Order Number: 41845
NIP: 11310850

Doctorate Thesis at Polytechnique Lille
University of Lille 1

Doctoral School of Engineering Science (EDSPI) - 072

Centre de Recherche en Informatique, Signal et Automatique de Lille
Speciality : Robotics and Automation

presented by

Ahmad Koubeissi

entitled

METHODOLOGY FOR UNIFIED MODELING OF SYSTEM OF SYSTEMS OF ENGINEERING

Presented on December 17, 2015

Jury composed of

Reviewers :

Taha BOUKHOBZA	Prof. at University of Lorraine, France
Dominique LUZEAUX	Dr. HDR at Polytechnique, France

Examiners :

Genevieve DAUPHIN TANGUY	Prof. at Ecole Centrale of Lille, France
Abdelaziz BENALLEGUE	Prof. at University of Versailles Saint Quentin en Yveline, France

Director of Thesis :

Rochdi MERZOUKI	Prof. at University of Lille 1, Sciences and Technology, France
Mohammad AYACHE	Dr. at Islamic University of Lebanon, Lebanon

Numéro d'Ordre: 41845
NIP: 11310850

Thèse de Doctorat de l'Université Lille 1
École Polytechnique Universitaire de Lille 1

Ècole Doctorale Sciences Pour lIngénieur (EDSPI)- 072

Centre de Recherche en Informatique, Signal et Automatique de Lille
Spécialité : Robotique et Automatique

présentée par

Ahmad Koubeissi

intitulée

Méthodes de Modélisation Unifiée de Système de Systèmes d' Ingénierie

Soutenue le 17 Décembre, 2015

devant le jury composé de

Rapporteurs :	
Taha BOUKHOBZA	Prof. à l'Université de Lorraine, France
Dominique LUZEAUX	Dr. HDR à Polytechnique, France
Examinateurs :	
Genevieve DAUPHIN TANGUY	Prof. à l'Ecole Centrale de Lille, France
Abdelaziz BENALLEGUE	Prof. à l'Université de Versailles Saint Quentin en Yveline
Directeurs de Thèse :	
Rochdi MERZOUKI	Prof. à l'Université de Lille 1, France
Mohammad AYACHE	Dr. à l'Université Islamique du Liban, Liban

University of LILLE *1, Sciences and Technology*
Polytechnique LILLE
Doctoral School of Engineering Science (EDSPI) - 072
Centre de Recherche en Informatique Signal et Automatique de Lille
Cité Scientifique
Boulevard Paul LANGEVIN
59655 VILLENEUVE D'ASCQ
FRANCE.

To the Soul of My Father...

To My Mother, Wife and Lovely Children...

Acknowledgement

I would like to thank the honorable jury members for providing a lot of insightful comments during the review process and dissertation of my Ph.D. thesis.

I am very grateful to the dearest supervisor, Prof. Rochdi Merzouki, for his continuous motivation, guidance and support which made possible, the successful completion of this work.

All respect and admiration go to my second supervisor, Ass. Prof. Mohammad Ayache, a true brother, partner and friend.

Warm love and support from my family is the main factor of success throughout all years of research. Lastly, praise goes to Almighty Allah whose help and guidance has given me the strength needed to complete this work.

Abstract

The main focus of this thesis is on multilevel graphical modeling of behavior and organization of a set of component systems in a System of Systems concept. In the present work, we model the wireless communication link among component systems, with all its major effects, using a graphical modeling tool for describing flow of power in multi-physics domain called Bond Graph. This model permits identifying mathematically, the quality of signal received at information sink based on model parameter values. We conduct an experiment on cooperative behavior of two humanoids in a system of systems concept and demonstrate how we are able to experimentally measure the parameter values defined in our model.

Next we justify how we are able to evaluate quantitatively, the fault tolerance level of a wireless communication link and introduce the need for redundant links in system of systems. Then we propose a methodology for coupling Hyper Graph, used for modeling the organization of component systems, and Bond Graph in multilevel graphical modeling of system of systems. Finally, we discuss another system of systems with two cooperating component systems, a Unmanned Aerial Vehicle (UAV) and an intelligent autonomous vehicle (robuTAINer), in which the UAV will supply navigation information to the intelligent autonomous vehicle to be able to safely maneuver in a confined space. We develop algorithms for robuTAINer detection and navigation from UAV. Then we practically test our algorithms and analyze the obtained results.

Keywords: *System of Systems, Graphical Modeling, Hyper Graph, Bond Graph, Wireless Communication Link, Fault Tolerance Level, Intelligent Autonomous Vehicle.*

Acronyms

ABM	Agent Based Modeling
ARR	Analytical Redundant Relationships
AWGN	Additive White Gaussian Noise
BG	Bond Graph
CS	Component System
dBm	Decibel Milliwatt
DOF	Degree of Freedom
DPL	Data Packet Loss
FDI	Fault Detection and Isolation
FoS	Federation of Systems
FSM	Fault Signature Matrix
FT	Fault Tolerance
FTL	Fault Tolerance Level
GPS	Global Positioning System
HBG	Hybrid Bond Graph
HG	Hyper Graph
IAV	Intelligent Autonomous Vehicles
ITS	Intelligent Transportation System
LAN	Local Area Network
MAS	Multi Agent System
NL	Noise Level
PTX	Transmitted Power
RSS	Received Signal Strength
RTK	Real Time Kinematics
RTT	Round Trip Time

SMS	State Model Software
SoS	System of Systems
SoSE	System of Systems Engineering
SPP	Single Point Positioning
UAV	Unmanned Aerial Vehicle
UTM	Universal Transverse Mercator
WBG	Word Bond Graph
WCL	Wireless Communication Link

Contents

List of Figures

List of Tables

Chapter 1

General Introduction

Contents

1.1 Framework and context of the thesis

The PhD thesis was prepared within the research group 'Méthodes et Outils pour la Conception Intégrée de Systèmes (MOCIS) [1]', of the 'Centre de Recherche en Informatique, Signal et Automatique et Lille' (CRIStAL - UMR CNRS 9189) [2]'. It is developed in the framework of academic co-supervision partnership between university of Lille1 [3] and Islamic University of Lebanon (IUL) [4]. This work was developed under the supervision of Mr. R. Merzouki, Professor at Ecole Polytechnique Universitaire de Lille 1 and Mr. M. Ayache, Associate Professor at department of Biomedical Instrumentation from IUL.

The research group MOCIS has an extensive experience with integrated design of complex systems, including modeling, structural analysis, control and diagnosis through the use of a unifying tool, called BG. Figure 1.1 shows the topological organization of this group. This integrated design

[1] http://www.mocis-lagis.fr/

[2] cristal.univ-lille.fr

[3] www.univ-lille1.fr

[4] www.iul.edu.lb

can focus on the microscopic system, example of an elementary robot as a physical system, or to the high level macroscopic system, including other robots, environment, communication between robot to robot or robot to environment. Such class of systems is conceptualized with architecture of SoS.

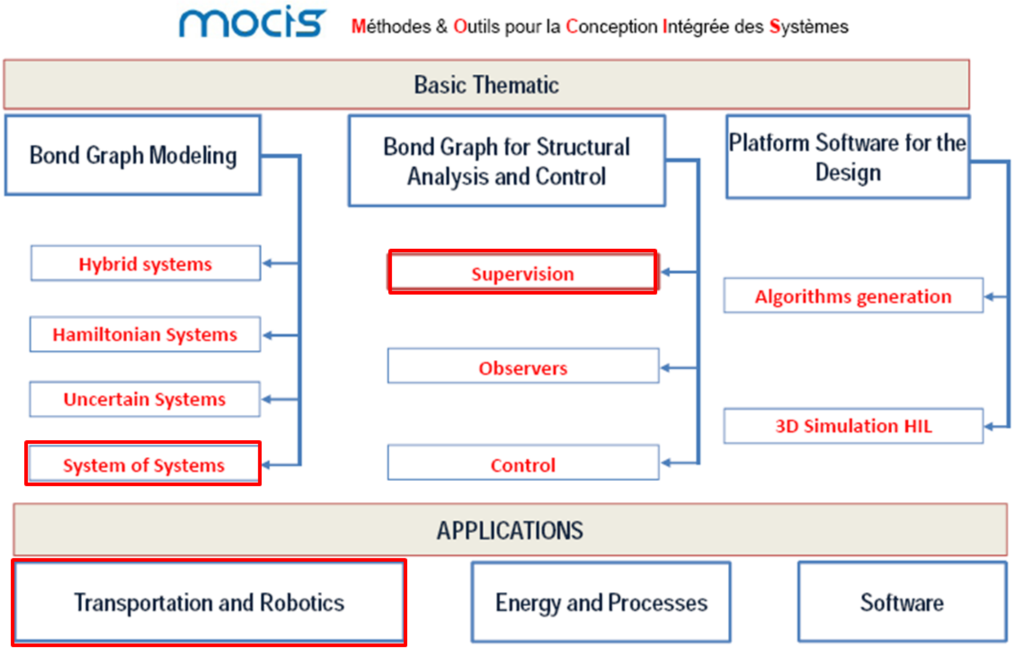


Figure 1.1 – Organization of the works developed within the group MOCIS.

In this context, several research works have been published over the last 6 years in the field of modeling SoS (framework of this thesis) and cooperation control analysis; the works of the group are summarized as follows: (i) Organizational and Behavioral modeling of SoS [Khalil 12], (ii) SoS multi-levels supervision [Kumar 14], (iii) Agent Based Modeling (ABM) of SoS [Soyez 15] (iv) traffic management of a set of intelligent autonomous vehicles [Gelareh 13].

1.2 Thesis Objective

The present work proposes a recent research within the MOCIS group initialized in 2008 (Fig 1.1). It concerns the modeling of a class of large scale systems, represented by the concept of SoSE. This concept is well represented in CRIStAL Lab, in biological environment (Medical exploration or treatment applications) or technological environment (Transport and handling applications) [5].

Thus, the main goal of this PhD work is to elaborate a methodology for unified organizational and behavioral modeling of SoSE. The WCL between CS is also modeled in presence of disturbing perturbations, linear (attenuation) and nonlinear (distortion, data loss and time delays). Using the WCL model, we are able to relate the transmitted waveform by one CS with the received waveform by other CS. This is necessary to identify the Fault Tolerance Level (FTL) of each WCL so that we are able to compare various SoSE configurations in terms of communication reliability and robustness.

1.3 Problem Statement

This PhD thesis addresses the issue of modeling and supervision of large scale systems, often identified by the concept of SoS. A SoSE is a concept describing a set of CS, heterogeneous and independent at management and operation levels, which can coexist and cooperate to accomplish common goals that one CS cannot achieve alone [Maier 96]. These CSs are geographically dispersed without any power exchange between them. The SoSE can be reconfigured, by adding or removing CSs with conserving the same properties of the resulting SoSE [Jamshidi 09c].

SoS consists of components that are spread across multiple hierarchical levels: micro, meso and macro, where for each level, a CS can have multiple structures. Example of Figure 1.2 describes the organization of port operation using IAVs. Such a large scale system, which is difficult to represent analytically requires a graphical model representing the CS for each level and for each structure. The model should represent the CS from its: operational and managerial independencies, interaction with other components, geographical dispersion, and evolutionary development.

A graphical approach was proposed in [Khalil 12] using a HG model-based of SoS. This modeling approach allowed describing the organizational properties of a SoS. Thus, the HG generalizes the concept of the graph by introducing the notion of a hyperedge (Figure 1.3(a)). In this case, the edge of the HG is not subject to a restriction on the number of connected vertices. In other words, the hyperedges don't connect only one or two vertices, but any number of vertices [Berge 87]. The graphical model thus obtained can be used for implementing a strategy of supervision of all components at different levels. [Kumar 14] developed an extended BG modeling to describe the

[5] www.intrade-nwe.eu

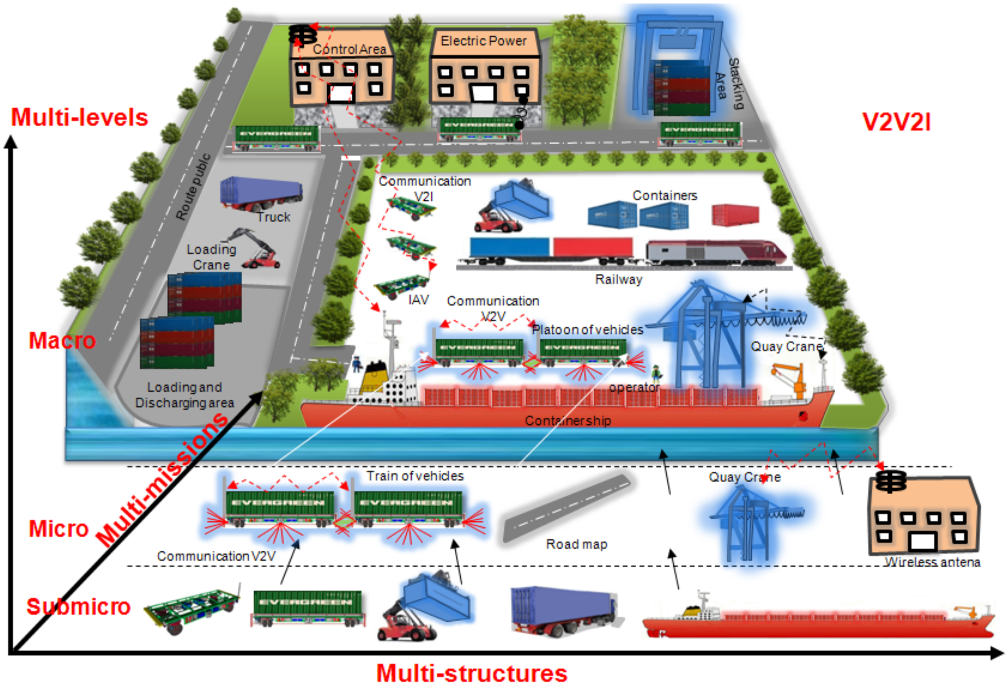


Figure 1.2 – Multi-levels organization of port operation using IAVs.

behavioral property of a SoSE. In this model, the communication between CSs is described by modulated sources of flow or effort (Figure 1.3(b)). This extension is not able to describe the organizational properties of the SoSE, in terms of missions, tasks and macroscopic management. Other model is proposed in [Soyez 15] is based on ABM approach aims to develop a cooperative organization of agents able to satisfy the properties of SoS for supervision purpose (Figure 1.3(c)).

Thus, our **main contributions in this PhD work are the development of BG model of a WCL in SoS concept, including linear and non linear channel effects, and a unified methodology of graphical modeling, that allows describing both behavioral and organizational properties of SoS.** This methodology inspires from the contributions of [Khalil 12] and [Kumar 14], where it allows unifying between the HG-based organizational model and BG-based behavioral model. The aim is to unify the properties of the both graphical techniques in order to obtain multi-levels, multi-missions and multi-structures model of SoS, used either for simulation or supervision objectives.

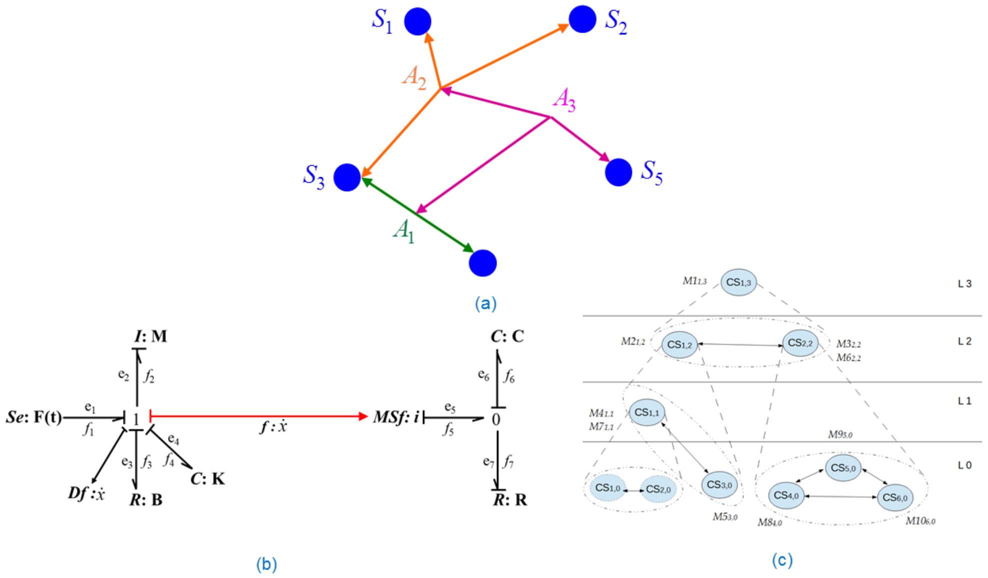


Figure 1.3 – Graphical Modeling approaches of SoSE: (a) HG, (b) BG, (c) multi-agents and multi-levels.

1.4 Publications

The obtained results have made the topic of the following publications:

— Journal

1. A. Koubeissi, M. Ayache, R. Merzouki, 'Unified Method for Modeling a System of Systems: application to intelligent transportation system', IEEE Transactions on Systems, Man, and Cybernetics, Part A, under review.

— International Conference

1. A. Koubeissi, M. Ayache, R. Merzouki, B. Conrard, 'Bond Graph Model-Based for Fault Tolerance Level Assessment of a Wireless Communication Link in a System of Systems Concept', 10th IEEE System of Systems Engineering Conference (SoSE), pp. 358 - 363, Antonio, US, 2015.

2. P. Kumar, R. Merzouki, B. Ouldbouamama A. Koubeissi, 'Bond Graph Modeling of a Class of System of System', 10th IEEE System of Systems Engineering Conference (SoSE), pp. 280 - 28, San

Antonio, US, 2015.

3. W. Khalil, R. Merzouki, A. Koubeissi, B. Ouldbouamama, B. Conrard 'Contribution to System of Systems Modeling', 10th IEEE System of Systems Engineering Conference (SoSE), pp. 182 - 186, San Antonio, US, 2015.

4. A. Koubeissi, R. Merzouki, M. Ayache, 'Bond Graph Model for system of systems Wireless', 2014 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM) July 8-11, 2014, pages: 427 - 432, Besanon, France.

5. A. Koubeissi, M. Ayache, R. Merzouki, 'Multilevel graphical modeling for system of systems: Bondgraph model for a wireless communication link with redundancy', IEEE International Conference on Robotics and Biomimetics, pages 2055 - 2061, Bali, Indonesia, (ROBIO 2014).

6. A. Koubeissi, M. Ayache, M.Abbas, R. Merzouki, B. Conrard, 'Fault Tolerance Level Assessment of a Wireless Communication Link in a System of Systems Concept Modeled using Bond Graph', IEEE International Conference on Advances in Biomedical Engineering (ICABME), Sept 16-18, 2015, pages: 65 - 68, Hadath, Lebanon.

1.5 Thesis Organization

The document is organized in 5 chapters as following:

— The second chapter concerns the state of the art developed to classify methods and techniques for SoS modeling. This chapter allows making the positioning of our PhD work comparing to existing literature in modeling large scale systems.

— The third chapter introduces the modeling of WCL in the context of SoSE, including the characteristics related to data-loss, distortion and data-delay. These disturbing phenomena can affect the overall management of a SoSE concept. The FTL of each WCL is studied in this chapter, in order to evaluate the communication reliability and robustness. A didactic application for cooperation between humanoid robots is developed, to exploit model benefits at microscopic level and at macroscopic level, by developing a simulator that demonstrates how we can assess the FTL of each WCL in SoS.

— The fourth chapter presents a methodology for unified modeling of SoSE including organizational and behavioral properties. This methodology represents the main contribution of modeling SoSE. The model can offer possibilities of supervision based on adequate structural analysis. Of course the fundamental properties in sense of [Maier 98] are demonstrated through the obtained modeling approach.

— The fifth chapter deals with the development of an automatic navigation of an IAV in the port environment based on its cooperation with an UAV. This SoS between IAV and UAV is used to assist the land vehicle in its autonomous navigation. The UAV is used as external GPS information, added as supplementary data fusion for the IAV navigation. The feasibility of the concept of guidance an IAV based on quadrotor UAV system has been demonstrated by CRIStAL in [Merzouki 14].

— Finally, the document is ended with a conclusion, summarizing the main contributions of this Ph-D and discussing future derivative works.

Chapter 2

System of Systems Modeling : State of the Art

Contents

2.1 System of Systems Terminology

SoS is a concept describing a set of independent CSs cooperating to achieve a common task. SoS concept emerged in mid 90s of the past century due to increased complexity in engineering systems. In fact, several definitions exist describing the terminology of SoS depending on their application. However, there is no unified definition agreed worldwide. We start by listing some notable definitions concerning SoS concept:

- [Jamshidi 09a]: *Systems of systems are large-scale integrated systems that are heterogeneous and independently operable on their own, but are networked together for a common goal.*

- [Jamshidi 09a]: *System of systems is a 'supersystem' comprised of other elements that themselves are independent complex operational systems and interact among themselves to achieve a common goal. Each element of a SoS achieves well-substantiated goals even if they are detached from the rest of the SoS.*

- [Maier 98]: *A system of systems is a set of collaboratively integrated systems that possess two additional properties: operational independence of the components and managerial independence of the components.*

- [DAG 10]: *System of Systems is a set of arrangement of independent systems that are related or connected to provide a given capability. The loss of any part of the system will degrade the performance or capabilities of the whole.*

The concept behind SoS differs from that of complex systems. A complex system is a system that exhibits some or all of the following characteristics: feedback, robustness, spontaneous order, emergent and hierarchal organization [James 13]. Although *a complex system is constantly trying to change and evolve over time in unpredictable ways* [Hatch 06], It differs structurally from a SoS, which describes a concept of federation between several CSs, which should realize a global mission with respect to specific properties [Sage 01], [Sage 07]. In fact, there exists five necessary requirements to be satisfied in SoS concept [Maier 96]:

1. **Operational independence of CSs:** Each CS in a SoS is an independent entity by itself. If the CS is detached from the SoS then it can operate successfully and perform operations on its own.

2. **Managerial independence of CSs:** Each CS in a SoS is managed as a separate entity, meaning that the managed CSs can evolve independently without regard for the other CSs that are part of the SoS.

3. **Geographic distribution of CSs:** CSs are dispersed over a relatively wide geographic area depending on the size of SoS. CSs do not

physically interact but may exchange information.

4. **Emergent behavior:** CSs cooperate effectively in acSoS to achieve tasks that one CS alone cannot achieve. In other words, the global mission of SoS ca be only achieved with the cooperative behavior of its CSs.

5. **Evolutionary and adaptive development:** SoS is never fully formed or achieved. Missions allocated to CSs continuously evolve to achieve the global mission of SoS. CSs may be added or removed from SoS without disrupting the global mission.

Definition 2.1: SoSE is an emergent field in engineering dedicated for the study of structural analysis, control, estimation, stability study, filtering, and simulation in SoS concept. SoSE is defined as " *The design, deployment, operation, and transformation of metasystems that must function as an integrated complex system to produce desirable results. These metasystems are themselves comprised of multiple autonomous embedded complex systems that can be diverse in technology, context, operation, geography, and conceptual frame*" [Keating 03].

2.2 System of Systems Applications

Since the introduction of its terminology, the concept of SoS has been applied in several fields including biology, medicine, transportation, management, ship environment, military, agriculture, economy and many other environments. In Table 2.1, we list several publications contributing to SoS concept and their fields of interest.

2.3 Systems of Systems Modeling Techniques

Active studies in SoSE aim at developing an integrated modeling and simulation environment that satisfies the requirements of a SoS with all its complexities. When analyzing a SoS, it is essential to examine several design options to be able to optimize the effectiveness of its execution [Campbell 05].

In modeling SoS, graphical techniques are preferred over other modeling techniques due to their ease of implementation and interpretation in large scale SoS. There exists three major modeling techniques for SoS: state modeling, organizational modeling and behavioral modeling.

Table 2.1 – SoS applications in different domains.

