t = 6$ from the trip-level cluster analysis. Each axis of the radar chart represents one cluster variable. These variables represent the proportions of microsegments that are assigned to the cluster indices $ci_m = 1$ to $ci_m = 6$. The proportions range from 0% in the center of the chart to 50% in the outermost circle. Each cluster on trip-level is represented by one line. As an example, the trip corresponding to the cluster center of $ci_t = 6$ consists of 26% microsegments which are assigned to $ci_m = 6$ and 5% of microsegments which are assigned to $ci_m = 3$. Figure 6 also shows that the proportions for trips represented by $ci_t = 2$ and $ci_t = 4$ overlap significantly. Likewise $ci_t = 1$ and $ci_t = 5$ have similar curves, but differ in the axes $ci_m = 5$ and $ci_m = 6$. The cluster $ci_t = 6$ clearly stands out due to higher values for $ci_m = 6$ and the cluster $ci_t = 3$ differs by an overall flatter distribution. The proportions of microsegments assigned to $ci_m = 2$ are

k	Silhouette	Davies–Bouldin	Calinski–Harabasz
2	0.240	1.804	731.8
3	0.175	1.740	631.8
4	0.183	1.625	598.5
5	0.185	1.534	559.7
6	0.189	1.441	540.9
7	0.174	1.450	504.3
8	0.163	1.459	479.6

Table 4: Scores to determine the number of clusters within cluster analysis on trip level.

the highest, ranging between 35% to 45%, whereas the proportions of $ci_m = 3$ are the lowest with proportions under 9%.

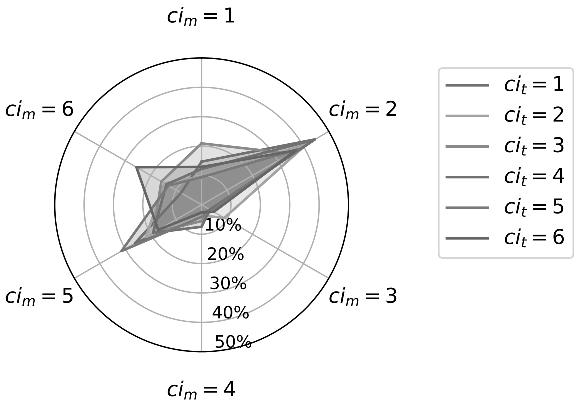


Figure 6: Radar chart of trip-level cluster centers of clusters $ci_t = 1,...,6$ representing temporal proportion of microsegment clusters $ci_m = 1,...,6$.

5 Results and Discussion

The clustering of individual points of driving data from electric city buses leads to the differentiation of four driving scenarios. Cluster $ci_p = 4$ represents scenarios in which the vehicle is not moving or moving at a very slow velocity, for instance at bus stops. By contrast, $ci_p = 2$ indicates normal driving in urban traffic at velocities ranging from 6 to 19 $\frac{m}{s}$. The remaining two clusters contain scenarios in which the bus is either ascending or descending an incline, irrespective of its velocity. When consecutive data points are combined into microsegments based on the preceding cluster results and the microsegments are then clustered again, a similar result is obtained. Again there is a cluster containing microsegments with low median velocity and a cluster containing microsegments with high median velocity. Furthermore, there are two clusters, each predominantly containing microsegments with either altitude gain or loss. However, two additional clusters are identified, representing microsegments with high altitude gain or loss, or both, and low to moderate values for the median velocity. The additional clusters can be attributed to the replacement of slope with the altitude gain and loss and the fact that a driving section is now considered instead of individual points. Therefore sections with altitude gain as well as altitude loss can occur.

Analyzing the results of the clustering concerning trips, it can be seen that the most common microsegments occurring in a trip are microsegments with $ci_m = 2$ and $ci_m = 5$ with over 30% for some trips. Those microsegments correspond to segments with only moderate values for altitude gain and loss. This aligns with the topography of Zurich, as the city center is predominantly flat, while the surrounding areas feature hilly terrain [14]. This also corresponds to the observation that all clusters contain less than 9% of microsegments from $ci_m = 3$ and $ci_m = 4$ which represent microsegments with high altitude difference. The distribution of microsegments across trips is relatively balanced. One reason contributing to this finding is that trips have a duration of at least five minutes and can span over multiple hours and therefore increase the probability of many different microsegments being contained in one trip. Another reason might be that the dataset is limited to city buses within a single city, resulting in routes that are inherently similar. Because of the variables available and to mitigate the complexity of the clustering results only two or three variables, respectively, are considered in the first step. However, these variables are only capable of representing the actual driving sections to a limited extent.

6 Conclusion

The aim of this research is to develop an approach to effectively segmenting driving data and finding groups of similar driving sections using a clustering algorithm. For this purpose a dataset of electric city buses operating in Zurich is used. After data preparation, all data points are being clustered in a first cluster analysis. The categorization of the data points using the cluster results enables a segmentation of the dataset into microsegments. The dataset is aggregated on these microsegments and a second cluster analysis is conducted. The results show, that the characteristic patterns obtained in the first cluster analysis of points can also be seen in the second cluster analysis of microsegments. Additionally, the dataset is segmented into longer driving sections, referred to as trips, based on the duration of standstills between two consecutive driving sections. All microsegments are temporally assigned to the trips and the proportion of microsegments with the same cluster index are computed and used as an input variable for the cluster analysis on trip-level. The results demonstrate minimal variation in the distribution of microsegments across the trips. Considering the setting of Zurich, the segmentation of microsegments and distribution of trips achieves plausible results.

In summary, this approach enables to identify representative driving sections by segmentation of driving data based on cluster analyses. Thereby a basis for generating representative driving cycles is provided, which are essential to fully exploit the capabilities of longitudinal dynamics simulations. In a further study, more variables, for example a congestion index could be taken into account, to potentially represent driving sections more accurately. Another subject of interest could be including driving data from other cities to gain information on the typical trips of city buses in general. Moreover, an investigation on data of intercity buses or coaches could discuss the transferability of the approach described in this paper to data of other bus classes.

Acknowledgement

This research is part of the project "HyCo", which is funded by the Federal Ministry for Digital and Transport under the grant number 03B10305B.

Publication Remark. This contribution is the revised version of the workshop version in ASIM Workshop GMMS/STS 2025 Tagungsband, p. 29-36; ISBN ebook: 978-3-903347-66-3, DOI 10.11128/arep.48

References

[1] Montazeri-Gh M, Fotouhi A, Naderpour A. Driving patterns clustering based on driving features analysis. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*. 2011;225(6):1301–1317. doi: 10.1177/2041298310392599.

[2] Brady J, O'Mahony M. Development of a driving cycle to evaluate the energy economy of electric vehicles in urban areas. *Applied Energy*. 2016;177:165–178. doi: 10.1016/j.apenergy.2016.05.094.

[3] Langner J, Grolig H, Otten S, Holzäpfel M, Sax E. Logical Scenario Derivation by Clustering Dynamic-Length-Segments Extracted from Real-World-Driving-Data. In: *Proceedings of the 5th International Conference on Vehicle Technology and Intelligent Transport Systems - VEHITS*. INSTICC, SciTePress. 2019; pp. 458–467. doi: 10.5220/0007723304580467.

[4] Chetouane N, Klampfl L, Wotawa F. Extracting information from driving data using k-means clustering. In: *Proceedings - SEKE 2021*, Proceedings of the International Conference on Software Engineering and Knowledge Engineering, SEKE. Knowledge Systems Institute Graduate School. 2021; pp. 610–615. doi: 10.18293/SEKE2021-118.

[5] Arthur D, Vassilvitskii S. k-means++: the advantages of careful seeding. In: *SODA '07: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics. 2007; pp. 1027–1035.

[6] Lloyd S. Least squares quantization in PCM. *IEEE Transactions on Information Theory*. 1982; 28(2):129–137. doi: 10.1109/TIT.1982.1056489.