Domain	Publications
Defense	[Dahmann 09]
	[Dickerson 09]
Air Vehicles	[Wilber 09]
	[Colgren 09]
Robotics	[Sahin 09a]
	[Sahin 09b]
Space Exploration	[Jolly 09]
	[Caffall 09]
Medicine	[Wickramasinghe 09]
	[Hata 09]
	[Raad 14]
Environment	[Hipel 09]
	[Agusdinata 09]
Earth Observations	[Shibasaki 09]
Infrastructures	[Thissen 09]
Transportation	[DeLaurentis 09]
Airport Operations	[Nahavandi 09]
Maritime	[Mansouri 09]
Education	[Lukasik 98]
Private Enterprise	[Carlock 01]

State modeling provides a convenient method to model and simulate multiple CS functions in a SoS using State Model Software (SMS). A state model includes a hierarchy for the states of each CS. Each state is either a leaf or a parent with child states. A root state is the parent of all states. The root state does not admit a parent. A child leaf is attached to its parent by means of a logical AND or logical OR gate. A state model must define non-conflicting initial states that the CS must occupy initially. A system might transition form one state to another, once at a time. Figure 2.1 shows a state model for a weapons SoS.

Organizational modeling of SoS permits establishing an architectural representation showing the organization of CSs in SoS. The major benefit is that we are able to apply a supervision strategy using Fault Detection and Isolation (FDI) techniques, to detect faults and perform structural reconfiguration of SoS by isolating faulty systems. The SoS can resume its operation after reconfiguration but with a degraded performance rather than halting the whole system. This is necessary to ensure the reliability and high availability

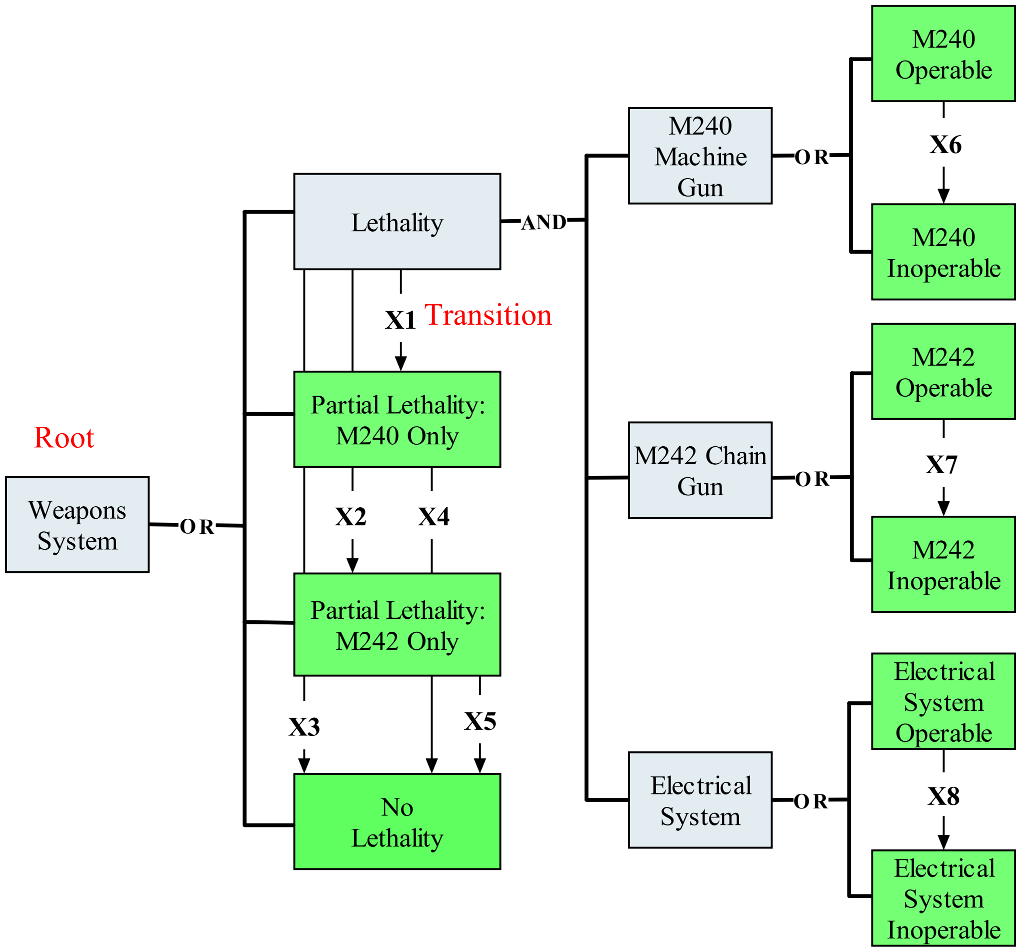


Figure 2.1 – State model for a simple weapons SoS. (Source [Campbell 05])

of the SoS.

Behavioral modeling of SoSs permits describing the behavior of SoS and to associate its performance with behavioral perturbations in physical CSs. It permits defining supervision strategies for SoS to recover from faulty situations in physical CSs.

In the framework of this thesis, we are mainly interested in organizational and behavioral modeling of SoS since our main contribution is to propose a unified methodology for modeling SoS based on coupling organizational and behavioral models. In the next two sections, we discuss organizational and behavioral modeling of SoS in literature.

2.4 Organizational Modeling of System of Systems

2.4.1 Organizational Modeling of System of Systems using Bigraph

In [Wachholder 15], a bigraph-based modeling approach is used to model emergent behavior in SoS applied to computer software systems. A bigraph is composed of two independent graph structures on the same set of nodes. On one hand, a graph describes placing of components by depicting their hierarchy, on the other hand, another graph is used to describe the network of relationships among components. A space model is defined where CSs are linked and located in a common space (location). Figure 2.2 shows a space model (left) and its bigraph representation (right).

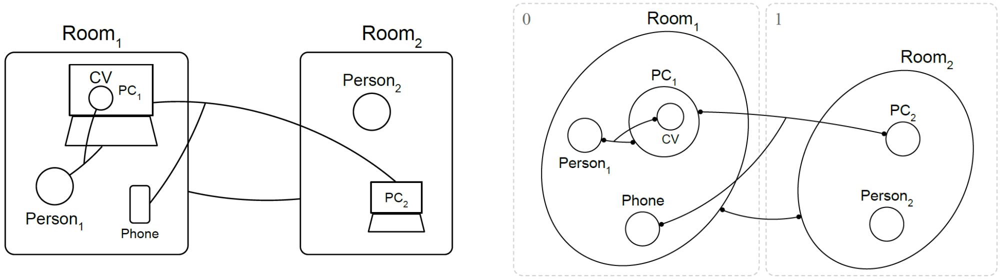


Figure 2.2 – A space model (left) and its bigraph representation (right). (Source [Wachholder 15]).

The main deliverable in [Wachholder 15] is the graphical modeling of the emergent behavior of CSs in a SoS.

2.4.2 Organizational Modeling of System of Systems using Agent Based Model

In [Soyez 15], a SoS is considered s a particular type of Federation of Systems (FoS). An organizational model for CSs in SoS using ABM approach is proposed. The main aim is to develop a cooperative organization of agents capable to satisfy the properties of SoS for supervision purpose. The organizational aspects of CSs are managed with the Agent-Group-Role model. Functional aspects, guiding SoS to accomplish its global goal, are handled via a functional specification.

[Soyez 15] introduced the notion of a **virtual** CS, which is a CS that violates temporarily some of the requirements of SoS. An **elementary** CS is one that cannot be subdivided into another CSs that respect SoS characteristics. The **top** CS is the SoS itself. CSs are modeled using agents. All static and dynamic features of Multi Agent System (MAS), necessary to describe SoSs are stated.

An agent or group capacity is the possibility for an entity to accomplish a goal. An agent role describes the behavior in context defined by a group. $CS_{n,\ell}$ designates the n^{th} CS and its representing agent in the level ℓ. Whereas a group of CS is a modeling of organization of below CSs and how CSs are organized to reach goals in SoS.

The notion of SoSs is applied to Intelligent Transportation System (ITS). In each group, one of the CSs plays the role of ”quayTransporter”, a CS with the capability of transporting goods with a certain capacity per time, and a ”quayTransmitter”, a CS with the capability of transmitting messages to other CSs with a certain capacity per time. Figure 2.3 shows hierarchal organization of groups in SoS in [Soyez 15].

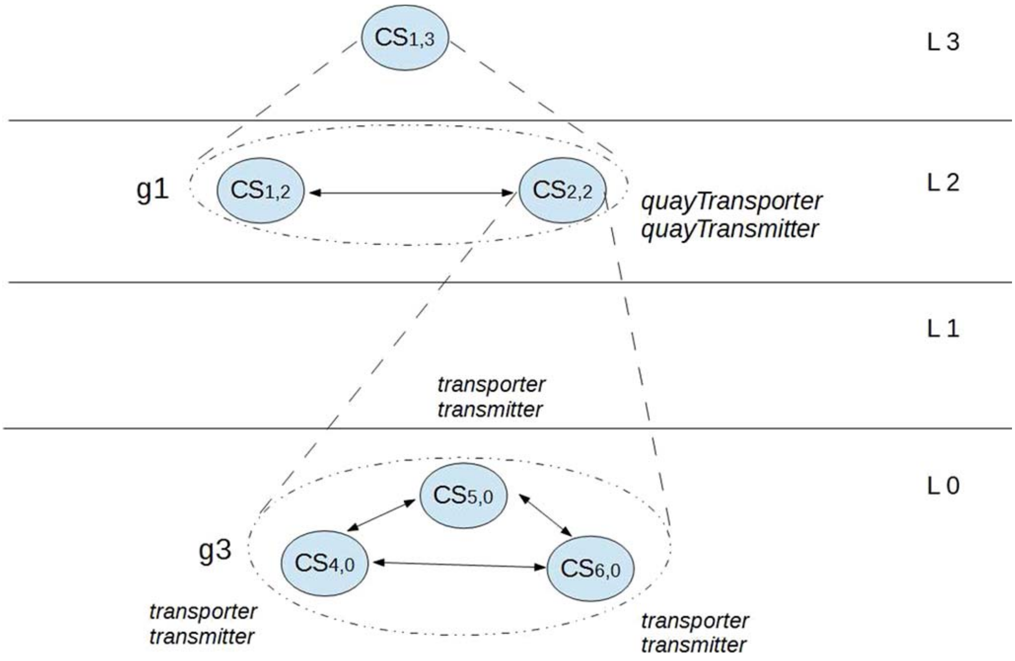


Figure 2.3 – Hierarchal representation of groups of CSs in SoS. (Source [Soyez 15]).

The main deliverable in [Soyez 15] is the possibility to conduct simulations to test control algorithms and to perform system diagnosis in SoS.

2.4.3 Organizational Modeling of System of Systems using Hyper Graph

The organization of several independent systems can be modeled by an ordinary graph. The ordinary graph techniques are usually used to model binary relationships among systems. However, in SoS, there are many interactions between CSs, thus we introduce the notion of hyperedge in HG which can connect any number of CSs. **A HG differs basically from a hyperset in the notion of interaction/communication between hyperedges.** For more information on hypersets, refer to Appendix C.

CSs in a SoS are independent complex operational systems that interact among themselves to achieve a common goal [Jamshidi 09a]. CSs exist in different number of structures i and at different organizational levels j. We denote by $CS_{i,j}$, a CS at level j having structure i (Figure 2.4).

According to [Khalil 12], a SoS where each CS satisfies a finite number of missions $\ell_{i,j}$ can be represented by a nested and valued HG(V,ξ). CSs are shown as vertices V, meaning that V = $\{CS_{i,0}, i \in N^*\}$ in HG. This means that the vertices are represented by the elementary component systems of level 0 and are connected by hyperedges ξ, where $\xi = \{CS_{i,j}, i \in N, j \in N^*\}$. This explains that the hyperedges represent the component systems of all the levels excluding level 0. CSs can exchange information with each other. Missions are represented on HG by valued functions M on hyperedges ξ defined as:

$$\begin{aligned} M &: \xi \mapsto M(\xi) \\ &: CS_{i,j} \mapsto M_{i,j} = \{M_{i,j}^{\ell_{i,j}}\} \end{aligned} \tag{2.1}$$

Since SoS is meant to be a large scale system, then CSs in SoS are organized by hierarchy in various abstraction levels (0 till z) such that each CS has a certain mission that contributes to a larger mission at a higher level. Physical CSs are placed at the lowest level (level 0) and their organization is represented using a CS at higher level. Which means hyperedges ξ will be represented by logical CSs starting level 1. The dynamic interaction of all CSs will lead to successful completion of SoS mission. ith component system at level j will be denoted by CS$_{i,j}$ and its corresponding mission M$_{i,j}$. Figure 2.4 shows a generalized hierarchal representation of SoS using HG.

A HG model of SoS can be represented graphically as hierarchal, or set-based. In set-based representation, organizational CSs, $CS_{i,j\neq 0}$, are represented by sets (hyperedges). Physical CSs, $CS_{i,0}$, connected to organizational

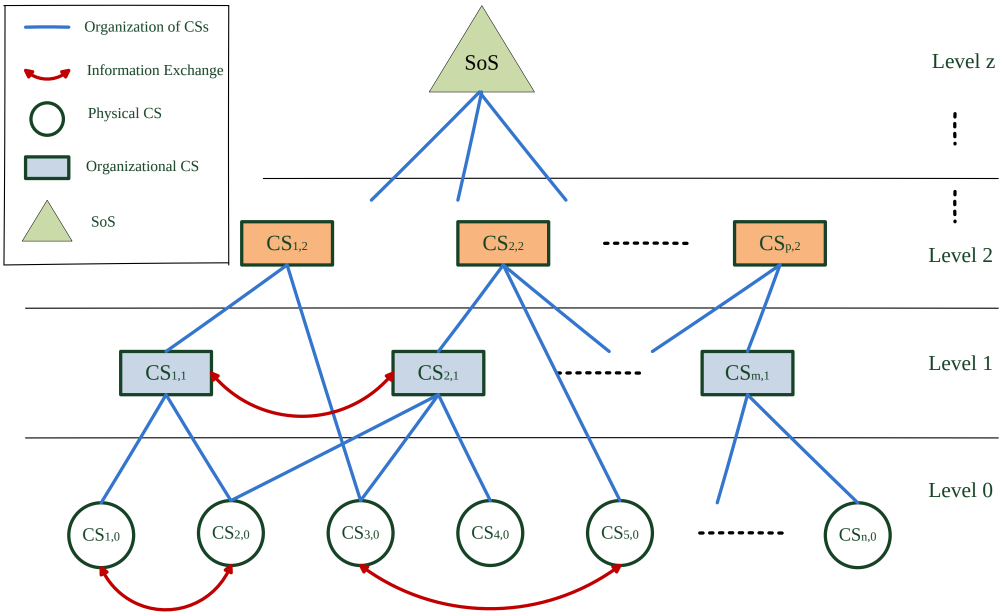


Figure 2.4 – Generalized hierarchal representation of SoS using HG.

CSs, $CS_{i,j\neq 0}$, are represented by vertices inside the sets. Note that **HG differs structurally from a hyperset by the fact of interaction between its hyperedges.** Figure 2.5 shows HG set-based representation (left) and hierarchal representation (right) of SoS.

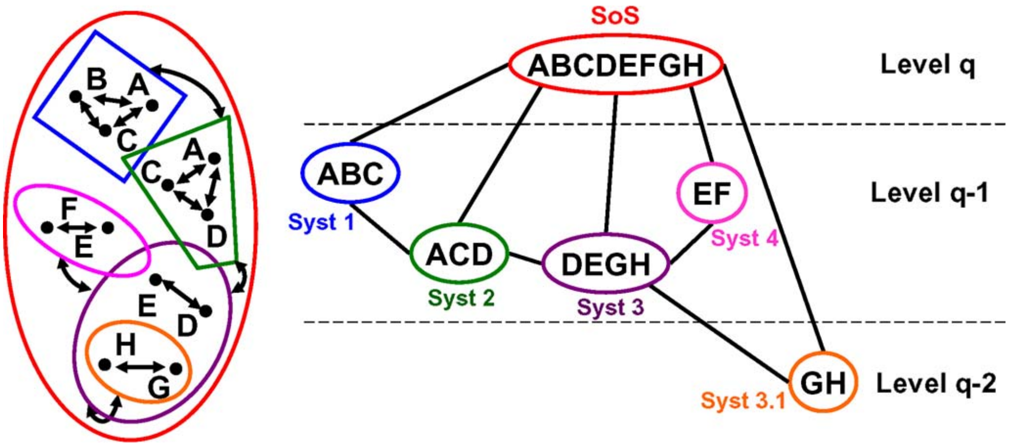


Figure 2.5 – HG set-based representation (left) and hierarchal representation (right) of SoS (Source [Khalil 12]).

In the work of [Khalil 12], the concept of SoS is applied to ITS. The main deliverable of this work is the implementation of a robust supervision algorithm capable of detecting and isolating faulty CSs. Then, a reconfiguration is applied to SoS after detaching the faulty CSs in order to resume operation even if the performance is degraded.

2.4.4 Limitations in Organizational Modeling of System of Systems

In general, **the main limitation in organizational modeling of SoS is that we are unable to model the flow of power in CSs.** Accordingly, we are **unable to relate the overall operation of SoS to elements within CSs**. Practically, this means that a failure in any element, say actuator for example , of a certain CS will lead to unnecessary detach of this CS from SoS since other actuators may have been used in the CSs replacing the failed one.

Another important issue to consider in all organizational models, developed so far, is the absence of WCL modeling among CSs. Recall that **CSs in SoS are dispersed geographically and they need to exchange information to fulfill their cooperative missions**. Accordingly, **CSs are supposed to communicate via a WCL to exchange critical information.** Disturbances over this link, such as attenuation, distortion, time delay and data loss are impossible to realize with the suggested modeling techniques. Practically, a failure in the WCL between two CSs will be regarded as a failure in either of the CSs and an unnecessary reconfiguration will be launched.

2.5 Behavioral Modeling of System of Systems

[Kumar 14] contributes primarily to behavioral modeling of SoS based on another graphical modeling tool, namely BG. The BG tool is used to model power exchange of systems in multi-physical domains. In the case of SoS, it is suited to model the power interaction between CSs at elementary level 0. A BG is a generalization of a graph denoted by the pair $\{S, A\}$ where S is the set of nodes (vertices) representing physical components, subsystems, or other basic elements called junctions and A is the set of power bonds representing bi-directional flow of energy between nodes. For additional information on BG, refer to Appendix B.

In [Kumar 14], modeling is extended to organization of CSs in SoS using same BG tool. Higher level (organizational) CSs are displayed my means of modulated source of flow, MSf, and modulated source of effort, MSe. In addition, the exchange of information signals between CSs is modeled using detector of flow, Df, and detector of effort, De, to emphasize the fact that no power is exchanged, just information. Figure 2.6 shows BG model of organizational CSs in [Kumar 14].

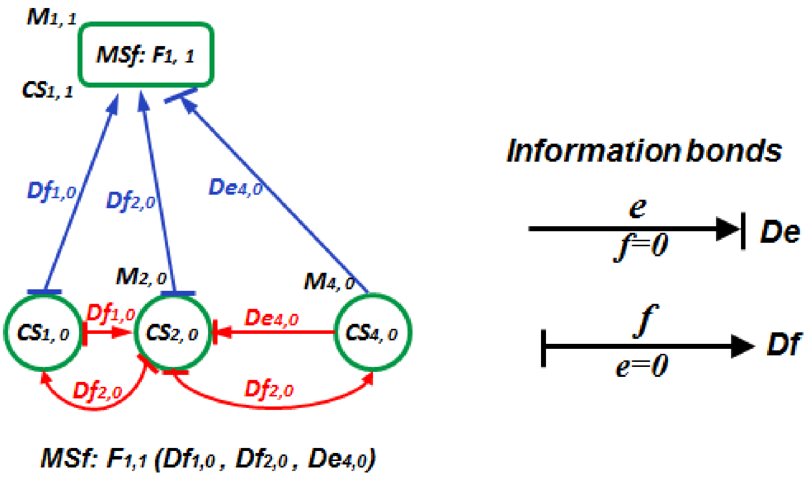


Figure 2.6 – Organizational modeling using BG. (Source [Kumar 14]).

SoS requirements are defined and verified in the proposed multilevel model of [Kumar 14]. The multilevel SoS model is applied to ITS considering traffic dynamic in a platoon of IAVs. Fault detection, isolation and reconfiguration algorithms are defined on the behavioral model of physical CSs in SoS. Figure 2.7 shows a multilevel model of SoS using BG.

The main limitation in modeling technique proposed by [Kumar 14] is the disability to describe the organizational properties of SoS, in terms of missions, tasks and macroscopic management. Accordingly, a supervision methodology is proposed in organizational modeling of SoS ([Khalil 12]) without considering the overall development of FDI technique at elementary level of CSs.

In addition, the communication link between CSs is noted in the model by a detector of effort De or a detector of flow Df. Channel effects are never modeled. Disturbances over this link, such as attenuation, distortion, time delay and data loss are impossible to realize with the suggested modeling techniques.

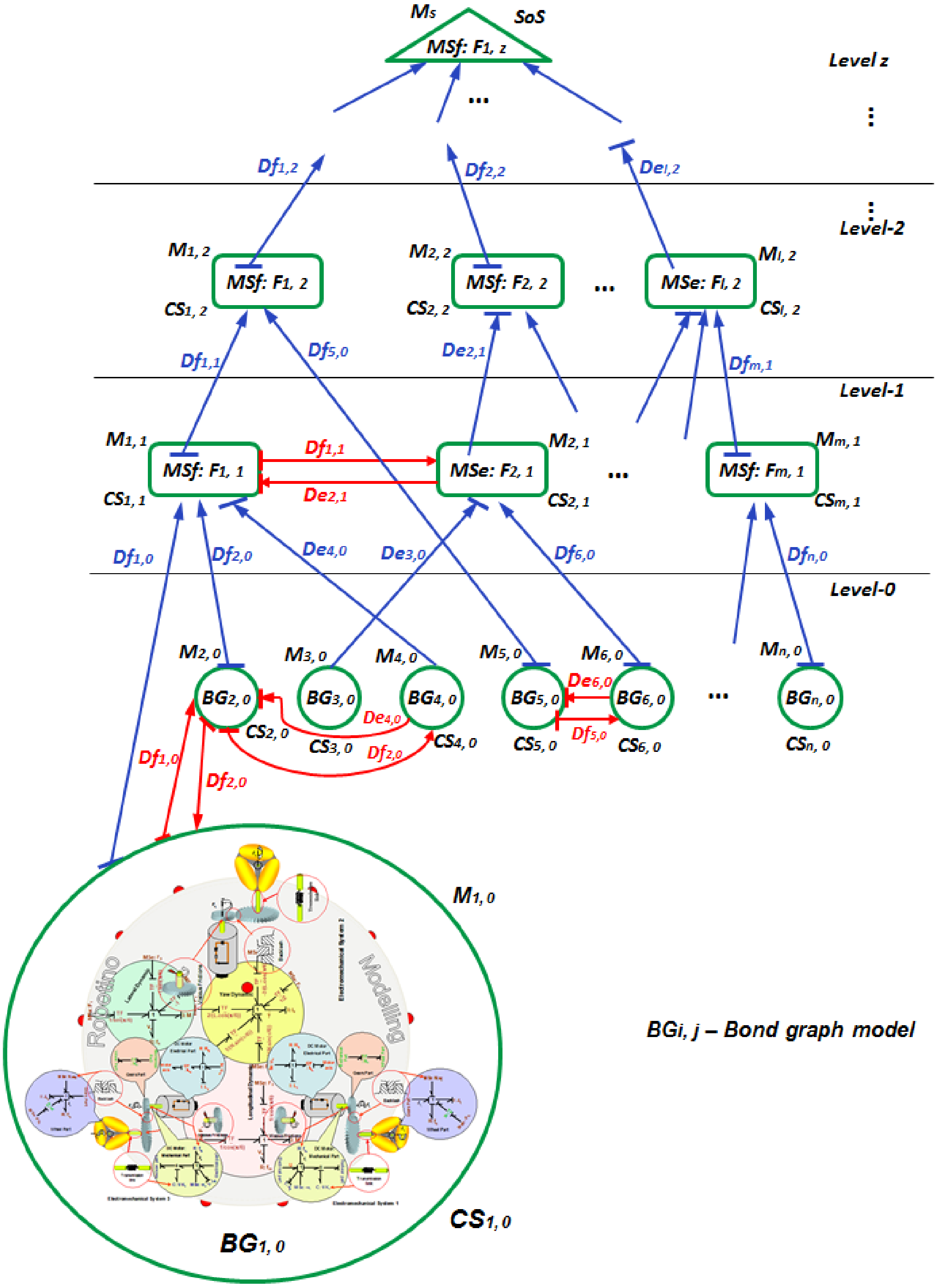


Figure 2.7 – Multilevel model of SoS using BG. (Source [Kumar 14]).

2.6 Conclusion

Throughout this chapter, we introduced the terminology of SoS though there is no unified definition adapted globally. We presented the requirements of SoS as suggested in [Maier 96]. We defined the term SoSE and differentiated between complex systems and SoS. We discussed the implementation of SoS concept in different domains: medicine, transportation, management, ship environment, military, robotics and many other fields.

Modeling a SoS is essential to analyzing its performance, efficiency and evolution over time. In literature, several contributions have been made for SoS modeling. We distinguish three basic types of modeling SoS: State, Behavioral and Organizational. In state modeling ([Campbell 05]), SMS is used to define nodes and allocate their states. Each node (system) can transit from one state to another, once at a time. In organizational modeling, the organization of CSs is shown graphically either by using a HG ([Khalil 12]), or using ABM (Soyez15), or bigraphs ([Wachholder 15]). In behavioral modeling ([Kumar 14]), BG is used to model flow of power in CSs and the organization of CSs in SoS.

In our work, we are mainly interested in organizational and behavioral models since we are seeking to couple both models. The main limitations in existing organizational and behavioral models can be summarized as follows:

- Organizational models are good in representing organization of CSs in a SoS. However, modeling the behavior of physical component systems is not possible using HG, ABM or bigraph. Which means that we are unable to associate the SoS performance with the physical behavior of its components.
- Behavioral model are best suited for modeling behavior of CSs by modeling the transfer of power between constituting elements of a CS. However, with behavioral model, we are unable to describe the organizational properties of SoS, in terms of missions, tasks and macroscopic management
- The WCL between two communicating CSs is not modeled appropriately. Linear (attenuation) and nonlinear (distortion, time delay, data loss) channel effects are not noted. Accordingly, its not clear what would happen in organizational or behavioral models if WCL fails or exhibits some disturbances.

Consequently, **the main contribution of this thesis is a unified methodology for modeling SoS based on coupling behavioral and organizational models** so that we can represent the organization or missions and tasks in SoS and at the same time be able to associate its performance to elements within CSs. In addition, **we model, using BG, the WCL between CSs with common linear and nonlinear channel effects**. Figure 2.8 summarizes SoS modeling techniques and their features.

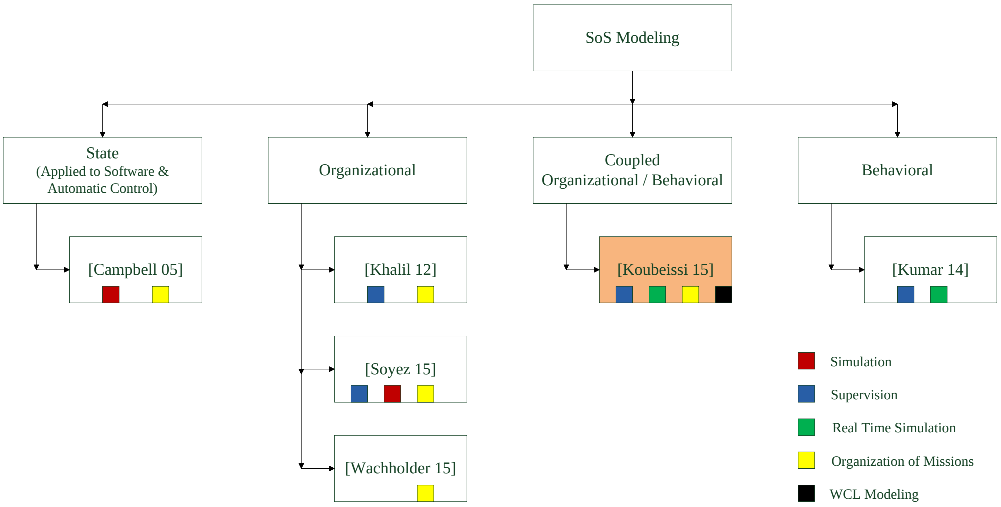


Figure 2.8 – SoS modeling techniques and features.

Chapter 3

Modeling of Wireless Communication Link

Contents

3.1 Introduction

In a SoS, CSs are spread geographically over a relatively wide geographical area with no direct physical interaction. On the other hand, CSs need to exchange information to accommodate with the emergent behavior exploited in SoS and to cooperate effectively to achieve SoS common mission. Accordingly, **a WCL will exist between each couple of behavioral (physical) CSs that need to cooperate and coordinate.**

Modeling the WCL among technological CSs in a SoS is critical for the overall modeling of SoS mission. In the case of absence of WCL modeling in SoS, it is extremely difficult to anticipate what happens at the receiving CS if data is delayed or is lost during transmission. Fault detection algorithms will assume a faulty situation in either of the communicating CSs [Merzouki 10], whereas the source of fault is the communication link itself.

In the notion of SoS, [Kumar 14] modeled WCL between two CSs, using BG, by means of detectors of effort (De) or detectors of flow (Df) (Figure 3.1). Whereas in [Khalil 12], exchange of information between two CSs is noted by a bidirectional arrow on HG (Figure 3.2). In [Soyez 15], the communication link between CSs is modeled mathematically by means of a function that specifies the minimum number of messages transmitted per second. Such CS is called "*quayTransmitter*". Each defined group of CSs in [Soyez 15] must contain at least one "*quayTransmitter*" capable of communicating with other CSs. Consequently, in all SoS models, **linear and nonlinear communication channel effects were not taken into consideration.** By linear channel effects, we mean attenuation, whereas nonlinear channel effects include distortion, congestion, time delays and data loss.

In this chapter, we present one of our main contributions in this thesis, which is a unified BG model of a WCL in a SoS concept. Why did we choose BG to model our WCL? In the case of graphical modeling of SoS, it is suited that the communication link should be represented also graphically, in order to maintain a homogenous representation of the overall model and to describe the main imperfections and disturbances which can occur on the link while all CS are operating. At the second step, when the unified SoS model is obtained, it is an opportunity to apply all the structural analysis on this commonly behavioral and organizational model; Furthermore, in the context of our thesis, we couple two graphical modeling techniques, namely HG and BG. We start by HG hierarchal model of SoS and proceed through our methodology (developed in Chapter 4), till we get a SoS where behavioral and organizational CSs are modeled using BG. So in order to have a unified BG model of SoS, we chose to model the WCL among CSs using BG.

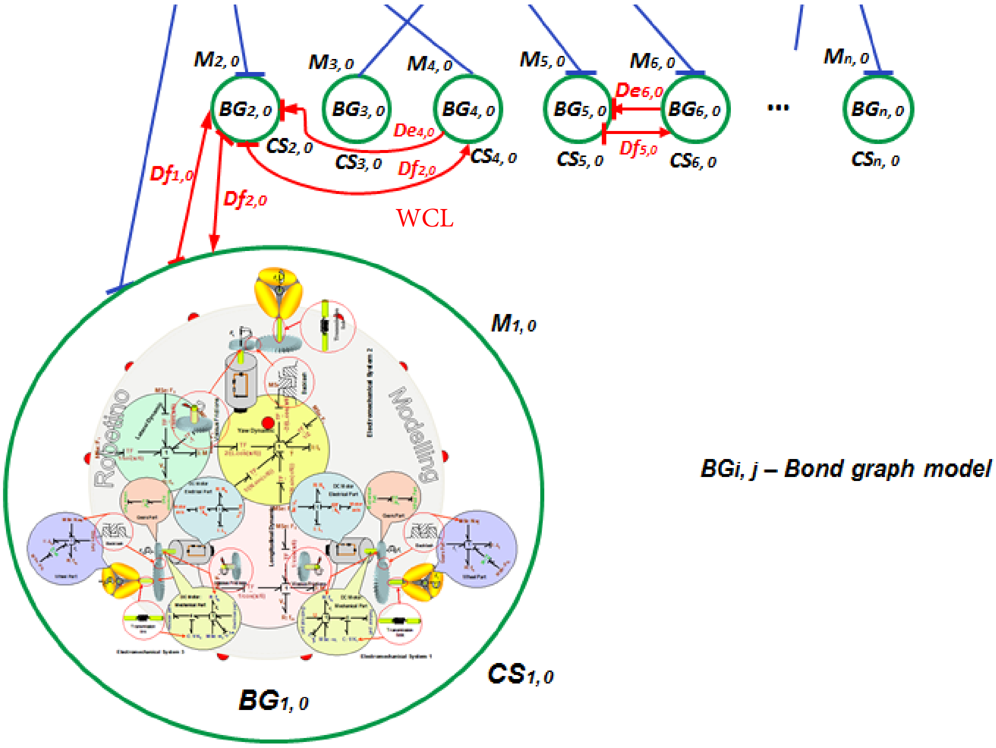


Figure 3.1 – WCL model using BG in [Kumar 14].

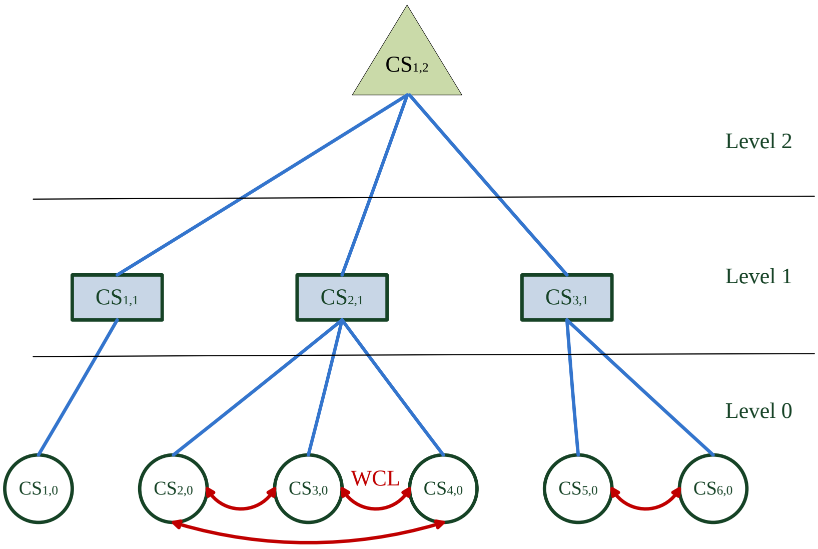


Figure 3.2 – WCL notion using HG in [Khalil 12].

Away from the context of SoS, using BG to model a WCL has been rarely deployed. In [Venkata 12], a Bond Graph model has been developed for wireless static channel as a link with path loss contributing to radio frequency signal attenuation. The model has been part of BG modeling for energy harvesting wireless sensor networks as depicted in Figure 3.3. A single resistive element has been used to model the wireless channel with transmission power, receiver sensitivity and fading margin being factors determining the value of resistance. This model lacks modularity in design. Moreover, channel utilization, time delay and data loss effects were not realized.

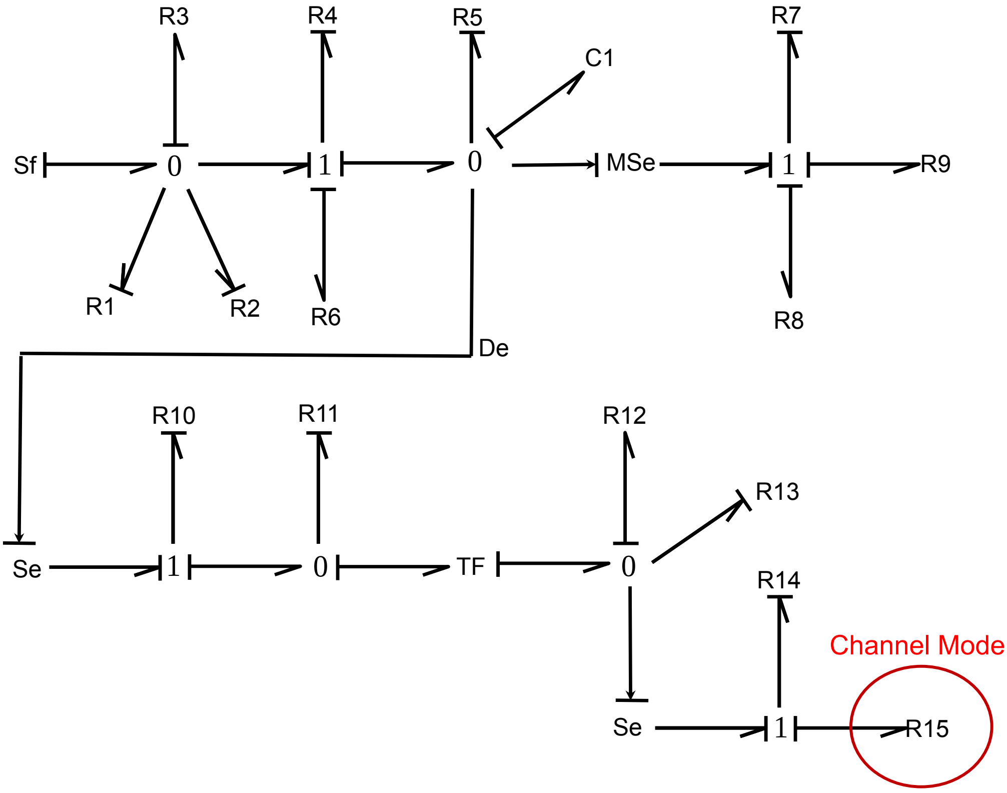


Figure 3.3 – BG modeling for energy harvesting wireless sensor networks in [Venkata 12].

In [Witrant 05], a passive model for a non-homogeneous lossless transmission line based on the classical LC circuit was developed, with time-varying inductance and capacitance. A time-varying impedance and a variable transmission delay result in the communication channel model. However, its validity is limited to wired communication and not wireless (Figure 3.4).

In [Merzouki 10], a wired transmission channel (serial cable) is modeled by a succession of RLC cells, one cell per each meter length of the cable, with R being the resistance, L the inductance and C the capacitance of cable

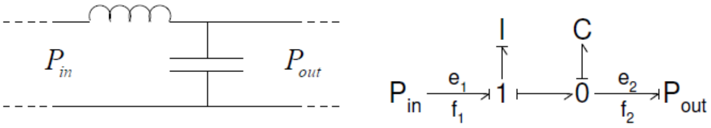


Figure 3.4 – Differential element of a transmission line and its equivalent BG representation in [Witrant 05].

of length l. R, L and C are identified experimentally according to nature of material used in cable. Similar to [Witrant 05], the validity of this model is limited to wired communication (Figure 3.5).

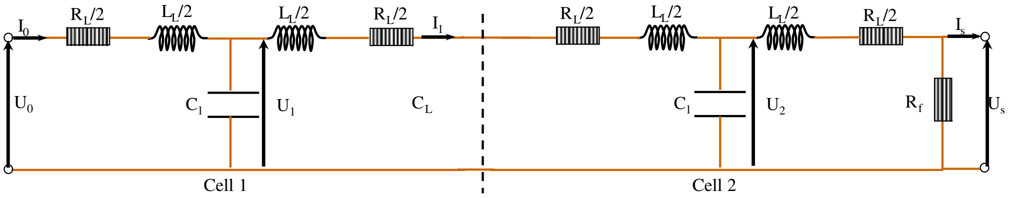


Figure 3.5 – Serial cable with 2 meters of length represented by 2 RLC cells in [Merzouki 10].

In our model, **we consider both linear and nonlinear channel effects. We derive structural and behavioral equations from the BG model and relate the output waveform to considered model parameters. We focus our modeling on only wireless communication link because in the definition of SoS, all the CSs are not linked physically. They only exchange information using a wireless channel.** We test the model in two different set of parameter values. Later in this chapter, we analyze model benefits at microscopic and macroscopic levels for a SoS [Koubeissi 15].

3.2 Channel Effects

In our work, we treated channel effects that are common to almost all WCLs for the benefit of generalization, keeping in mind that our model is slightly oriented towards basic packet-based communications heavily used among **robotized component systems** or other technological CSs equipped with standard Wi-Fi modem. The following channel effects are considered: attenuation (linear), distortion, packet data loss and round trip time (non-linear).

All signals propagating in wireless media are subject to Additive White Gaussian Noise (AWGN) due to collision of electrons in free space leading to decrement in amplitude of signal defined as attenuation and radical change in waveform defined as distortion. AWGN is also known as thermal noise. Additive means that the noise signal components are added to those of propagating signal. White indicates that noise is present at all frequencies. The probability distribution function of noise follows a Gaussian curve. AWGN is totally non-deterministic. Meaning that, knowing power of noise at an instant t_1 does not imply the power of noise at instant t_2. Channel congestion is another important factor in packet-based communications. When a link becomes congested, the probability of losing packets during transmission increases and more time delay will occur in round trips [Haykin 01] [Molisch 11].

In the notion of SoS, we make the following assumptions and definitions:

- **Definition 3.1**: We define C(t) to be the basic analog carrier signal transmitted from $CS_{i,j}$ to $CS_{i',j'}$ where i is the the i^{th} CS in same level and j describes the j^{th} hierarchal level of SoS. This carrier signal may be modulated by an analog or digital baseband message signal M(t), regardless of the modulation type used. The modulated waveform, at the transmitter $CS_{i,j}$ will be represented by S(t), whereas the modulated waveform received by $CS_{i',j'}$ will be denoted by S'(t) (Figure 3.6).

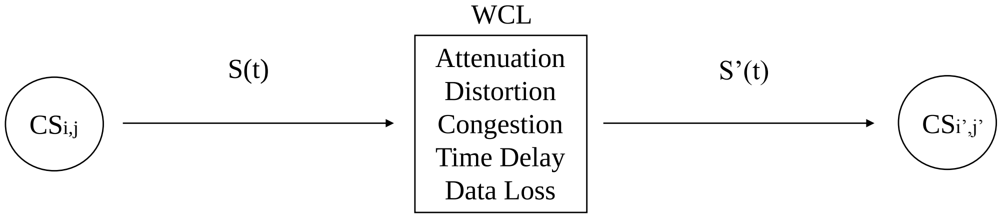


Figure 3.6 – S(t) is signal transmitted by $CS_{i,j}$, received as S'(t) by $CS_{i',j'}$.

- **Definition 3.2**: Attenuation is defined by decrement in amplitude of signal received by $CS_{i',j'}$. We denote by K, the attenuation coefficient such that 0 < K < 1. Thus, the received signal S'(t) is expressed in function of S(t) as follows:

$$S'(t) = K \times S(t) \;\; (Linear\ Scale) \tag{3.1}$$

$$S'_{dB} = S_{dB} + 10 \times Log_{10}(K) \;\; (Decibel\ Scale) \tag{3.2}$$

- **Definition 3.3**: Distortion is a radical change in the waveform of the received signal S'(t) by $CS_{i',j'}$, compared to S(t) transmitted by $CS_{i,j}$. Since distortion is totally random, here we consider the distortion function to be a gaussian function denoted by Gauss(t) and B is the depth of this gaussian function. as B increases, more distortion occurs. u and v are arbitrary chosen floating numbers in the Gaussian function [Guo 11].

  $$Gauss(t) = \exp(-\frac{(t-u^2)}{2v^2}) \tag{3.3}$$

  $$S'(t) = B \times Gauss(t) \times S(t) \tag{3.4}$$

- **Definition 3.4**: Congestion is a measure of percentage utilization of the network. As utilization increases, network congestion will occur [Jacobson 88]. To keep things simple, we define β to be the max number of possible transmissions over a WCL, and α the number of CSs transmitting simultaneously at a time. We define the channel utilization percentage η as follows:

  $$\alpha = \Sigma a_{ij}/a_{ij} = 1 \; if \; CS_{i,j} \; is \; currently \; transmitting \tag{3.5}$$

  $$\eta = \frac{\alpha}{\beta} \times 100 \tag{3.6}$$

  We did not model congestion directly in our BG, instead, we modeled its consequences on the transmitted signal which are time delays and data loss.

- **Definition 3.5**: CSs exchange data in the form of packets. Under severe distortion or heavy congestion, data packets will be lost. Although some upper layer protocols operating in technological CSs will ask for retransmissions (such as TCP/IP) in case of data loss, we are mainly concerned with the number of packets lost during a communication session [Jacobson 88]. To represent data loss in our model we will introduce a periodic square signal x(t) defined as follows:

  $$\begin{aligned} &x(t) = 1 \; for \; 0 < t < T_{on} \\ &x(t) = 0 \; for \; T_{on} < t < T_0 \end{aligned} \tag{3.7}$$

  T_0 is the period of the square signal x(t). T_{on} is the time interval in x(t) for which x(t) = 1. We denote by τ, the time interval in x(t) from T_{on} till T_0 for which x(t) = 0 (τ = T_0 - T_{on}). In our SoS, we define data loss as absence of signal in received waveform S'(t) (zero voltage). In case of data loss, we relate S'(t) and S(t) as follows:

$$S'(t) = S(t) \times x(t) \tag{3.8}$$

It is clear now that τ will govern the amount of data loss in S'(t). As τ increases, more data loss is noted.

— **Definition 3.6**: Time delay or latency is the amount of time needed by a transmitted signal S(t) from $CS_{i,j}$ to reach its destination $CS_{i',j'}$. The larger the distance separating the CSs, the more latency will occur. There are no generic measures available to measure the exact time delay on a communication link. Instead, we ping $CS_{i',j'}$ from $CS_{i,j}$ several times. Round Trip Time (RTT) is the average time taken by ping commands to do a round trip. For that, we assume roughly the time delay to be half that of RTT. in case of time delay t_d, S(t) and S'(t) are related as follows:

$$S'(t) = S(t - t_d) \tag{3.9}$$

3.3 Definition of Model Parameters

Our model includes five input parameters:

1. Transmitted Power (PTX): represents the amount of power transmitted at source in Decibel Milliwatt (dBm). The typical PTX in wireless Local Area Network (LAN) is 30 dBm [Gough 13].
2. Received Signal Strength (RSS): represents the amount of power received at sink in dBm. Minimum RSS for reliable packet delivery is -70 dBm. Whereas the maximum achievable RSS in wireless mode is -30 dBm noted by IEEE802.11 standard for generic Wi-Fi modems normally embeded in robotized CSs [Gough 13].
3. Noise Level (NL): represents the amount of noise power during idle state in dB.
4. Data Packet Loss (DPL): Represents the number of packets lost during transmission and/or reception.
5. RTT: Represents the amount of time, in seconds, needed for data packet to be transmitted and then acknowledged by receiver.

Now our contribution is to relate the attenuation coefficient K with parameters PTX and RSS. Recall that S'(t) = K × S(t). We note:

$$K = \frac{S'(t)}{S(t)} \; (Linear) \tag{3.10}$$

$$K_{dB} = Power(S(t))_{dB} - Power(S'(t))_{dB} \; (Decibel) \tag{3.11}$$

And since PTX is the power of transmitted signal and RSS is the power of received signal, then we may write PTX = $Power(S(t))_{dB}$ and RSS = $Power(S'(t))_{dB}$. Consequently:

$$K_{dB} = RSS - PTX \; (Decibel) \tag{3.12}$$

$$K = 10^{(\mathrm{RSS{-}PTX})/10} \; (Linear) \tag{3.13}$$

Now we relate B, the depth of the Gaussian function responsible for distortion, with NL such that:

$$B_{dB} = \mathrm{NL} \; (Decibel) \tag{3.14}$$

$$B = 10^{(\mathrm{NL})/10} \; (Linear)) \tag{3.15}$$

Which means when NL increases (noisy environment) then B will increase, causing more distortion to transmitted signal.

Next we need to relate τ, the interval in x(t) from T_{on} to T_0 for which x(t) = 0, with the parameter DPL. Recall that as DPL increases, τ gets wider indicating more data loss in the received signal S'(t). We define δ to be an arbitrary time constant. We note:

$$\tau = \delta \times \mathrm{DPL} \tag{3.16}$$

Finally, we relate the time delay t_d in received waveform S'(t) to model parameter RTT such that:

$$t_d = \frac{\mathrm{RTT}}{2} \tag{3.17}$$

The above equation is a rough estimation for time delays since the ping utility calculates RTT, that is, the time to go and come back to a host. It is not necessary to have the time for sending ping packets same as that for receiving ping packets. However, for simplifying matters, we consider it to be half of RTT.

3.4 Model Realization

3.4.1 Bond Graph Model of a Wireless Communication Link

We realize our BG model of WCL taking into consideration modularity in design. Meaning that, modules can be added and removed without disrupting the whole model. All bonds are effort activated meaning that flow value is zero. This is necessary to ensure that we are modeling an information link and no power is being exchanged between component systems.