[7] Berzi L, Delogu M, Pierini M. Development of driving cycles for electric vehicles in the context of the city of Florence. *Transportation Research Part D-transport and Environment*. 2016;47:299–322. doi: 10.1016/j.trd.2016.05.010.

[8] Rousseeuw P. Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis. Comput. Appl. Math. 20, 53-65. *Journal of Computational and Applied Mathematics*. 1987; 20:53–65. doi: 10.1016/0377-0427(87)90125-7.

[9] Davies DL, Bouldin DW. A Cluster Separation Measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1979;PAMI-1(2):224–227. doi: 10.1109/TPAMI.1979.4766909.

[10] Caliński T, Harabasz J. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*. 1974;3(1):1–27. doi: 10.1080/03610927408827101.

[11] Chetouane N, Wotawa F. On the application of clustering for extracting driving scenarios from vehicle data. *Machine Learning with Applications*. 2022; 9:100377. doi: 10.1016/j.mlwa.2022.100377.

[12] Widmer F, Ritter A, Onder CH. ZTBus: A large dataset of time-resolved city bus driving missions. *Scientific Data*. 2023;10(1). doi: 10.1038/s41597-023-02600-6.

[13] Valhalla contributors. Valhalla: Open Source Routing Engine for OpenStreetMap. `https://github.com/valhalla/valhalla`. Accessed: 2024-11-17.

[14] NASA. Shuttle Radar Topography Mission (SRTM). `https://www.earthdata.nasa.gov/data/instruments/srtm`. 2000. Accessed: 2024-11-22.

[15] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*. 2011;12:2825–2830.

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

In Memoriam – Agostino G. Bruzzone

With profound sadness, the Federation of European Simulation Societies (EUROSIM) announces the passing of Professor Agostino G. Bruzzone of the University of Genoa, Italy, who died in September 2025.

Professor Bruzzone was an outstanding scholar, leader and friend. As full professor at the University of Genoa, he made enduring contributions to the community of modelling and simulation across many domains including industrial logistics, autonomous systems, maritime and -defence applications.

As President of the Italian society Liophant Simulation and a devoted contributor to EUROSIM, he championed collaboration across Europe's simulation societies and built bridges between academia and practice.

His leadership was marked by a rare combination of deep technical insight, visionary thinking and warm personal engagement. Under his guidance the simulation community worldwide grew stronger, more interconnected and ever more relevant—particularly at the intersection of modelling & simulation with systems engineering, digital twins and autonomous operations.

We honour his memory, his scholarship and his service. In this time of loss, we also reaffirm our commitment to carry forward the work to which he dedicated so much: fostering excellence in simulation science, strengthening our networks and nurturing the next generation of researchers and practitioners.

On behalf of EUROSIM and all its member societies, we extend our deepest condolences to his family, colleagues and students. His influence will long endure in our community and in the many professionals and researchers he inspired.

May he rest in peace.
Francesco Longo, November 2025

Francesco Longo
Full Professor of Industrial Systems Engineering
Director of Modeling & Simulation Center - Laboratory of Enterprise Solutions
DIMEG, University of Calabria
Via Ponte Pietro Bucci, Cubo 45C - Third floor,
Rende (CS) – Italy; T +39 0984 494891| f.longo@unical.it

Calendar of Events

December 7 - 10, 2025, Seattle, WA, USA

2025 Winter Simulation Conference

Look to the Future! Simulation 2050 and Beyond

meetings.informs.org/wordpress/wsc2025/

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

www.msc-les.org/eurosim2026/

together with

I3M 2026

The 23rd International Multidisciplinary
Modelling & Simulation Multi-Conference

www.msc-les.org/i3m2026/

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

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

Abstracts or full draft papers:	**May 31st, 2026**
Acceptance Notification:	**June 10th, 2026**
Camera Ready Papers:	**July 10th, 2026**

www.msc-les.org/eurosim2026/

www.sne-journal.org

www.argesim.org