The model is divided into six modules as follows:

1. **Information Source**: The information signal S(t), generated by ($\mathrm{CS}_{i,j}$), is input at modulated source of effort (MSe) with parameter PTX denoting its power.

2. **Attenuation**: A modulated transformer (MTF) is used to represent the attenuation process, with RSS parameter being the modulating input. Attenuation coefficient, in dBm, is calculated as PTX - RSS.

3. **Distortion**: A Gaussian noise block is invoked to achieve distortion. The amplitude of distortion is governed by NL. The attenuated signal effort is added to noise by means of '1' junction.

4. **Data Loss**: Data loss is represented by a modulated transformer whose modulated input is driven by a pulse wave. The duration of pulse wave is inversely proportional to DPL.

5. **Time Delay**: A time delay block is added with parameter RTT/2 suggesting time delay interval.

6. **Information Sink**: The output of time delay block is carried by means of a '0' junction to the unity resistive element representing the signal received by ($\mathrm{CS}_{i',j'}$).

The WBG of our model is depicted in Figure 3.7. It consists of six main modules stated previously: Information Source, Attenuation, Distortion, Data Loss, Time Delay and Information Sink. Power variables {e,f} are denoted at input and output of each module. S(t), the signal transmitted by

$CS_{i,j}$, denotes the information source, whereas S'(t), the received version of S(t) by $CS_{i',j'}$, denotes the information sink.

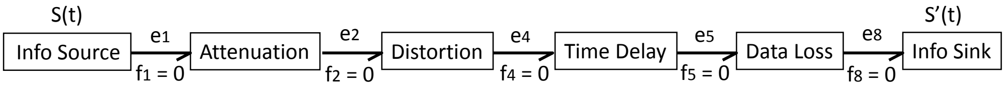


Figure 3.7 – WBG model of WCL.

The complete behavioral model of our WCL is shown in Figure 3.8. Note that all flow variables are null since all bonds are efforts activated to ensure no power is being transferred from one CS to another but only information.

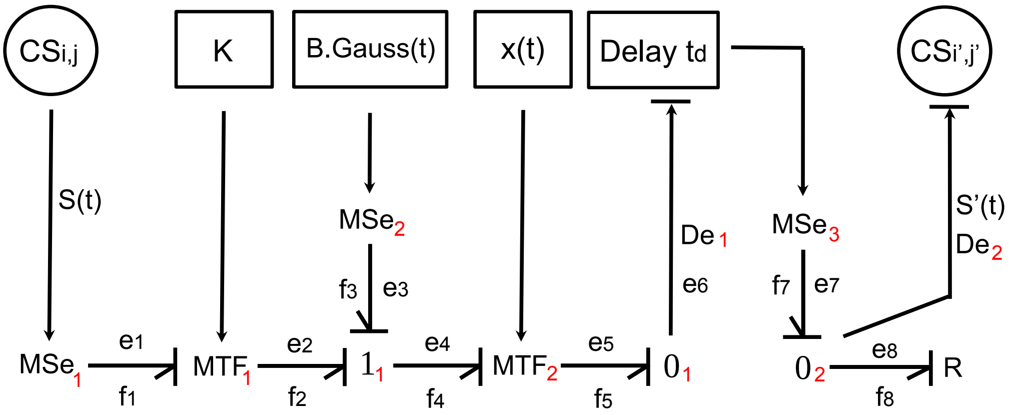


Figure 3.8 – BG model of WCL.

3.4.2 Model Equations

With every BG model, you can derive two sets of equations: Structural and Behavioral. Structural equations are deduced from basic structures in BG such as 0-type junctions, 1-type junctions, transformers and gyrators. Whereas Behavioral equations are basically deduced by relating output to input variables in structural equations.

We start first by structural equations derived from our BG model. The input signal at the modulated source of effort MSe_1 is S(t), the signal transmitted by $CS_{i,j}$. The modulated transformer MTF_1 denotes the attenuation process, with K being the attenuation variable. We note the following equations:

$$e_2(t) = K.e_1(t) \tag{3.18}$$

$$f_2(t) = K^{-1}.f_1(t) \tag{3.19}$$

The input signal at the modulated source of effort MSe_2 is B × Gauss(t), the signal representing distorting noise. At junction 1_1, the attenuated waveform is added to the distorting noise. We note the following:

$$e_4(t) = e_2(t) + e_3(t) \tag{3.20}$$

$$f_4(t) = f_2(t) = f_3(t) \tag{3.21}$$

The modulated transformer MTF_2 denotes the data loss process, with x(t) being the periodic square signal described previously. We note the following equations:

$$e_5(t) = e_4(t).x(t) \tag{3.22}$$

$$f_5(t) = f_4(t).x^{-1}(t) \tag{3.23}$$

At junction 0_1, the input signal, being attenuated, distorted with loss of data is carried to a time delay block using Laplace transform to impose a time delay of t_d:

$$e_6(t) = e_5(t) \tag{3.24}$$

$$f_6(t) + f_5(t) = 0 \tag{3.25}$$

The input signal at the modulated source of effort MSe_3 is the delayed signal. At junction 0_2, we note:

$$e_8(t) = e_7(t) \tag{3.26}$$

$$f_8(t) + f_7(t) = 0 \tag{3.27}$$

The resistance R is used to fix the causality of the junction. A detector of effort De_7 will detect the received signal S'(t) at the information sink at $CS_{i',j'}$.

Concerning the behavioral equations, we substitute the values of all effort signals with their corresponding values, and relate the output signal of the model S'(t) to input signal S(t).

At MSe_1:

$$e_1(t) = S(t) \tag{3.28}$$

$$f_1(t) = 0\ (effort\ activated\ bond) \tag{3.29}$$

At MTF_1:

$$e_2(t) = K.e_1(t) = K.S(t) \tag{3.30}$$

$$f_2(t) = K^{-1}.f_1(t) = 0 \tag{3.31}$$

At junction 1_1:

$$e_4(t) = e_2(t) + e_3(t) = K.S(t) + B.Gauss(t) \tag{3.32}$$

$$f_4(t) = f_2(t) = f_3(t) = 0 \tag{3.33}$$

At MTF_1:

$$e_5(t) = e_4(t).x(t) = (K.S(t) + B.Gauss(t)).x(t) \tag{3.34}$$

$$f_5(t) = f_4(t).x^{-1}(t) = 0 \tag{3.35}$$

At junction 0_1:

$$e_6(t) = e_5(t) = e_4(t).x(t) = (K.S(t) + B.Gauss(t)).x(t) \tag{3.36}$$

$$f_6(t) + f_5(t) = 0 \Rightarrow f_6(t) = 0 \tag{3.37}$$

At time delay block:

$$e_7(t) = e_6(t - t_d) = (K.S(t - t_d) + B.Gauss(t - t_d)).x(t - t_d) \tag{3.38}$$

$$f_7(t) = 0 \tag{3.39}$$

At junction 0_2:

$$e_8(t) = e_7(t) = (K.S(t - t_d) + B.Gauss(t - t_d)).x(t - t_d) \tag{3.40}$$

$$f_8(t) = 0 \tag{3.41}$$

finally, S'(t) = $e_8(t)$. Hence we relate output to input signal:

$$S'(t) = (K.S(t - t_d) + B.Gauss(t - t_d)).x(t - t_d) \tag{3.42}$$

3.4.3 Model Testing

Now after we have related the received waveform S'(t) to the transmitted waveform S(t) using structural and behavioral equations derived from our BG model of WCL, we are able to depict the waveform S'(t) analytically and graphically at the receiver side. This is important if we need to anticipate the robustness and reliability of a WCL between two CSs or if we need to do further signal processing.

In this section we provide a numerical application on our model parameters PTX, NL, RSS, DPL and RTT and consider S(t) to be a basic cosine signal, with angular frequency of 10 rad/s and peak-to-peak voltage of 3V, for testing purposes. S(t) could be chosen to be any other random signal but we chose it to be cosine for the simplicity of comparing it with the received waveform s'(t). We draw accordingly the waveform of the output waveform S'(t). In the next section, we demonstrate how we can measure the parameter values by experimentation in the context of SoS.

We provide numerical values for parameters in two different cases. Table 3.1 lists the values of parameters in each case. In case I, we consider a slightly noisy environment (NL = -50 dB) with minimum data loss and reasonable time delays. Whereas in case II, we consider more noisy environment (NL = -70 dB) with more data loss and significant time delays.

Table 3.1 – Parameter values for two different cases.

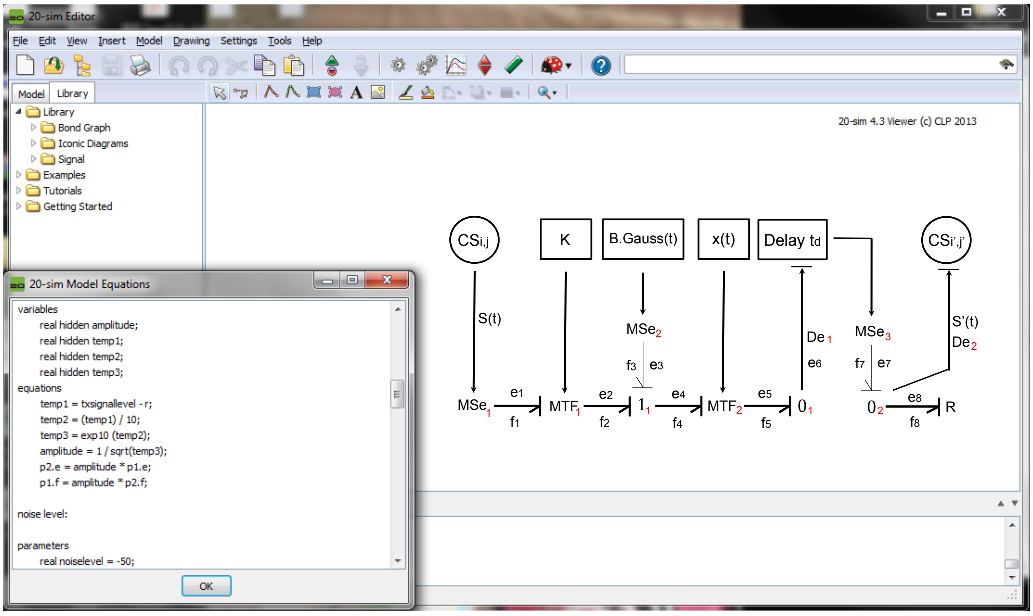
	PTX	RSS	NL	DPL	RTT
Unit	dBm	dBm	dB	Packets	ms
Case 1	30	-20	-50	5	100
Case 2	30	-40	-70	10	800

Simulation results are built automatically using 20Sim software and exported as comma-separated-values file. We then use Matlab to plot the input and output waveforms, S(t) and S'(t) respectively. Figure 3.9 shows the model implemented using [20Sim] software.

The transmitted signal S(t) by $CS_{i,j}$ and the received signal S'(t) by $CS_{i',j'}$ for parameter values in case 1 and case 2, represented in table 3.1, are presented in Figure 3.10(a), 3.10(b) and 3.10(c) respectively in the time domain.

We note the following:

1. The received signal waveform in case 2 is more attenuated due to the fact that RSS in case 2 (-40 dBm) is lower than in case 1 (-20 dBm).
2. The received signal waveform in case 2 is more distorted due to the



```
variables
    real hidden amplitude;
    real hidden temp1;
    real hidden temp2;
    real hidden temp3;
equations
    temp1 = txsignallevel - r;
    temp2 = (temp1) / 10;
    temp3 = exp10 (temp2);
    amplitude = 1 / sqrt(temp3);
    p2.e = amplitude * p1.e;
    p1.f = amplitude * p2.f;

noise level:

parameters
    real noiselevel = -50;
```

Figure 3.9 – BG model of WCL implemented on [20Sim].

fact that NL in case 2 (-70 dB) is greater than in case 1 (-50 dB) which indicates more Gaussian noise.

3. In case 2, DPL (10) is higher than in case 1 (5), which implies that in case 2, more packets are being lost.

4. In case 2, RTT (800 ms) is higher than in case 1 (100 ms), which implies that in case 2, more latency occurs in the received signal. This can be depicted in Figure 3.9(c), which indicates more latency before S'(t) reaches CS$_{i',j'}$.

The amplitude of transmitted signal $\mid S(f) \mid$ in frequency domain by $CS_{i,j}$ and the received signal $\mid S'(f) \mid$ by $CS_{i',j'}$, in dBm, for parameter values in case 1 and case 2, are now presented in Figure 3.11(a), 3.11(b) and 3.11(c) respectively in the frequency domain. We can clearly see that the in case 2, $S'(f)$ has been severely distorted by noise. Figure 3.12 shows the phase spectrum of S(t) and S'(t) for the two cases I and II. Note that plots in Figures 3.12(b) and 3.12(c) appear smoother than that in 3.12(a) since phase angle edges are wrapped (smoothed) automatically by Matlab and that can be deduced from high radian values at the vertical axes of 3.12(b) and 3.12(c). We can note that noise affects amplitude much more than it affects phase angles and that is a major advantage of angle modulation techniques over amplitude modulation ones in communications.

Now in the coming sections, we exploit the benefits of our BG model for

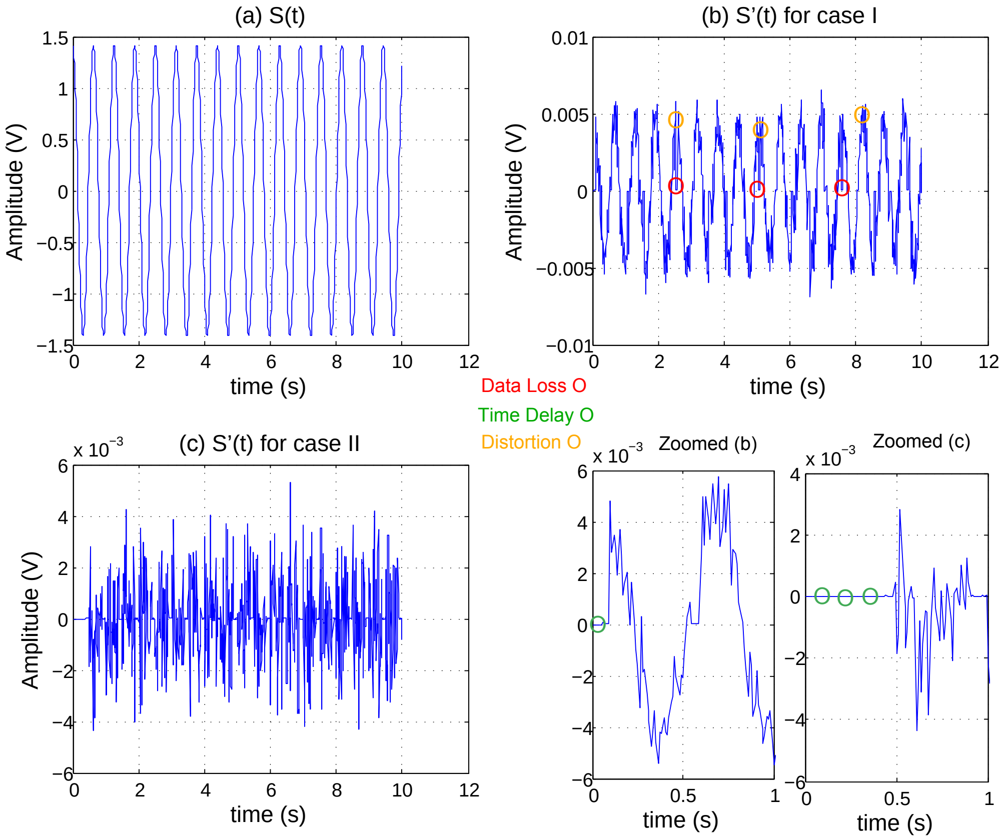


Figure 3.10 – (a) S(t) transmitted signal by $CS_{i,j}$ (b) S'(t) signal received by $CS_{i',j'}$ for case 1 (c) and case 2.

WCL at microscopic and macroscopic levels of SoS. We examine the benefits of our model at microscopic level by conducting an experiment on cooperative behavior of NAO humanoid robots in SoS context. At macroscopic level, we assess the FTL of each WCL so that we are able to compare various SoS configurations in terms of WCL robustness and reliability. We present a supervision algorithm for failed WCL detection and SoS reconfiguration for using redundant WCL instead of the failed one.

3.5 Power Measurements in Leader/Follower System of Systems

Throughout this section, we intend to exploit the benefits of our WCL model at microscopic level in SoS concept. As earlier discussed in this chap-

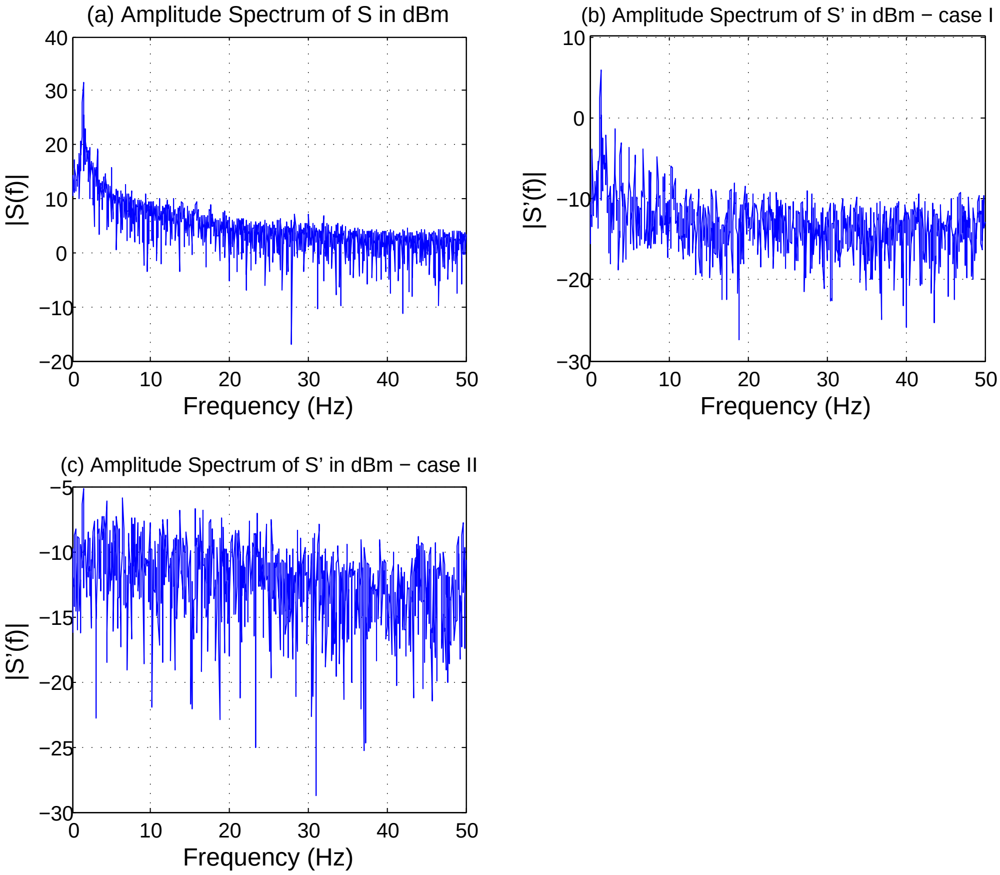


Figure 3.11 – (a) $\mid S(f) \mid$ *(b)* $\mid S'(f) \mid$ *for case 1 (c) and case 2.*

ter, we are able to detect the waveform at the receiving CS once we set the values for the model parameters, namely PTX, NL, RSS, DPL and RTT. Meaning that, at any time instant t_0 where parameter values are set/known, we can represent mathematically and graphically, the information signal received by a CS in both time and frequency domains which is extremely important if further signal processing should be done on the receiving signal in future work. But instead of manually fixing the values of those parameters, we conduct an experiment using two humanoid CSs, in the context of SoS and demonstrate how we can extract the values of those parameters by experimentation.

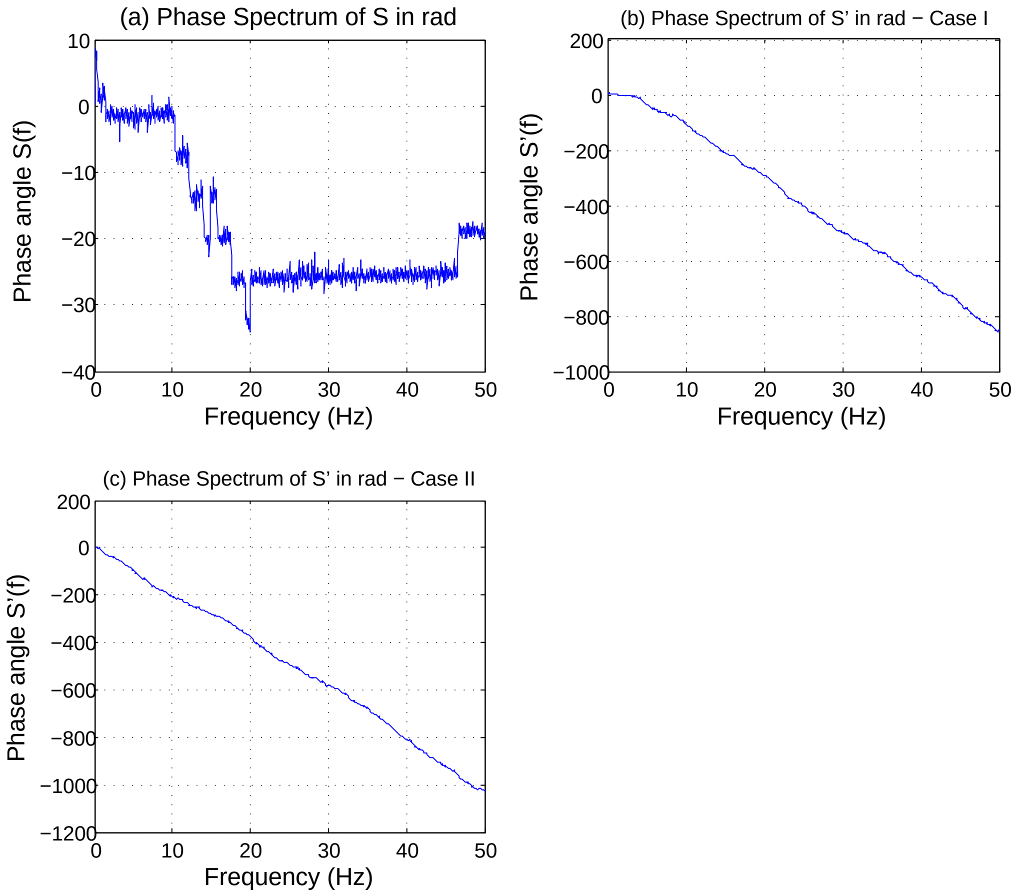


Figure 3.12 – (a) Phase angle of S(f) (b) $S'(f)$ for case 1 (c) and case 2.

3.5.1 Leader/Follower Humanoids in the Context of SoS

Consider a SoS with a global mission of transporting rods and boards across a confined space, say warehouse for example. Four NAO humanoids are used as CSs at level 0: $\text{CS}_{1,0}$, $\text{CS}_{2,0}$, $\text{CS}_{3,0}$ and $\text{CS}_{4,0}$. Each couple of humanoids consists of a leader and a follower. The leader humanoid will be responsible for detecting and avoiding obstacles, while the follower humanoid will be responsible for keeping a reasonable distance (40 to 50 cm) and orientation (max deviation of 0.2 radians) with leader. That would be necessary considering that each couple is transporting a rod or a board across a room. The coordination of each couple is represented by organizational CSs at level 1: ($\text{CS}_{1,1}$ and $\text{CS}_{2,1}$) with a sub-mission of carrying a single item. Level 2 identifies the coordination of both couples to accomplish the main mission of

SoS depicted by ($CS_{1,2}$). Figure 3.13 shows a HG hierarchal representation of the described SoS of four NAO Humanoids.

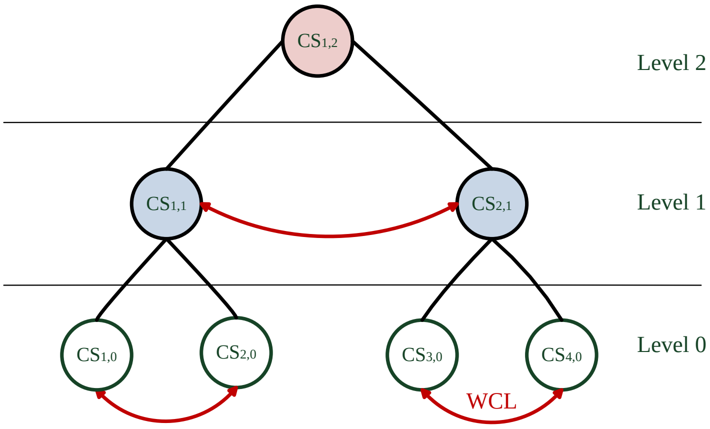


Figure 3.13 – HG hierarchal representation of the described SoS.

All SoS necessary requirements mentioned in section I.A are satisfied. Meaning that NAO CSs are operating independently and each one of them is managed as a separate entity. All NAOs are cooperating to achieve a goal that one CS cannot achieve and are spread over a geographical area with no direct power exchange. If one NAO CS fails then another one will be chosen to complete its task, which implies that NAO CSs can be added or removed without disrupting the characteristics of the resulting SoS.

3.5.2 NAO Humanoid Description

NAO is a humanoid robot, 58 cm tall and weighting 4.3 Kg, developed by the French manufacturer Aldebaran with spectacular 25 Degree of Freedom (DOF) [Aldebaran]. It includes a wide range of sensors, actuators, cameras, and interfaces. NAO image is presented in Figure 3.14. It has a Linux-based operating system. It can be programmed mainly using C++ and python. In addition, there exists a compact .Net development kit for developing Windows based applications interacting with this humanoid. NAO has been frequently used in Robocup (football competition among humanoids) due to its robustness, anti-collision systems, fast walking speeds and falling detectors.

Figure 3.14 – NAO by Aldebaran.

3.5.3 Experiment Procedure

In our experiment, we used two NAO humanoids, namely Marjo and Geoline, to do the necessary power measurements and present the cooperative behavior between two CSs to achieve a common task. We developed an interactive .NET application that has the responsibility of reading power measurements received by both humanoids and saving the results in a text-based log file for further analysis. Moreover, the application displays a detailed mission log showing synchronization messages exchanged between the two robots during mission execution. Figure 3.15 displays the experiment setup constituting of two NAOs.

The .NET application implemented has the responsibility of supplying the default behavior to each NAO. In specific, Marjo will act as the leader and Geoline as the follower. The ultrasonic sensors in each robot will be used for detecting obstacles (Marjo) and measuring distances (Geoline). Remember that the leader humanoid will be responsible for detecting and avoiding obstacles with proximity less than 50 cm, while the follower humanoid will be responsible for keeping a reasonable distance between 40 and 50 cm with leader and maximum deviation in orientation up to 0.2 radians.

Additional features are also available in the application such as displaying video signals from NAO cameras, opening and closing hands, detecting landmarks (useful for location information within indoor environment), display battery and connection status. The application is robust and uses multi-threading for processing tasks of each NAO independently. Conse-

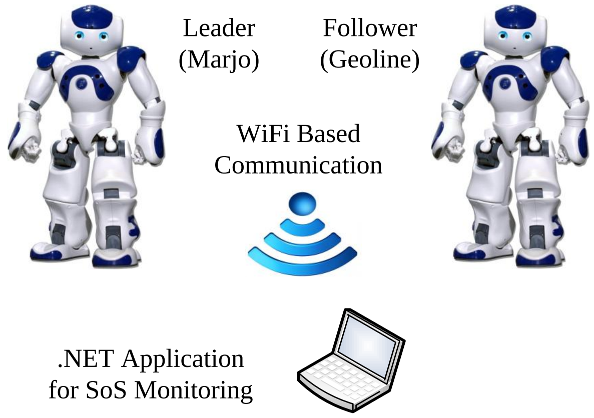
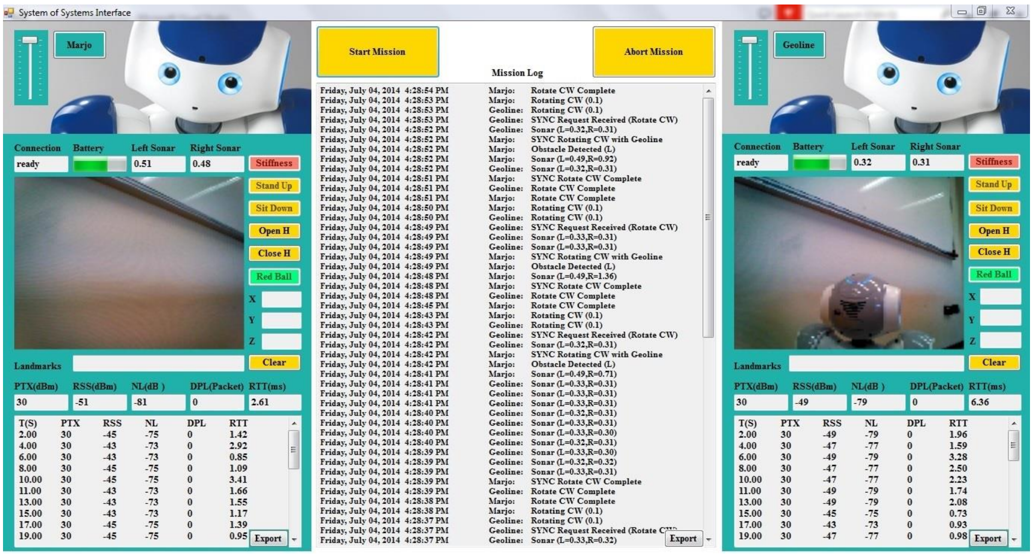


Figure 3.15 – Experiment Setup.

quently, power measurements and cooperative behavior are done in independent threads for optimizing response of humanoid robot. Figure 3.16 displays a snapshot of the .NET application used.



Marjo

Connection	Battery	Left Sonar	Right Sonar
ready		0.51	0.48

X
Y
Z
Landmarks
Clear

PTX(dBm)	RSS(dBm)	NL(dB)	DPL(Packet)	RTT(ms)
30	-51	-81	0	2.61

T(S)	PTX	RSS	NL	DPL	RTT
2.00	30	-45	-75	0	1.42
4.00	30	-43	-73	0	2.92
6.00	30	-43	-73	0	0.85
8.00	30	-45	-75	0	1.09
10.00	30	-45	-75	0	3.41
11.00	30	-43	-73	0	1.66
13.00	30	-43	-73	0	1.55
15.00	30	-43	-73	0	1.17
17.00	30	-45	-75	0	1.39
19.00	30	-45	-75	0	0.95

Mission Log

```
Friday, July 04, 2014  4:28:54 PM     Marjo:     Rotate CW Complete
Friday, July 04, 2014  4:28:53 PM     Marjo:     Rotating CW (0.1)
Friday, July 04, 2014  4:28:53 PM     Geoline:   Rotating CW (0.1)
Friday, July 04, 2014  4:28:53 PM     Geoline:   SYNC Request Received (Rotate CW)
Friday, July 04, 2014  4:28:52 PM     Geoline:   Sonar (L=0.32,R=0.31)
Friday, July 04, 2014  4:28:52 PM     Marjo:     SYNC Rotating CW with Geoline
Friday, July 04, 2014  4:28:52 PM     Marjo:     Obstacle Detected (L)
Friday, July 04, 2014  4:28:52 PM     Marjo:     Sonar (L=0.49,R=0.92)
Friday, July 04, 2014  4:28:52 PM     Geoline:   Sonar (L=0.32,R=0.31)
Friday, July 04, 2014  4:28:51 PM     Marjo:     SYNC Rotate CW Complete
Friday, July 04, 2014  4:28:51 PM     Geoline:   Rotate CW Complete
Friday, July 04, 2014  4:28:51 PM     Marjo:     Rotate CW Complete
Friday, July 04, 2014  4:28:50 PM     Marjo:     Rotating CW (0.1)
Friday, July 04, 2014  4:28:50 PM     Geoline:   Rotating CW (0.1)
Friday, July 04, 2014  4:28:49 PM     Geoline:   SYNC Request Received (Rotate CW)
Friday, July 04, 2014  4:28:49 PM     Geoline:   Sonar (L=0.33,R=0.31)
Friday, July 04, 2014  4:28:49 PM     Geoline:   Sonar (L=0.33,R=0.31)
Friday, July 04, 2014  4:28:49 PM     Marjo:     SYNC Rotating CW with Geoline
Friday, July 04, 2014  4:28:49 PM     Marjo:     Obstacle Detected (L)
Friday, July 04, 2014  4:28:48 PM     Marjo:     Sonar (L=0.49,R=1.36)
Friday, July 04, 2014  4:28:48 PM     Marjo:     SYNC Rotate CW Complete
Friday, July 04, 2014  4:28:48 PM     Geoline:   Rotate CW Complete
Friday, July 04, 2014  4:28:45 PM     Marjo:     Rotate CW Complete
Friday, July 04, 2014  4:28:43 PM     Marjo:     Rotating CW (0.1)
Friday, July 04, 2014  4:28:43 PM     Geoline:   Rotating CW (0.1)
Friday, July 04, 2014  4:28:42 PM     Geoline:   SYNC Request Received (Rotate CW)
Friday, July 04, 2014  4:28:42 PM     Geoline:   Sonar (L=0.32,R=0.31)
Friday, July 04, 2014  4:28:42 PM     Marjo:     SYNC Rotating CW with Geoline
Friday, July 04, 2014  4:28:42 PM     Marjo:     Obstacle Detected (L)
Friday, July 04, 2014  4:28:41 PM     Marjo:     Sonar (L=0.49,R=0.71)
Friday, July 04, 2014  4:28:41 PM     Geoline:   Sonar (L=0.33,R=0.31)
Friday, July 04, 2014  4:28:41 PM     Geoline:   Sonar (L=0.33,R=0.31)
Friday, July 04, 2014  4:28:41 PM     Geoline:   Sonar (L=0.33,R=0.31)
Friday, July 04, 2014  4:28:40 PM     Geoline:   Sonar (L=0.32,R=0.31)
Friday, July 04, 2014  4:28:40 PM     Geoline:   Sonar (L=0.33,R=0.31)
Friday, July 04, 2014  4:28:40 PM     Geoline:   Sonar (L=0.33,R=0.30)
Friday, July 04, 2014  4:28:40 PM     Geoline:   Sonar (L=0.32,R=0.31)
Friday, July 04, 2014  4:28:39 PM     Geoline:   Sonar (L=0.33,R=0.30)
Friday, July 04, 2014  4:28:39 PM     Geoline:   Sonar (L=0.32,R=0.32)
Friday, July 04, 2014  4:28:39 PM     Geoline:   Sonar (L=0.33,R=0.31)
Friday, July 04, 2014  4:28:39 PM     Marjo:     SYNC Rotate CW Complete
Friday, July 04, 2014  4:28:39 PM     Geoline:   Rotate CW Complete
Friday, July 04, 2014  4:28:38 PM     Marjo:     Rotate CW Complete
Friday, July 04, 2014  4:28:38 PM     Marjo:     Rotating CW (0.1)
Friday, July 04, 2014  4:28:37 PM     Geoline:   Rotating CW (0.1)
Friday, July 04, 2014  4:28:37 PM     Geoline:   SYNC Request Received (Rotate C
Friday, July 04, 2014  4:28:37 PM     Geoline:   Sonar (L=0.33,R=0.32)
```

Geoline

Connection	Battery	Left Sonar	Right Sonar
ready		0.32	0.31

X
Y
Z
Landmarks

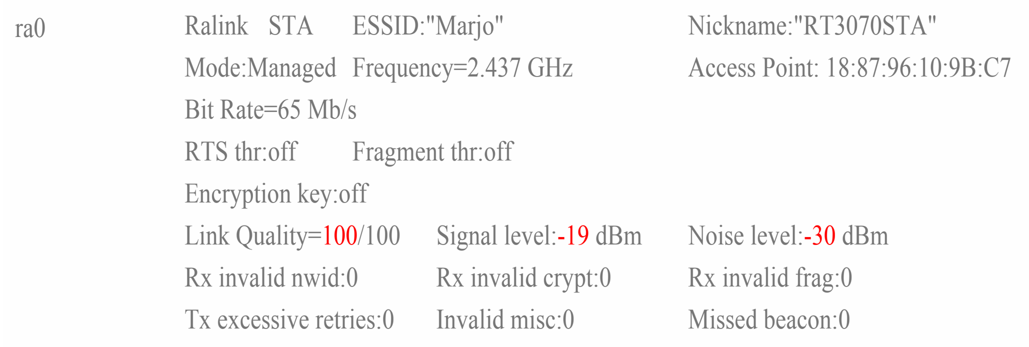
PTX(dBm)	RSS(dBm)	NL(dB)	DPL(Packet)	RTT(ms)
30	-49	-79	0	6.36

T(S)	PTX	RSS	NL	DPL	RTT
2.00	30	-49	-79	0	1.96
4.00	30	-47	-77	0	1.59
6.00	30	-49	-79	0	3.28
8.00	30	-47	-77	0	2.50
10.00	30	-47	-77	0	2.23
11.00	30	-49	-79	0	1.74
13.00	30	-49	-79	0	2.08
15.00	30	-45	-75	0	0.73
17.00	30	-43	-73	0	0.93
19.00	30	-47	-77	0	0.98

Figure 3.16 – .NET application for SoS Supervision.

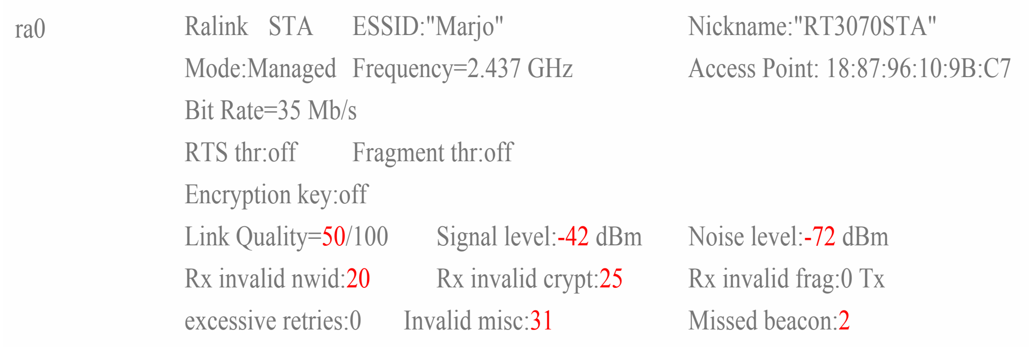
To get power measurements on Linux-based operating system in NAO, you may use the *iw* command followed by the name of the interface as an argument, for example *wlan0* in our case. We are interested mainly in RSS, NL and PTX. To calculate DPL per session, you may add received invalid fragments, transmitted excessive retries and invalid miscellaneous fragments. To get RTT for traveling packets over the wireless link, we issue a ping command followed by the number of ping packets to be sent (for example: *ping -c 3 192.168.1.1*). The result would be the RTT for each ping packet and the average RTT. We have used secure shell library in our .NET application so that we can execute shell commands and get the corresponding output on Microsoft Windows platform.

A sample output after executing *iw* command on a linux based operating system is shown in Figure 3.17 in which the link quality is maximum with no data loss. Whereas executing the same command in a noisy environment would give less link quality and more noise level. Lost packets will start to show up as missed beacons, invalid miscellaneous packets and invalid cryptography packets (Figure 3.18).



```
ra0         Ralink   STA     ESSID:"Marjo"             Nickname:"RT3070STA"
            Mode:Managed  Frequency=2.437 GHz          Access Point: 18:87:96:10:9B:C7
            Bit Rate=65 Mb/s
            RTS thr:off        Fragment thr:off
            Encryption key:off
            Link Quality=100/100    Signal level:-19 dBm    Noise level:-30 dBm
            Rx invalid nwid:0       Rx invalid crypt:0      Rx invalid frag:0
            Tx excessive retries:0  Invalid misc:0          Missed beacon:0
```

Figure 3.17 – Result after executing iw *command on NAO in a noise free environment.*



```
ra0         Ralink   STA     ESSID:"Marjo"             Nickname:"RT3070STA"
            Mode:Managed  Frequency=2.437 GHz          Access Point: 18:87:96:10:9B:C7
            Bit Rate=35 Mb/s
            RTS thr:off        Fragment thr:off
            Encryption key:off
            Link Quality=50/100     Signal level:-42 dBm    Noise level:-72 dBm
            Rx invalid nwid:20      Rx invalid crypt:25     Rx invalid frag:0 Tx
            excessive retries:0   Invalid misc:31           Missed beacon:2
```

Figure 3.18 – Result after executing iw *command on NAO in a noisy environment.*

3.5.4 Experiment Results

After successful execution of experiment procedure, we created text-based log files including power measurements received by each NAO and RTT intervals over 100 seconds. It is good to note here that the experiment has been divided into three time intervals as shown in Table 3.2:

Table 3.2 – Experiment Time Intervals.

Interval	Range (Seconds)	Environment
T_1	From 1 till 30	Non Congested - Slightly Noisy
T_2	From 31 till 60	Non Congested - Remarkably Noisy
T_3	From 61 till 100	Congested - Remarkably Noisy

Figure 3.19 shows variations over run time, related to leader humanoid Marjo, in values of four parameters defined in our BG model of WCL at leader humanoid Marjo: (a) RTT, (b) RSS, (c) NL and (d) DPL. The value of the fifth parameter: PTX is fixed during experiment run time at +30 dBm. Figure 3.20 shows parameter values related to follower humanoid Geoline.

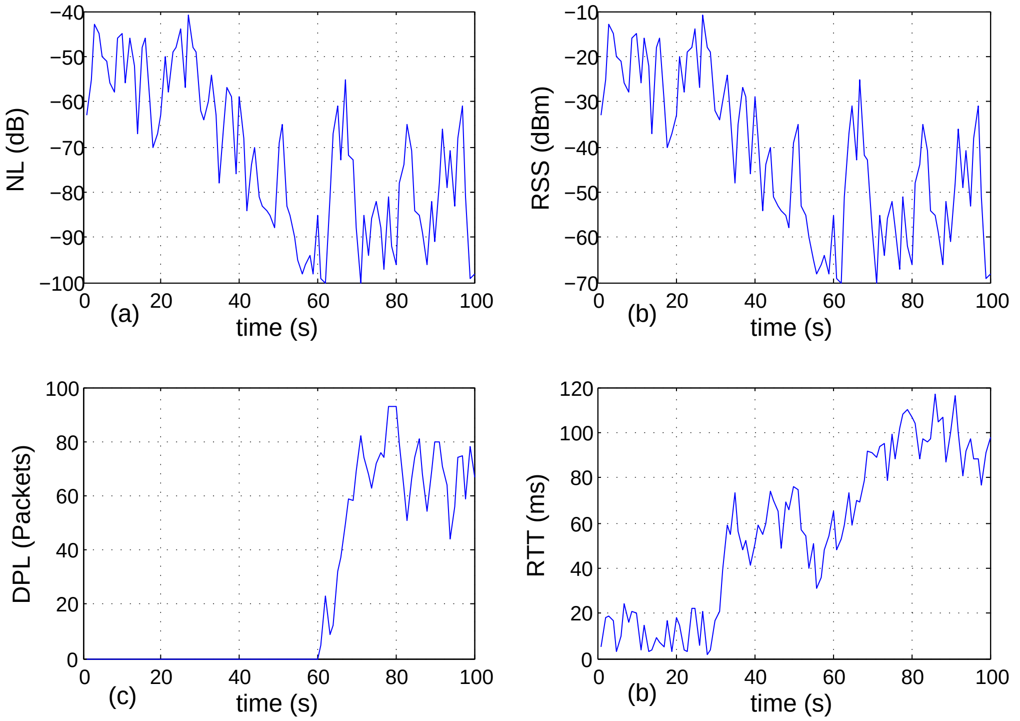


Figure 3.19 – Marjo parameters: (a) RSS (b) NL (c) DPL (d) Average RTT.

In both Figures, 3.19 and 3.20, we note the following:

1. Subplots (a) and (b) exploit the fact that RSS(dBm) = PTX(dBm) + NL(dB). For that reason, subplot of NL is the same as that of RSS but shifted -30 dBm vertically.

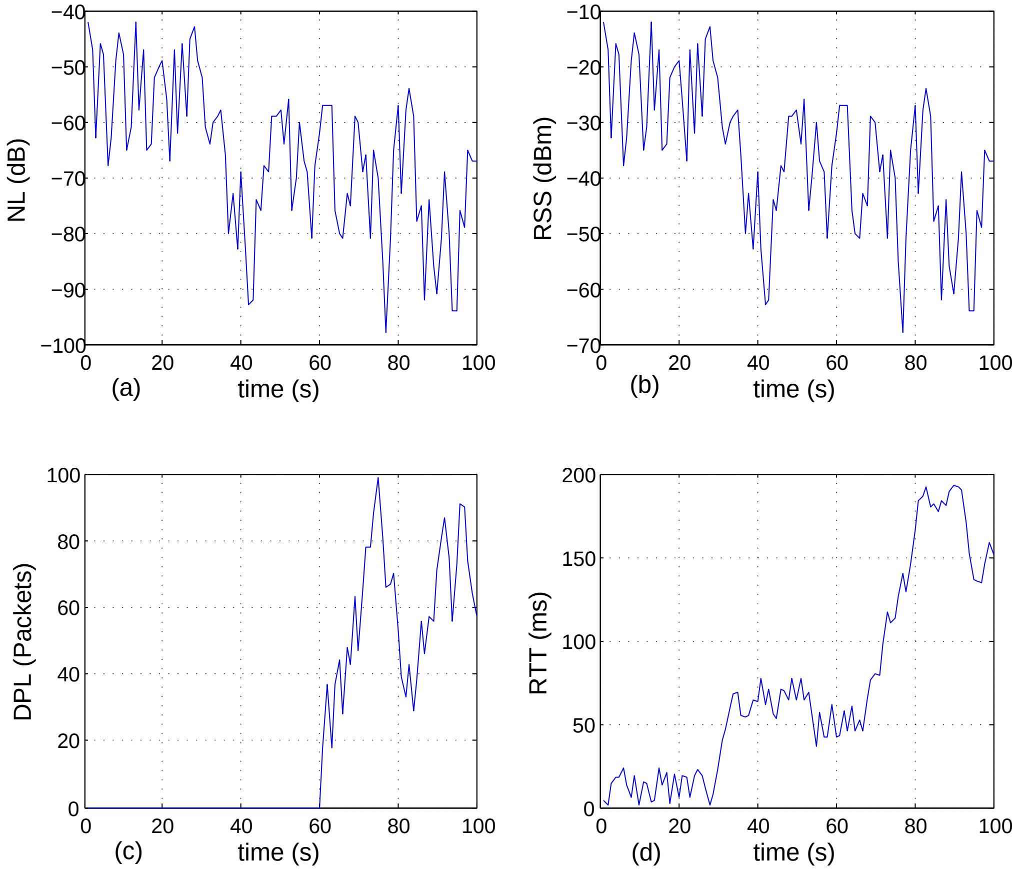


Figure 3.20 – Geoline parameters: (a) RSS (b) NL (c) DPL (d) Average RTT.

2. RSS drops remarkably from interval T_1 to T_2 and T_3 simultaneously due to increase in NL.
3. DPL is absent when no congestion exists (T_1 and T_2) and starts increasing when congestion occurs in T_3.
4. RTT values increase rapidly when congestion occurs.

As a conclusion, we note that RSS might fall beyond acceptable thresholds (less than -90 dBm) under the existence of congestion and noisy environment. In addition, DPL will start also to increase, causing upper layer applications to ask for retransmission and consequently will lead to heavy congestion and finally the communication link will fail.

Next, we exploit the benefit our WCL model at macroscopic level by assessing the FTL of WCL in SoS, which helps us figure out the robustness of a SoS, communication wise, and introduce the concept of redundant communication links.

3.6 Fault Tolerance of a Wireless Communication Link

3.6.1 Fault Tolerance Terminology

The term Fault Tolerance (FT) has been extensively used in engineering systems over the past couple of decades [Johnson 84] [Laprie 85]. In fact, FT is an essential requirement for reliability of systems. A FT system is one that can resume operation properly, but with degraded quality rather than failing completely, in the event of failure of one or few of its components. FT can be achieved by anticipating faulty situations and designing the system to handle them [Dubrova 13] [Menychtas 12].

3.6.2 Fault Tolerance in the Notion of System of Systems

SoS are meant to be highly reliable and available systems. due to emergent behavior, CSs are continuously configured and their respective missions are relatively updated to cope with the SoS mission requirements. In [Khalil 12], using HG, its possible to detect local or global faults in SoS after applying a supervision strategy from ascending direction. The reconfiguration of SoS is possible to overcome the faulty situation when the available systems satisfies a maximum number of constraints. In this case, the faulty CS is dropped from the SoS. In [Kumar 14], the developed behavioral model is simulated for normal and faulty scenarios. A fault detection and isolation algorithm based on BG approach, developed in [Ould-Boumama 03] has been developed to cope with faulty elements within the CS itself.

Our job in the context of this thesis, having developed a BG model for WCL in the notion of SoS is to study its FT. Meaning that, we need to deploy a mechanism for assessing FTL of WCL. This would allow us to analyze the robustness of a SoS configuration on a communication link basis. If a WCL is not reliable, then we have to reconfigure our SoS to use a redundant WCL. In the coming section, we discuss the need for redundant communication links in SoS. Later we define our mechanism to assess FTL.

3.6.3 Necessity of Redundant Wireless Communication Links

A direct WCL is one where information is routed from one CS, say $CS_{i,j}$, to another CS, say $CS_{i',j'}$, with no intermediary nodes. Whereas information could have been routed from $CS_{i,j}$ to $CS_{i'',j''}$, for example, and then from $CS_{i'',j''}$ to $CS_{i',j'}$. The latter is called a composite WCL between $CS_{i,j}$ and $CS_{i',j'}$, redundant to the direct one.

The need for redundant WCL arises hugely in wireless mobile communications since transmitted information is subject to shadowing and multi-path fading. Under shadowing, $CS_{1,0}$ is unable to receive information from $CS_{2,0}$, for example, due to the existence of a relatively large obstacle near $CS_{1,0}$. Figure 3.21 depicts usage of redundant WCL in case of shadowing.

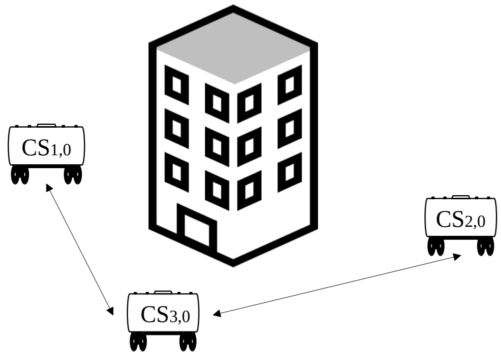


Figure 3.21 – Usage of redundant WCL in case of shadowing.

Multi-path fading suggests that the transmitted signal will propagate from $CS_{2,0}$ to $CS_{1,0}$, for example, with multiple paths due to the existence of several obstacles between them. At $CS_{1,0}$, signals with same information from multiple paths will lead to either a constructive interference or destructive one. Figure 3.22 shows $CS_{1,0}$ receiving multiple versions of the same signal transmitted by $CS_{2,0}$.

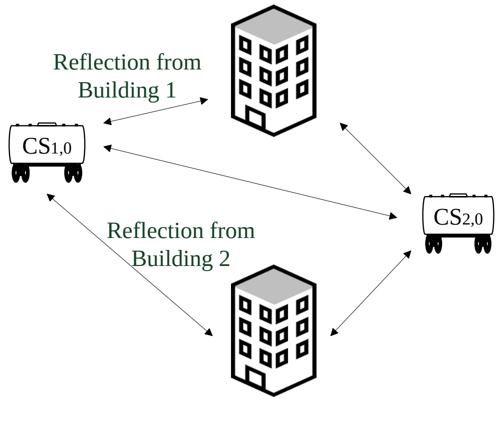


Figure 3.22 – $CS_{1,0}$ receiving multiple versions of signal transmitted by $CS_{2,0}$.

3.6.4 Quantitative Analysis on Fault Tolerance Level

Key Points

In our work, we will define two types of WCL among CSs of SoS: direct and composite. A direct link between $CS_{1,0}$ and $CS_{2,0}$, for example, means that data will be transmitted straight from $CS_{1,0}$ to $CS_{2,0}$ without passing through other intermediary component systems. Whereas a composite link between $CS_{1,0}$ and $CS_{2,0}$ involves routing the data through at least one intermediary CS, say $CS_{3,0}$ for example. Composite links are never cyclic (no CS repeated twice) and the maximum number of hops is limited by the number of CSs in SoS. Eventually, a composite link will be defined as a chain of m direct links with m being greater than two. For example, in SoS with five CSs, there exists one direct link between $CS_{1,0}$ and $CS_{2,0}$ and 15 composite links defined as follows: 1-4-2, 1-4-3-2, 1-3-2, 1-3-4-2, 1-5-2, 1-5-4-2, 1-4-5-2, 1-4-5-3-2, 1-5-3-4-2, 1-4-3-5-2, 1-5-4-3-2, 1-3-5-2, 1-3-4-5-2, 1-5-3-2, 1-3-5-4-2. A failure in any direct link composing a composite link will result in the failure of the whole composite link.

First, we will define a square matrix for direct links DL[n][n] where n is the number of CSs in SoS. Each entry DL_{ij} denotes the state of the direct link between $CS_{i,0}$ and $CS_{j,0}$. Value 1 will denote a valid WCL whereas value 0 will denote an invalid (failed) WCL. Now to determine whether a WCL is valid or not, we will define three thresholds for minimum acceptable RSS and maximum acceptable DPL and RTT values. Whenever any of the measured parameters falls beyond acceptable range, then the WCL is considered invalid (failed) and its corresponding entry DL_{ij} will be reset to zero, otherwise set to one. Note that the matrix DL is not symmetrical since there might exist a valid WCL from $CS_{1,0}$ to $CS_{2,0}$ and an invalid WCL from $CS_{2,0}$ to $CS_{1,0}$. This mainly occurs due to multipath propagation of wireless signals leading to either constructive or destructive interference. Entry values DL_{ii} will be neglected since they are meaningless (in simulation, they will be set to zero).

Second, another square matrix R[n][n] will be defined. This will be the **redundancy matrix used to quantify the level of FT of the proposed SoS**. The analysis of this matrix allows for a pair of transmitter-receiver to determine all the possible paths. Thus, one can determine the minimum number of failures preventing the considered transmission. **From reliability side, this corresponds to the minimum order of interruptions on this communication link**. As different transmissions are needed for a given system, the level of FT is then the minimum value of tolerance of the required communication link denoted by R_{ij}.

Each entry R$_{ij}$ indicates the reliability value of WCL set (direct + all composite) between CSs $\mathrm{CS}_{i,0}$ and $\mathrm{CS}_{j,0}$. When this value falls beyond a predefined threshold, then the WCL is no more reliable. Note that the predefined threshold is the minimum FTL of the WCL. Each value R_{ij} represents a decimal value of redundancy of the WCL between $\mathrm{CS}_{i,0}$ and $\mathrm{CS}_{j,0}$. A unity value indicates best possible redundancy, which means that the WCL has many redundancies in case the direct link fails. A null value indicates that the direct link and all composite links are invalid and consequently data cannot be routed between CSs. Calculating entry values R_{ij} will be explained thoroughly afterwards.

How to Estimate R_{ij} Values?

1. we start by setting the WCL threshold values defined by minimum RSS, maximum DPL and maximum RTT.

2. Measure parameter values for PTX, RSS, NL, DPL and RTT corresponding to WCL between $\mathrm{CS}_{i,0}$ and $\mathrm{CS}_{j,0}$ by experimentation such as that done in SoS of NAOs stated earlier in this chapter.

3. For the direct WCL between $\mathrm{CS}_{i,0}$ and $\mathrm{CS}_{j,0}$ to be valid, the parameter values must fall within acceptable thresholds chosen in step 1. At the end of this step, we are able to determine all direct link matrix values. Recall that DL_{ij} will be set to one whenever the direct WCL is valid, otherwise reset to zero.

4. The next step would be generating all possible composite links for the WCL between $\mathrm{CS}_{i,0}$ and $\mathrm{CS}_{j,0}$ in SoS. This resolves to a critical problem which is finding all possible non-cyclic paths among all nodes of a bidirectional un-weighted graph. The latter is a well-defined issue in networking. Solving this problem allows optimizing all routes in a network. Note that the number of possible routes increases exponentially with the number of nodes in the graph (number of CSs in SoS).

5. The number of hops is equal to the number of direct WCLs constituting the composite link. The reliability value is considered to be the reciprocal of the number of hops. Meaning that if we have 4 hops in a composite link, then its reliability value will be set to 0.25. This indicates that as the number of direct links in a composite link increases, then there will be a larger chance for this link to fail, and accordingly

its reliability decreases.

6. If entry DL_{ij} is zero, this means that direct WCL between $\mathrm{CS}_{i,0}$ and $\mathrm{CS}_{j,0}$ is invalid. Hence, all composite links having the direct link $\mathrm{CS}_{i,0}$ - $\mathrm{CS}_{j,0}$ as one of the chain links will admit an availability of zero.

7. The redundancy value R_{ij} is equal to the sum of reliability values for available composite links divided by the sum of reliability values for all composite links.

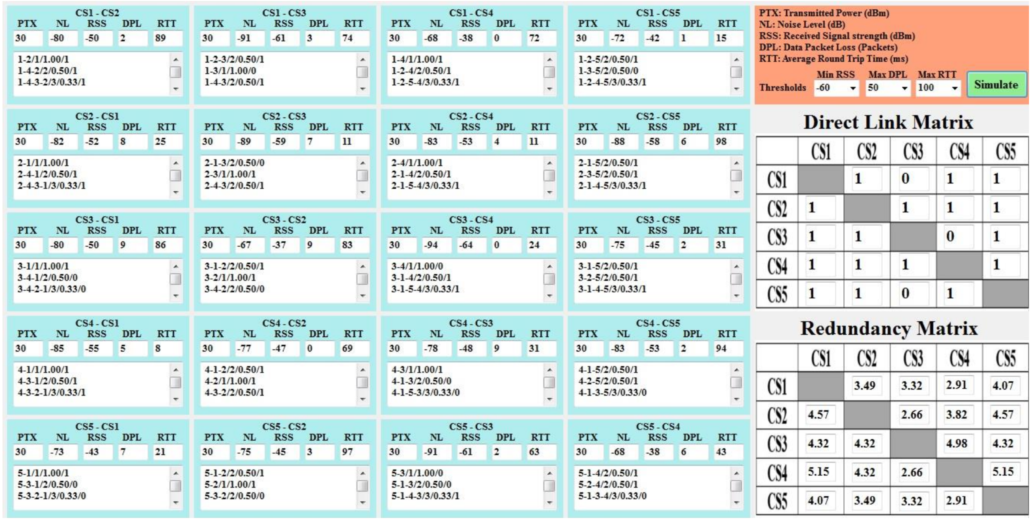
Simulator for Evaluating Fault Tolerance

In our work, we demonstrate through simulation, based on measured data, how we can evaluate the fault tolerance level of a WCL in SoS using the direct link matrix DL and the redundancy matrix R. For simplifying matters, we considered SoS with five physical component systems $\mathrm{CS}_{1,0}$, $\mathrm{CS}_{2,0}$, $\mathrm{CS}_{3,0}$, $\mathrm{CS}_{4,0}$ and $\mathrm{CS}_{5,0}$. We start the application by choosing acceptable threshold values for RSS, DPL and RTT and then we click simulate button. It generates WCL parameters for each WCL in SoS, and for each WCL, the parameters are compared with the defined thresholds, and Direct Link matrix entries are set accordingly. We find all possible composite links for all direct WCLs, and figure out the number of hops, the reliability value and the availability of all composite links based on entry values DL_{ij} in direct link matrix. Finally, we calculate redundancy matrix entries as explained earlier in this section. Figure 3.23 shows a snapshot of implemented simulator for assessing fault tolerance level of WCL in SoS.

3.7 Conclusion

Throughout this chapter, we have implemented a generic BG model of a WCL between two CSs in SoS taking into consideration linear and non-linear channel effects. We deduced from BG, the structural and behavioral equations governing the model. We were able to define the waveform of the received signal analytically and graphically. We conducted an experiment on NAO humanoids in a leader/follower scenario and demonstrated how we are able to extract parameter values experimentally. Then we introduced the notion of FT of WCL to be able to analyze its reliability.

We presented an algorithm on how to analyze quantitatively the FTL and discussed the need of redundant WCL in SoS. We implemented a simulator



CS1 - CS2

PTX	NL	RSS	DPL	RTT
30	-80	-50	2	89

1-2/1/1.00/1
1-4-2/2/0.50/1
1-4-3-2/3/0.33/1

CS1 - CS3

PTX	NL	RSS	DPL	RTT
30	-91	-61	3	74

1-2-3/2/0.50/1
1-3/1/1.00/0
1-4-3/2/0.50/1

CS1 - CS4

PTX	NL	RSS	DPL	RTT
30	-68	-38	0	72

1-4/1/1.00/1
1-2-4/2/0.50/1
1-2-5-4/3/0.33/1

CS1 - CS5

PTX	NL	RSS	DPL	RTT
30	-72	-42	1	15

1-2-5/2/0.50/1
1-3-5/2/0.50/0
1-2-4-5/3/0.33/1

CS2 - CS1

PTX	NL	RSS	DPL	RTT
30	-82	-52	8	25

2-1/1/1.00/1
2-4-1/2/0.50/1
2-4-3-1/3/0.33/1

CS2 - CS3

PTX	NL	RSS	DPL	RTT
30	-89	-59	7	11

2-1-3/2/0.50/0
2-3/1/1.00/1
2-4-3/2/0.50/1

CS2 - CS4

PTX	NL	RSS	DPL	RTT
30	-83	-53	4	11

2-4/1/1.00/1
2-1-4/2/0.50/1
2-1-5-4/3/0.33/1

CS2 - CS5

PTX	NL	RSS	DPL	RTT
30	-88	-58	6	98

2-1-5/2/0.50/1
2-3-5/2/0.50/1
2-1-4-5/3/0.33/1

CS3 - CS1

PTX	NL	RSS	DPL	RTT
30	-80	-50	9	86

3-1/1/1.00/1
3-4-1/2/0.50/0
3-4-2-1/3/0.33/0

CS3 - CS2

PTX	NL	RSS	DPL	RTT
30	-67	-37	9	83

3-1-2/2/0.50/1
3-2/1/1.00/1
3-4-2/2/0.50/0

CS3 - CS4

PTX	NL	RSS	DPL	RTT
30	-94	-64	0	24

3-4/1/1.00/0
3-1-4/2/0.50/1
3-1-5-4/3/0.33/1

CS3 - CS5

PTX	NL	RSS	DPL	RTT
30	-75	-45	2	31

3-1-5/2/0.50/1
3-2-5/2/0.50/1
3-1-4-5/3/0.33/1

CS4 - CS1

PTX	NL	RSS	DPL	RTT
30	-85	-55	5	8

4-1/1/1.00/1
4-3-1/2/0.50/1
4-3-2-1/3/0.33/1

CS4 - CS2

PTX	NL	RSS	DPL	RTT
30	-77	-47	0	69

4-1-2/2/0.50/1
4-2/1/1.00/1
4-3-2/2/0.50/1

CS4 - CS3

PTX	NL	RSS	DPL	RTT
30	-78	-48	9	31

4-3/1/1.00/1
4-1-3/2/0.50/0
4-1-5-3/3/0.33/0

CS4 - CS5

PTX	NL	RSS	DPL	RTT
30	-83	-53	2	94

4-1-5/2/0.50/1
4-2-5/2/0.50/1
4-1-3-5/3/0.33/0

CS5 - CS1

PTX	NL	RSS	DPL	RTT
30	-73	-43	7	21

5-1/1/1.00/1
5-3-1/2/0.50/0
5-3-2-1/3/0.33/0

CS5 - CS2

PTX	NL	RSS	DPL	RTT
30	-75	-45	3	97

5-1-2/2/0.50/1
5-2/1/1.00/1
5-3-2/2/0.50/0

CS5 - CS3

PTX	NL	RSS	DPL	RTT
30	-91	-61	2	63

5-3/1/1.00/0
5-1-3/2/0.50/0
5-1-4-3/3/0.33/1

CS5 - CS4

PTX	NL	RSS	DPL	RTT
30	-68	-38	6	43

5-1-4/2/0.50/1
5-2-4/2/0.50/1
5-1-3-4/3/0.33/0

PTX: Transmitted Power (dBm)
NL: Noise Level (dB)
RSS: Received Signal strength (dBm)
DPL: Data Packet Loss (Packets)
RTT: Average Round Trip Time (ms)

	Min RSS	Max DPL	Max RTT
Thresholds	-60	50	100

Direct Link Matrix

	CS1	CS2	CS3	CS4	CS5
CS1		1	0	1	1
CS2	1		1	1	1
CS3	1	1		0	1
CS4	1	1	1		1
CS5	1	1	0	1	

Redundancy Matrix

	CS1	CS2	CS3	CS4	CS5
CS1		3.49	3.32	2.91	4.07
CS2	4.57		2.66	3.82	4.57
CS3	4.32	4.32		4.98	4.32
CS4	5.15	4.32	2.66		5.15
CS5	4.07	3.49	3.32	2.91	

Figure 3.23 – Snapshot of implemented simulator.

that demonstrates how we are able to calculate redundancy matrix entries corresponding to FTL in a SoS composed of five CSs.

In the next chapter, we introduce a methodology for coupling behavioral and organizational models developed using BG and HG respectively in a unified multilevel SoS model. Our BG model of a WCL will be used in the unified multilevel coupled model to denote exchange of information among CSs.

Chapter 4

Methodology of Modeling System of Systems Engineering

Contents

4.1 Introduction

In this chapter, a methodology is developed to describe the both properties of SoS, namely organizational and behavioral using a unified model approach. Our main contribution, after providing the BG model of a WCL in a SoS concept, is to establish a methodology for coupling HG and BG in SoS modeling. Thus, the whole model can be used for either simulation or supervision purposes, where it describes the dynamic behaviors of the elementary CSs and their communication, with the management of missions at organizational levels. The proposed methodology includes seven steps listed as follows:

1. Model the set-based representation of SoS using HG.
2. Generate the valued graph corresponding to set-based representation of SoS.
3. Generate the dual graph of the resulting valued graph.
4. Use BG to model physical CSs in the resulting dual graph.
5. Use BG to model WCL among physical CSs in the resulting dual graph.
6. Use HBG to model the management of missions attributed to CSs.
7. Check the fundamental organizational properties of the SoS.

4.1.1 System of Systems Properties Depicted by Hyper Graph

Before we proceed with our methodology, using HG multi-level hierarchal modeling, we demonstrate analytically the five main properties of a SoS: operational independence, managerial independence, geographic dispersion, emergent behavior and evolutionary and adaptive development of SoS.

Operational independence of Component Systems

A physical CS $CS_{i,j=0}$ in a SoS is operationally independent if and only if it belongs to a vertex which represents that it is operationally independent. It can process its mission $M_{i,j=0}$ using its own allocated resources:

$$CS_{i,j=0} \in V = \{CS_{i',j'}\}_{i' \in N^*, j'=0} \tag{4.1}$$

Moreover, an organizational CS $CS_{i,j\neq 0}$ in a SoS is operationally independent if and only if it belongs to a hyperedge which represents that it is operationally independent. It can process its mission $M_{i,j\neq 0}$ using its own allocated resources:

$$CS_{i,j\neq 0} \in \xi = \{CS_{i',j'}\}_{i'\in N^*,j'\in N^*} \tag{4.2}$$

Managerial independence of Component Systems

Physical CS $CS_{i,j=0}$ and $CS_{i',j'=0}$ in a SoS are managerially independent if and only if they can manage their missions independent to each other:

$$\forall CS_{i,j=0}, CS_{i',j'=0} \in V : M_{i,j=0} \bigcap M_{i',j'=0} = \emptyset \tag{4.3}$$

Similarly, organizational CS $CS_{i,j\neq 0}$ and $CS_{i',j'\neq 0}$ in a SoS are managerially independent if and only if they can manage their missions independent to each other:

$$\forall CS_{i,j\neq 0}, CS_{i',j'\neq 0} \in \xi : M_{i,j\neq 0} \bigcap M_{i',j'\neq 0} = \emptyset \tag{4.4}$$

Geographical dispersion of Component Systems

Physical CS $CS_{i,j=0}$ and $CS_{i',j'=0}$ in a SoS are dispersed geographically if and only if they are not connected physically to each other but they can exchange information. In BG we use detectors of effort, De, or detectors of flow, Df, to denote exchange of information:

$$\forall CS_{i,j=0}, CS_{i',j'=0} \in V : CS_{i,j=0} \bigcap CS_{i',j'=0} = \emptyset \tag{4.5}$$

Similarly, organizational CSs $CS_{i,j\neq 0}$ and $CS_{i',j'\neq 0}$ in a SoS are dispersed geographically if and only if they are not connected physically to each other but they can exchange information:

$$\forall CS_{i,j\neq 0}, CS_{i',j'\neq 0} \in \xi : CS_{i,j\neq 0} \bigcap CS_{i',j'\neq 0} = \emptyset \tag{4.6}$$

Emergent behavior

Physical CSs $CS_{i,j=0}$ in a SoS cooperate with each other to accomplish the mission of a higher CS since no CS alone can achieve the SoS mission:

$$\forall CS_{i,j=0} \in V : \bigcup(M_{i,j=0}) \rightarrow M_{i',j'\neq 0} \tag{4.7}$$

Similarly, organizational CSs $CS_{i,j\neq 0}$ in a SoS cooperate with each other to accomplish the mission of a higher CS:

$$\forall CS_{i,j\neq 0} \in \xi : \bigcup(M_{i,j\neq 0}) \rightarrow M_{i',j'\neq 0} \tag{4.8}$$

Evolutionary and adaptive development

A physical CS $CS_{i,j=0}$ in a SoS undergoes evolutionary and adaptive development if and only if the SoS structure is never fully formed or complete, meaning that CSs can change their organization to achieve a new mission set by SoS:

$$\forall CS_{i,j=0} : (V = V \setminus \{CS_{i,j=0}\}) \vee (V = V \bigcup \{CS_{i,j=0}\}) \tag{4.9}$$

Similarly, organizational CS $CS_{i,j=0}$ in a SoS undergoes evolutionary and adaptive development if and only if they change their organization to achieve a new mission set by SoS:

$$\forall CS_{i,j\neq 0} : (\xi = \xi \setminus \{CS_{i,j\neq 0}\}) \vee (\xi = \xi \bigcup \{CS_{i,j\neq 0}\}) \tag{4.10}$$

4.2 Graphical Modeling in System of Systems

Here we consider a generalized case for a SoS composed of multi-robot systems cooperating to achieve a common task. This multi-robot system can be considered as a SoS, since robots are dispersed geographically with no physical interaction but can exchange information among them. In addition, Each mobile robot is operationally and managerially independent. Furthermore, robots missions are continuously configured/updated for evolutionary development of the SoS. In Figure 4.1, a multi-robot SoS is shown, where a four mobile omnidirectional robots, called 'Robotino', are cooperating to achieve a common mission.

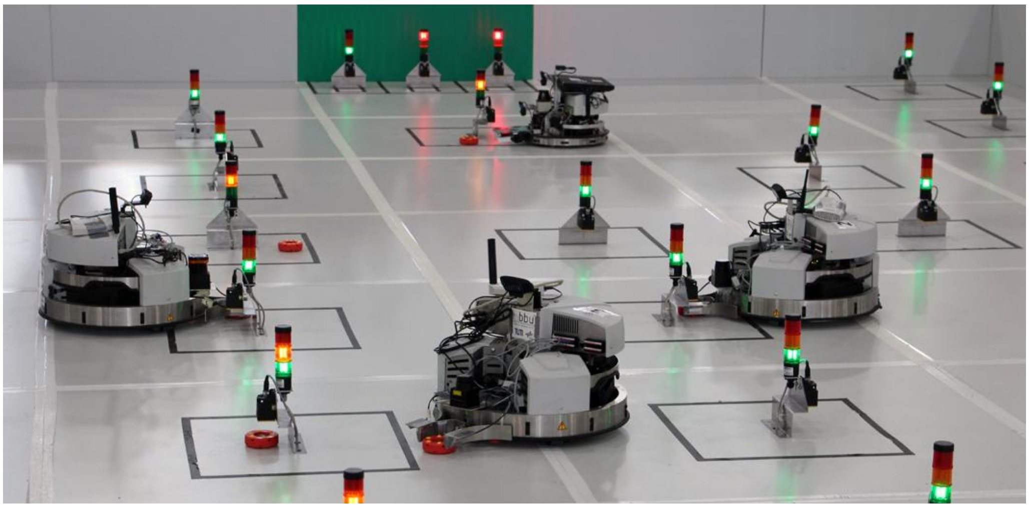


Figure 4.1 – A multi-robot SoS.

4.2.1 Model Set-Based Representation of System of Systems

The first step in our methodology is to model the set-based representation of SoS using HG. We note eight steps to model the set-based representation of CSs in SoS:

1. Identify all physical CSs, $CS_{i,0}$ in SoS. Each $CS_{i,0}$ will be represented by a vertex.
2. Identify the organization of every CS, $CS_{i,j\neq0}$ in SoS. Each organization $CS_{i,j\neq0}$ will be represented by a hyperedge.
3. Physical CSs, $CS_{i,0}$, belonging to same organization, $CS_{i,j\neq0}$, will be denoted by vertices inside the hyperedge of their organization.
4. Exchange of information (WCL) between physical CSs, $CS_{i,0}$ and $CS_{i',0}$, belonging to same organization, $CS_{i,j\neq0}$, is represented by a two sided arrow joining vertices corresponding to physical CSs $CS_{i,0}$ and $CS_{i',0}$.
5. Exchange of information (WCL) between organizational CSs, $CS_{i,j\neq0}$ and $CS_{i',j'\neq0}$, is also represented by a two sided arrow joining hyperedges corresponding to organizational CSs $CS_{i,j\neq0}$ and $CS_{i',j'\neq0}$.
6. Organizational CSs, $CS_{i,j\neq0}$, that are part of other bigger Organizational CSs, $CS_{i',j'/j'>j}$, are represented by a hyperedge, $CS_{i,j\neq0}$, nested in the hyperedge of the bigger organizational CS $CS_{i',j'/j'>j}$.

7. SoS is represented by a hyperedge containing all other hyperedges corresponding to organizational CSs.
8. Mission attributed to $CS_{i,j}$ will be denoted by function $M_{i,j}$.

Fig, 4.2 depicts a sample set-based representation of a SoS with seven vertices and five hyperedges.

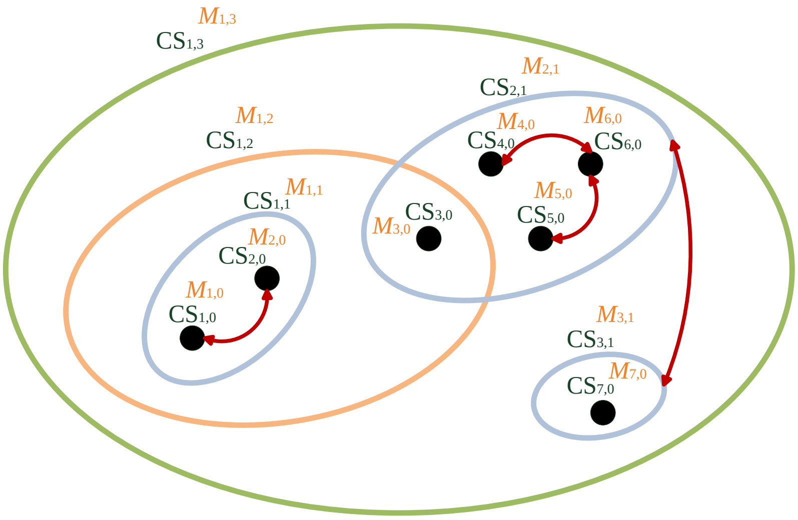


Figure 4.2 – Set-based representation of SoS using HG.

Note that the the red arrows indicate the exchange of information between physical CSs or organizational CSs. This set-based representation in Figure 4.2 can be written mathematically in the form:

$$HG = \{\{\{CS_{1,0}, CS_{2,0}\}, CS_{3,0}\}, \{CS_{3,0}, CS_{4,0}, CS_{5,0}, CS_{6,0}\}, \{CS_{7,0}\}\}$$

4.2.2 Generate The Valued Graph Corresponding to Set-Based Representation

In [Khalil 12], a definition has been established to represent a HG(V,ξ) as a valued graph $G^v\{V^v, \xi^v, \digamma\}$ such that:

1. Vertices V^v of G^v represent the hyperedges ξ of HG:

$$V^v = \{CS_{i,j}\}_{i \in N, j \in N^*} \tag{4.11}$$

2. Two vertices $CS_{i,j}$ and $CS_{i',j'}$ are connected in G^v if and only if the intersection between $CS_{i,j} \in \xi$ and $CS_{i',j'} \in \xi$ is not null:

$$\xi^v = \{E_{i,j}^{i',j'} = (CS_{i,j}, CS_{i',j'})/CS_{i,j} \bigcap CS_{i',j'} \neq \emptyset\} \tag{4.12}$$

3. Every edge $E_{i,j}^{i',j'} \in \xi$ of G^v is valued by the function $\mathcal{F}$ defined as follows:

$$\mathcal{F} : E_{i,j}^{i',j'} \mapsto \mathcal{F}_{i,j}^{i',j'} = CS_{i,j} \bigcap CS_{i',j'} \tag{4.13}$$

Figure 4.3. shows a set-based representation of SoS using HG(V,ξ) to the left and its corresponding valued graph $G^v\{V^v, \xi^v, \mathcal{F}\}$ to the right.

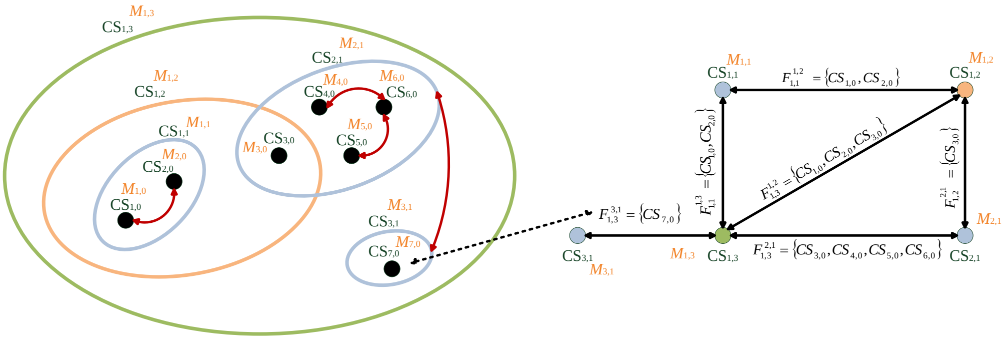

M1,3
CS1,3
M2,1
CS2,1
M1,2
CS1,2
M4,0
CS4,0
M6,0
CS6,0
M1,1
CS1,1
M5,0
CS5,0
CS3,0
M3,0
M2,0
CS2,0
M1,0
CS1,0
M3,1
CS3,1
M7,0
CS7,0
M1,1
CS1,1
$F_{1,1}^{1,2} = \{CS_{1,0}, CS_{2,0}\}$
M1,2
CS1,2
$F_{1,1}^{1,3} = \{CS_{1,0}, CS_{2,0}\}$
$F_{1,3}^{1,2} = \{CS_{1,0}, CS_{2,0}, CS_{3,0}\}$
$F_{1,2}^{2,1} = \{CS_{3,0}\}$
$F_{1,3}^{3,1} = \{CS_{7,0}\}$
M2,1
CS3,1
M3,1
M1,3
CS1,3
$F_{1,3}^{2,1} = \{CS_{3,0}, CS_{4,0}, CS_{5,0}, CS_{6,0}\}$
CS2,1

Figure 4.3 – Set-based representation using HG (left) and its corresponding valued graph (right).

Special Case: If the set-based representation of SoS is composed of one hyperedge, then performing the valued graph of this set-based representation will yield a single vertex, corresponding to the single hyperedge in set-based representation, with no edges to other vertices. In that case, the valued function $\mathcal{F}_{i,j}^{i',j'}$ is represented on a reflexive relationship from the single vertex to itself. This will ensure continuity of modeling for our methodology for the coming steps.

4.2.3 Generate The Dual Graph of the Resulting Valued Graph

For each valued graph $G^v\{V^v, \xi^v, \mathcal{F}\}$, we define its dual graph $G^*\{V^*, \xi^*\}$ such that:

$$CS_{i,0} \in \digamma^{i',j'}_{i,j} of \xi^{v} in G^{v} \Leftrightarrow CS_{i,0} \in V^{*} in G^{*} \tag{4.14}$$

$$CS_{i,j} \in V^{v} in G^{v} \Leftrightarrow CS_{i,j} \in \xi^{*} in G^{*} / \xi^{*} = \{\xi^{*}_{k} = (\forall CS_{i,0}/CS_{i,0} \in CS_{i,j})\} \tag{4.15}$$

which means that all physical CSs, $CS_{i,0}$, will be represented as vertices V_i^* in G^* and each organizational CS, $CS_{i,j\neq 0}$, will be represented as hyperedge ξ_i^* in G^* connecting all physical CSs belonging to this organizational CS. Figure 4.4 depicts the dual graph G^* of the valued graph presented in Figure 4.3.

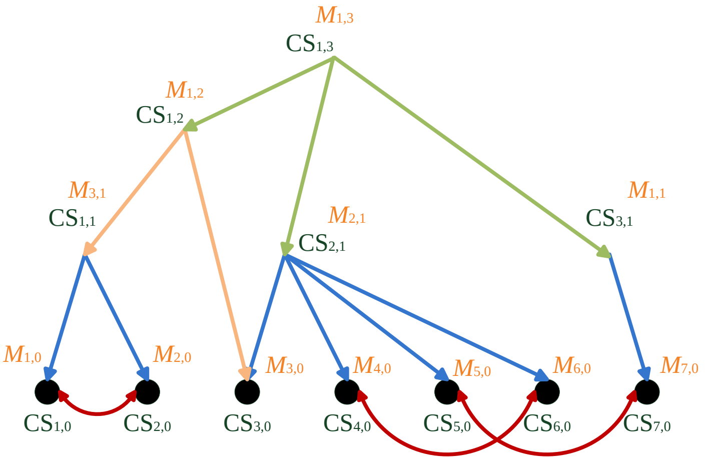


Figure 4.4 – Dual graph G^* of the valued graph presented in Figure 4.3

At this stage, the resulting dual graph G^* is the hierarchal representation of SoS. Now our next job in the coming sections is to to model, using BG, the physical CSs and the WCL among physical CSs in the resulting dual graph. Then we will use HBG to model the management of missions attributed to CSs in SoS.

4.2.4 Use BG to Model Physical CSs in The Resulting Dual Graph

Using BG to model flow of power in physical CSs depends radically on the physical system being used. In our generic example of multi-robot SoS, we consider omnidirectional robot called 'Robotino'. The behavioral model of this physical CS has been extensively developed in [Kumar 14] showing longitudinal, lateral and yaw dynamics. System state equations, junction structural equations, behavioral equations and dynamic equations have all been derived. Furthermore, the BG model is manipulated to obtain FDI model, then Analytical Redundant Relationships (ARR) were generated by following a causal path from known to unknown variables. A Fault Signature Matrix (FSM) is developed to enable detecting and isolating faults based on the obtained residuals.. The complete behavioral model of Robotino, depicted in [Kumar 14], is shown in Figure 4.5.

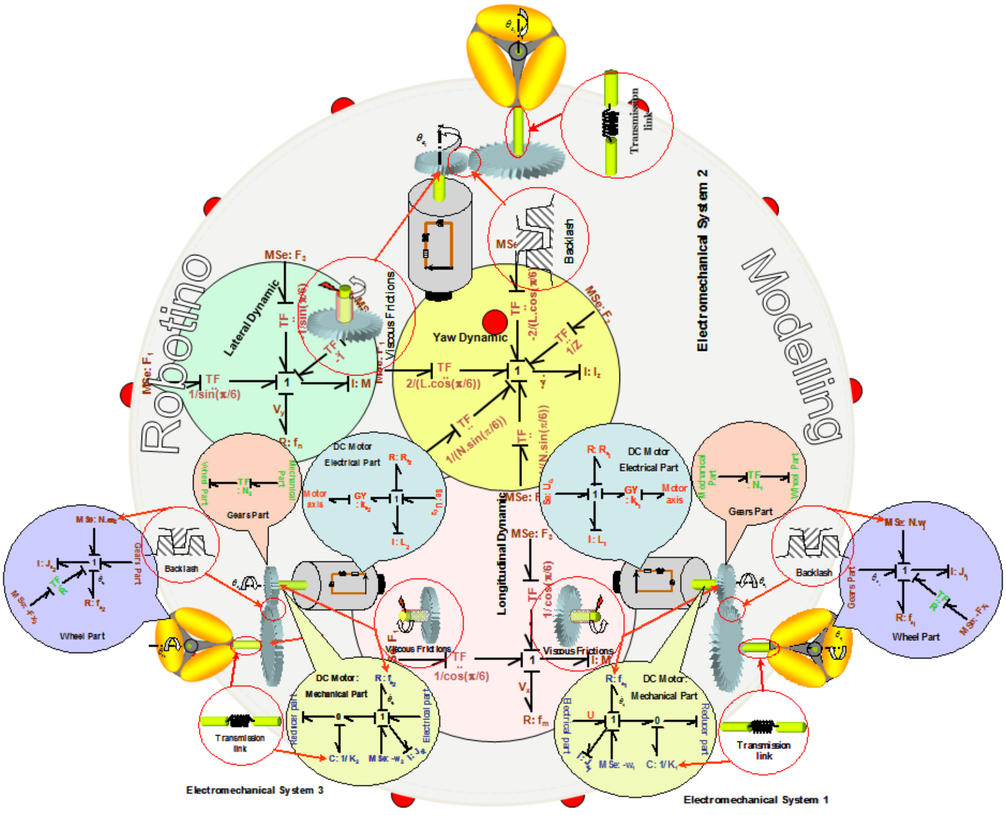


Figure 4.5 – The complete behavioral model of Robotino depicted in [KUMAR 2014].

4.2.5 Bond Graph Model of Wireless Communication Link in The Resulting Dual Graph

Now our job is to replace the bidirectional red arrows in the dual graph depicted in Figure 4.4 and representing the WCL among CSs by our BG model of WCL presented in Chapter 3. Figure 4.6 shows four CSs, namely $CS_{4,0}$, $CS_{5,0}$, $CS_{6,0}$ and $CS_{7,0}$, WCL among them and their corresponding organizational CSs.

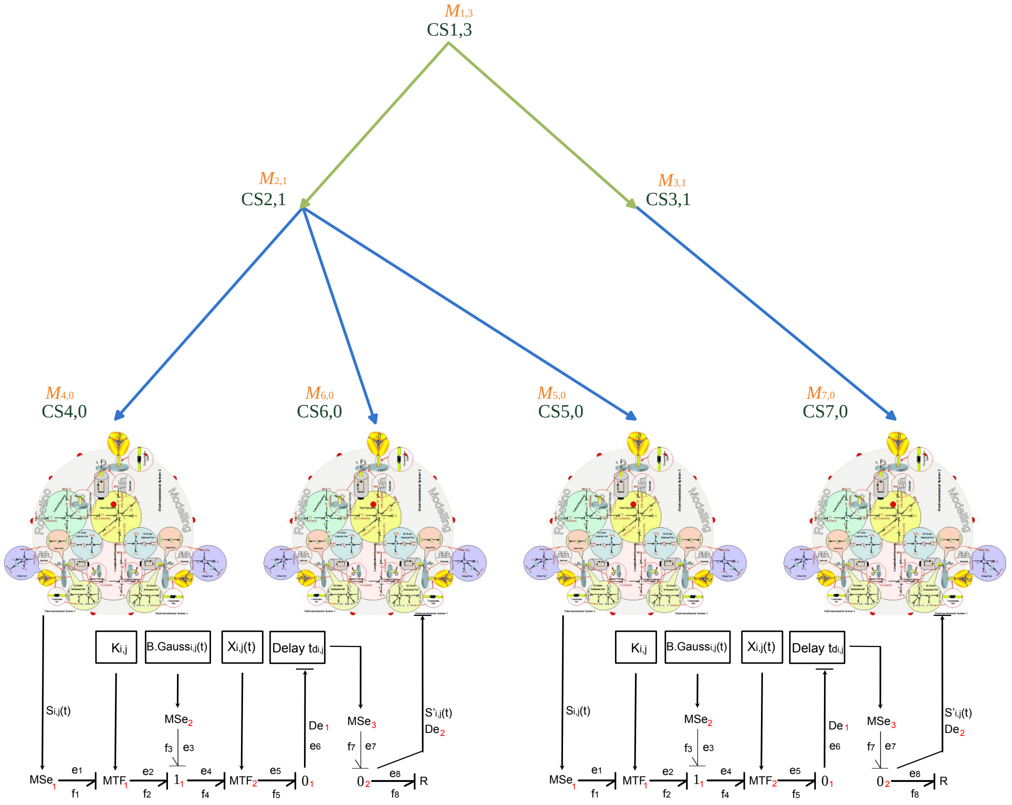


Figure 4.6 – Behavioral model of four CSs, with WCL among them and their organization.

4.2.6 Use Hybrid Bond Graph to Model The Management of Missions Attributed to Component Systems

Our next step is to model the management of missions attributed to CSs in the resulting dual graph of Figure 4.4. For that, we will use the HBG, which is ideal for modeling discontinuous behavior of hybrid systems. We begin by a notion on HBG then we discuss our modeling technique.

Notion on Hybrid Bond Graph

BG is geared towards modeling of continuous systems. There are various techniques to model the discontinuous behavior in hybrid systems. However, HBG is one of the best tools since it extends BG by incorporating controlled junctions to enable modeling of hybrid systems [Ming 10]. In discontinuous mode, 1-type junctions, denoted by 1_c, ensure a zero flow to all connected bonds when deactivated. Whereas, 0-type junctions, denoted by 0_c ensure a zero effort to all connected bonds when deactivated. When activated, both controlled junctions 1_c and 0_c function as ordinary 1 and 0 junctions respectively.

The notion of causality in HBG must be chosen carefully since when any of the controlled nodes is deactivated, this may lead to causality conflict in the connected bonds. To guarantee that we always assign the proper causality of HBG, a diagnostic HBG (DHBG) is achieved such that the causality assignment at each controlled node is correct regardless of the operating mode (Continuous or discrete).

To make things clearer, we consider a simple example of a hybrid system which is an electric circuit with a switch Sw to control the On/Off state of the circuit. The schematic of the circuit is shown in Figure 4.7.

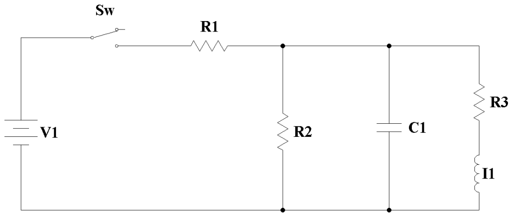


Figure 4.7 – Example on a hybrid system: electric circuit with a switch.

Using HBG, we can model the electric circuit of Figure 4.7 using two different causalities depicted in Figure 4.8(a) and Figure 4.8(b) respectively. Note that inactive bonds cause an invalid causality at junction 0_2 in Figure 4.8(a) where as inactive bonds don't pose an invalid causality on junction 0_2 in Figure 4.8(b).

Using Hybrid Bond Graph to Model The Management of Missions

Coming to multilevel SoS, we will use HBG to model the management of missions based on the fact that CSs undergo an emergent behavior in SoS, meaning that their missions dynamically change to accomplish the whole SoS mission. Consequently, controlled junctions will be used to model the interaction of physical CSs with organizational CSs such that if the controlled junction is deactivated, then this means that the physical component system participates no more in the organization.

Figure 4.9 shows the complete model of the SoS depicted in Figure 4.6 with HBG modeling the management of missions. Note that controlled 0-type switches 2, 4 and 6 are deactivated since physical CSs $CS_{4,0}$, $CS_{5,0}$ and $CS_{6,0}$ do not contribute to organizational CS $CS_{2,1}$, and 0-type switch

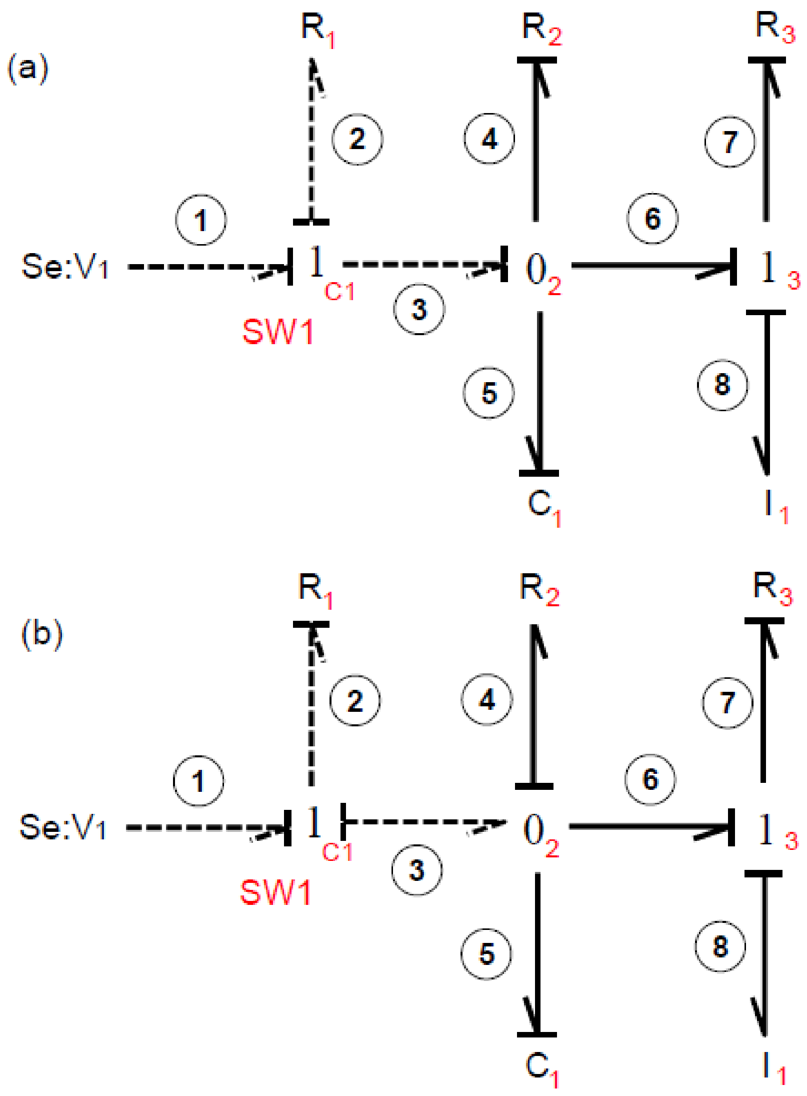


Figure 4.8 – HBG of circuit of Figure 4.8 with two different causalities [Ming 10].

7 is also deactivated since physical CS $CS_{7,0}$ does not contribute to organizational CS $CS_{3,1}$. Note that controlled switches in HBG are ordinary binary switches and not event based switches which means no need to add any further conditions on the controlled switches.

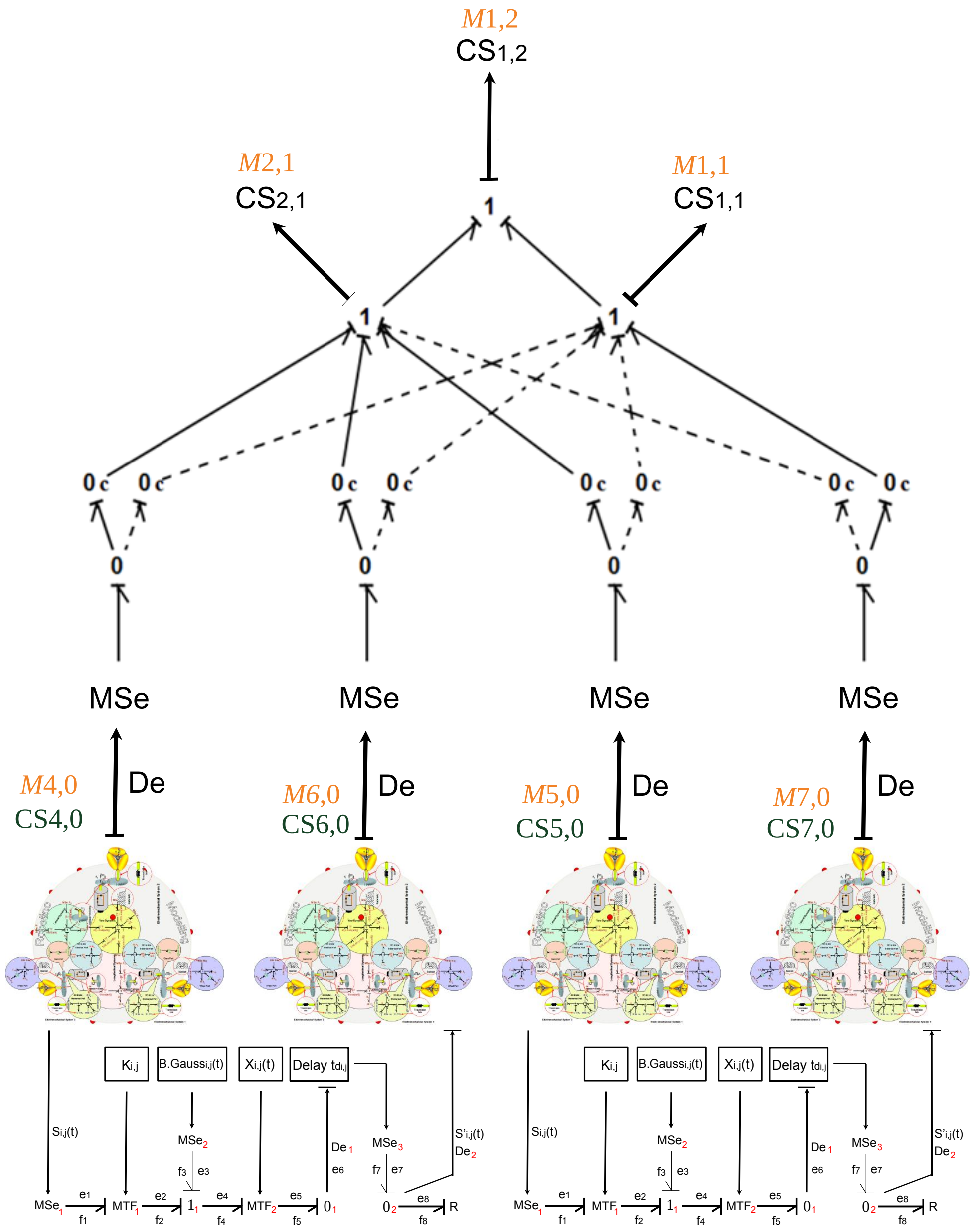


Figure 4.9 – Complete model of SoS depicted in Figure 4.6.

4.2.7 Check The Fundamental Organizational Properties of System of Systems

Now our goal is to verify that our methodology for modeling SoS conserves its fundamental organizational properties: operational independence of CSs, managerial independence of CSs, geographical dispersion of CSs, emergent behavior, and evolutionary and adaptive development. Bare in mind that our methodology is a coupled HG-BG model. It starts with an HG model of SoS and ends up with BG/HBG model of the same SoS. Accordingly, most definitions and techniques presented in [Kumar 14] on verifying SoS properties from BG can be restated here for the same objective.

Note that in out model, behavioral CSs at level 0 are modeled using BG. Which means that we will use 0-type junctions, 1-type junctions, modulated sources of effort MSe, modulated sources of flow MSf, transformers, gyrators, detectors of effort De and detectors of flow Df to build the behavioral model of CSs at level 0. Organizational CSs at level j $\neq$ 0 are modeled using HBG by means of 0-type controlled junctions, 0-type junctions, 1-type junctions, detectors of effort De and modulated sources of effort MSe. Hence, in our model, we define CSs as follows:

At level j = 0

$$CS_{i,j=0} \in V = \{0, 1, MSe, MSf, TF, GY, De, Df\} \tag{4.16}$$

At level j $\neq$ 0

$$CS_{i,j\neq 0} \in \xi = \{0_c, 0, 1, De, MSe\} \tag{4.17}$$

Operational independence of Component Systems

At level j = 0

$$CS_{i,j=0} \in V = \{CS_{i',j'}\}_{i' \in N^*, j'=0} \tag{4.18}$$

At level j $\neq$ 0

$$CS_{i,j\neq 0} \in \xi = \{CS_{i',j'}\}_{i' \in N^*, j' \in N^*} \tag{4.19}$$

Managerial independence of Component Systems

At level j = 0

$$\forall CS_{i,j=0}, CS_{i',j'=0} \in V : M_{i,j=0} \bigcap M_{i',j'=0} = \emptyset \tag{4.20}$$

At level j $\neq$ 0

$$\forall CS_{i,j\neq 0}, CS_{i',j'\neq 0} \in \xi : M_{i,j\neq 0} \bigcap M_{i',j'\neq 0} = \emptyset \tag{4.21}$$

Geographical dispersion of Component Systems

In our BG model of WCL between two CSs, we use detectors of effort De, modulated sources of effort MSe, modulated transformer MTF, 1-type junctions and 0-type junctions.

At level j = 0

$$\forall CS_{i,j=0}, CS_{i',j'=0} \in V : CS_{i,j=0} \bigcap CS_{i',j'=0} = \{De, MSe, MTF, 1, 0\} \tag{4.22}$$

We denoted the organization of CSs at level $\neq 0$ using HBG by detectors of effort De, modulated sources of effort MSe, o-type junctions, 1-type junctions and 0_c-type controlled junctions.

At level j ≠ 0

$$\forall CS_{i,j\neq 0}, CS_{i',j'\neq 0} \in \xi : CS_{i,j\neq 0} \bigcap CS_{i',j'\neq 0} = \{De, MSe, 0, 1, 0_c\} \tag{4.23}$$

Emergent behavior

At level j = 0

$$\forall CS_{i,j=0} \in V : \bigcup (M_{i,j=0}) \rightarrow M_{i',j'\neq 0} \tag{4.24}$$

At level j ≠ 0

$$\forall CS_{i,j\neq 0} \in \xi : \bigcup (M_{i,j\neq 0}) \rightarrow M_{i',j'\neq 0} \tag{4.25}$$

Evolutionary and adaptive development

At level j = 0

$$\forall CS_{i,j=0} : (V = V \setminus \{CS_{i,j=0}\}) \vee (V = V \bigcup \{CS_{i,j=0}\}) \tag{4.26}$$

At level j ≠ 0

$$\forall CS_{i,j\neq 0} : (\xi = \xi \setminus \{CS_{i,j\neq 0}\}) \vee (\xi = \xi \bigcup \{CS_{i,j\neq 0}\}) \tag{4.27}$$

4.3 Case Study: Multi-Robot Hockey Team as a System of Systems

To better explain our methodology for coupling HG and BG in SoS modeling, we study the following example of a SoS throughout the whole chapter. Consider a Hockey team (Figure 4.11) composed of six robots as follows: one goalkeeper, three defenders and two attackers. Each robot, whether a humanoid or an omnidirectional robot (Figure 4.10), is a physical CS at level zero and called also elementary CS. Accordingly, $CS_{1,0}$ is the goalkeeper, $CS_{2,0}$, $CS_{3,0}$, and $CS_{4,0}$ are the defenders and $CS_{5,0}$, $CS_{6,0}$ are the attackers. At level one, $CS_{1,1}$, $CS_{2,1}$ and $CS_{3,1}$ are organizational CSs and represent the goal keeping mission, the defensive mission and the attacking mission respectively. The whole Hockey team is represented by a master CS $CS_{1,2}$ at level two, which is in fact the SoS with a unified mission of winning the Hockey game. Defenders may communicate with each others. Attackers may also communicate with each other. The goal keeping mission may interact/communicate with the defensive mission, and the defensive mission may communicate with the attacking mission.

Each of the physical component systems $CS_{1,0}$, $CS_{2,0}$, $CS_{3,0}$, $CS_{4,0}$, $CS_{5,0}$ and $CS_{6,0}$ operate independently and can be managed as a separate entity since each robot has its own running operating system. CSs, being Hockey players, are dispersed geographically over the playground with no physical interaction and just exchanging information. No CS can win the Hockey game on his own. Accordingly, all physical CSs need to coordinate to achieve the SoS mission of winning the Hockey game. Finally, any physical CS can be substituted by another physical CS. Consequently, the SoS can be dynamically configured without disrupting the whole mission. As a summary, the whole SoS requirements, defined previously, are satisfied in our SoS example. We proceed with all the mentioned steps discussed in section 4.2 to model the Hockey team SoS using our methodology.

1. Model SoS set-based representation using steps defined in section 4.2.1 (Figure 4.11)
2. Deduce the valued graph corresponding to set-based representation of SoS (Figure 4.12)
3. Generate the dual graph of the resulting valued graph (Figure 4.13)
4. Use BG to model physical CSs in the resulting dual graph (Figure 4.14)
5. Use BG to model WCL among physical CSs in the resulting dual graph (Figure 4.15)

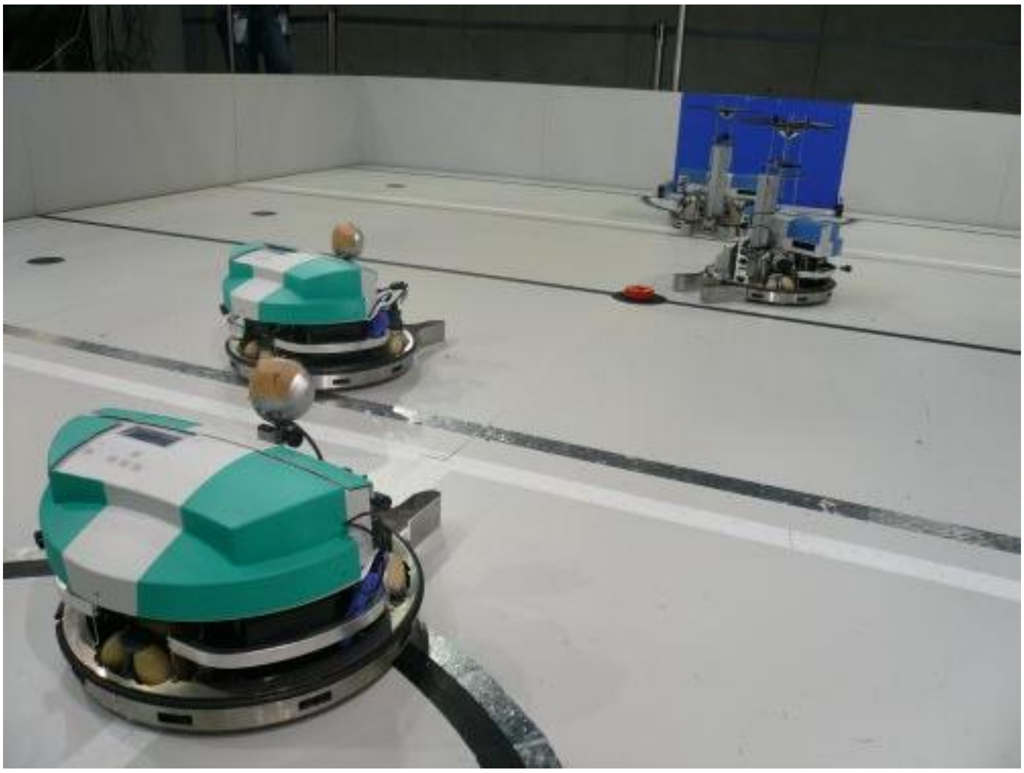


Figure 4.10 – Hockey Challenge in RoboCup.

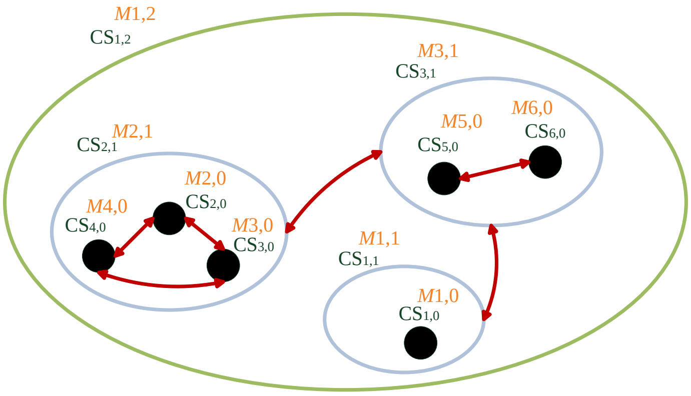


Figure 4.11 – HG set-based representation of SoS Hockey team.

6. Use HBG to model management of missions in attributed to CSs (Figure 4.16)

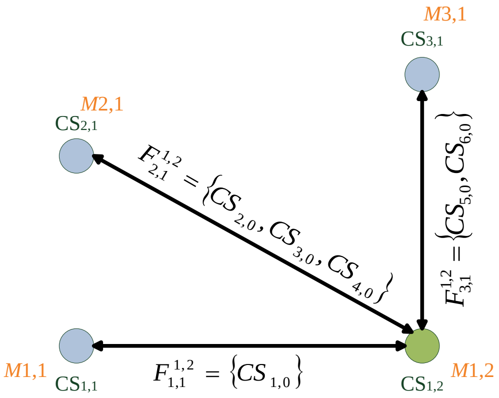

M3,1
CS3,1
M2,1
CS2,1
$F_{2,1}^{1,2} = \{CS_{2,0}, CS_{3,0}, CS_{4,0}\}$
$F_{3,1}^{1,2} = \{CS_{5,0}, CS_{6,0}\}$
M1,1
CS1,1
$F_{1,1}^{1,2} = \{CS_{1,0}\}$
CS1,2
M1,2

Figure 4.12 – Valued graph of set-based representation depicted in Figure 4.11.

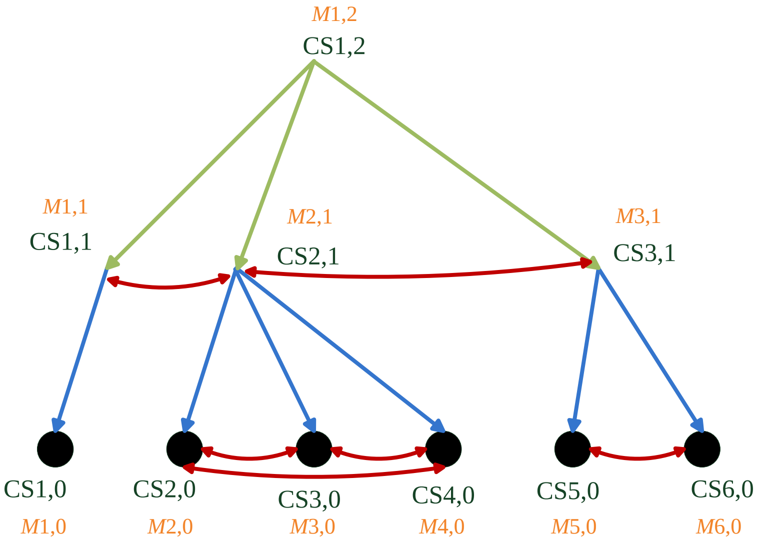


Figure 4.13 – Dual graph of valued graph representation depicted in Figure 4.12.

4.4 Conclusion

Modeling of SoS has been placed under focus extensively over the past couple of years. In [Khalil 12], an organizational model has been developed using HG, that permits modeling the organization of CSs in SoS and define supervision strategy for reconfiguring SoS upon failure of a CS. Whereas in [Kumar 14], a behavioral model has been developed using another graph-

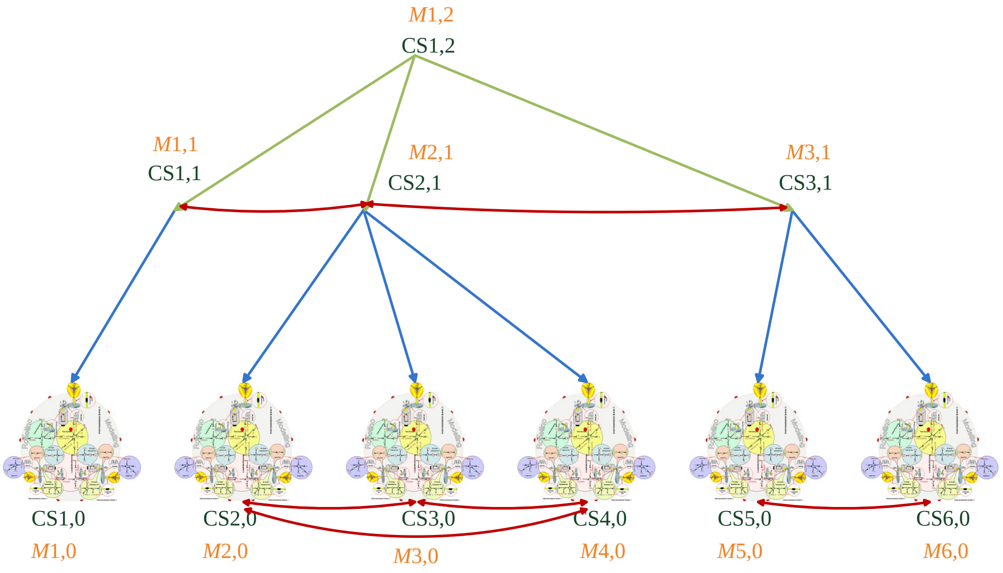


Figure 4.14 – Behavioral model of physical CSs using BG added to dual graph.

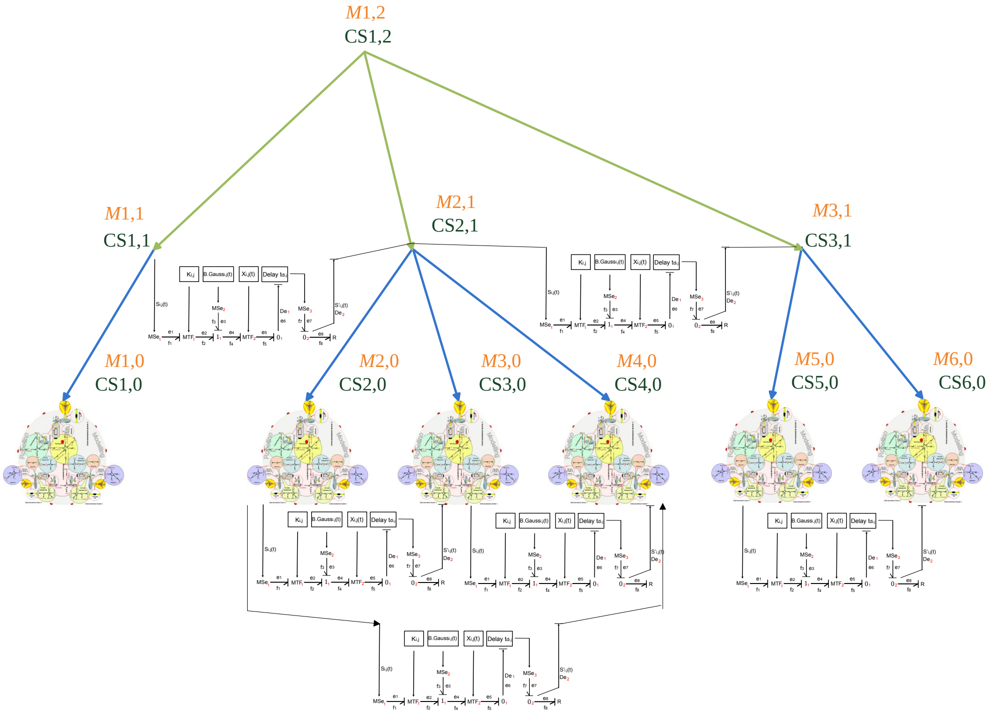


Figure 4.15 – BG model of WCL among physical CSs added to model.

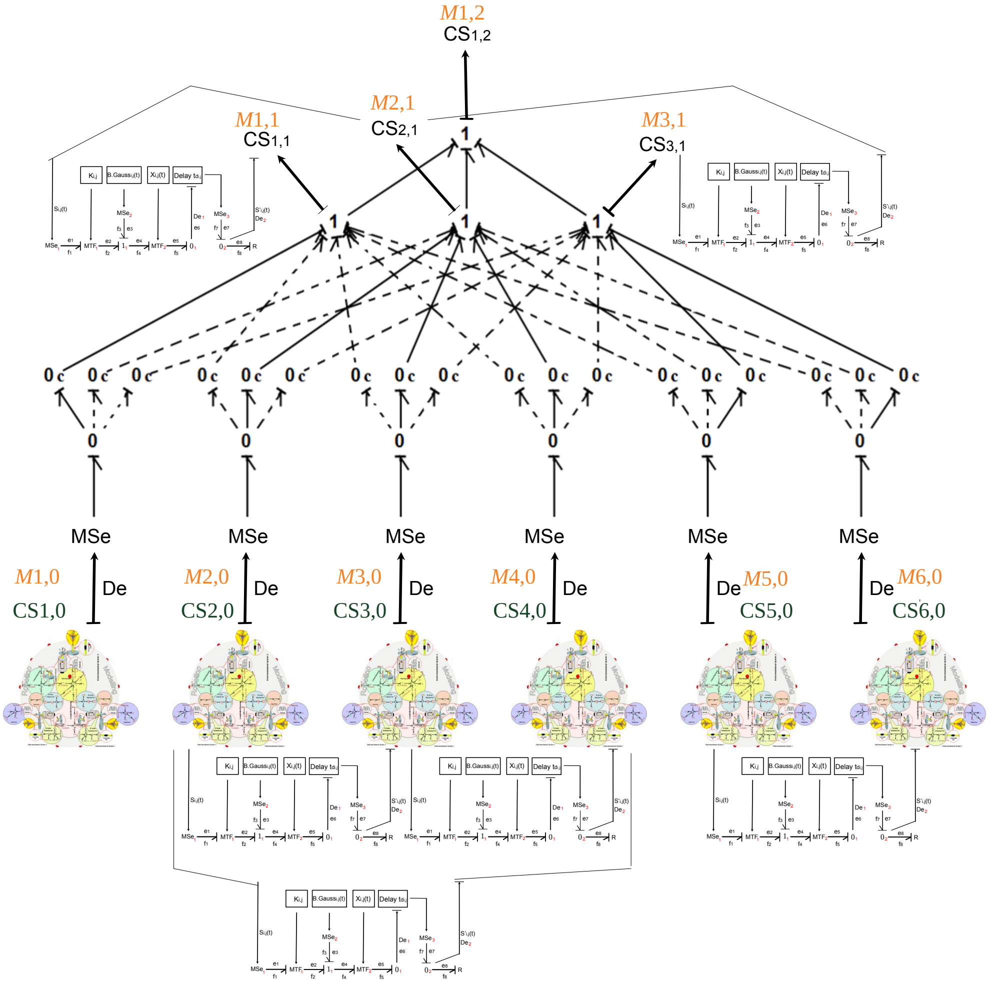


Figure 4.16 – Complete model of Hockey team SoS after adding HBG model of mission organization.

ical modeling tool BG that permits modeling transfer of power in multi-engineering domains within CS itself.

In this chapter we proposed a multilevel graphical modeling technique for SoS by coupling two graphical modeling techniques, namely HG and BG which permits modeling the organization of CSs in SoS and at the same time relate the performance of SoS to elements within CSs.

In our model, we start by SoS hierarchal representation using HG, depicted in [Khalil 12], then we deduce SoS set-based representation from hierarchal one. Then we deduce the valued graph corresponding to set-based representation of SoS and generate the dual graph of the resulting valued graph. After that, we use BG to model physical CSs in the resulting dual graph and WCL among physical CSs. Next, we use HBG to model management of missions in attributed to CSs in SoS. Finally, we verify the five SoS requirements in out proposed model. We summarize our methodology by an example on multi-robot Hockey team in the context of SoS.

In the next chapter, we consider a case study on SoS. We discuss the issue of IAV navigation by means of a UAV in the notion of SoS.

Chapter 5

Intelligent Autonomous Vehicle Navigation using Unmanned Aerial Vehicle

Contents

5.1 Introduction: Problematic in Intelligent Autonomous Vehicle Navigation

5.1.1 State of Art

IAV, or so called robotic vehicles, are becoming more and more, an important part of service robots. The main drive behind developing IAVs is to make road traffic more convenient, more safe and definitely more efficient to operate in both, wide and confined geographical areas [Duchon 12]. IAV can be defined as:

An IAV is a vehicle equipped with perception, reasoning and actuating devices that enable the automation of driving tasks such as safe lane following, obstacle avoidance, overtaking slower traffic, following the vehicle ahead, anticipating and avoiding dangerous situations, and determining the route [Siciliano 08].

IAVs have been used as transportation vehicles in ITS [Khalil 12], [Kumar 14]. ITS is the term given to road transportation mechanism when combined with information and communication technology. The term "transportation" can stand for transporting passengers or goods. Different transporting modes are depicted in Figure 5.1.

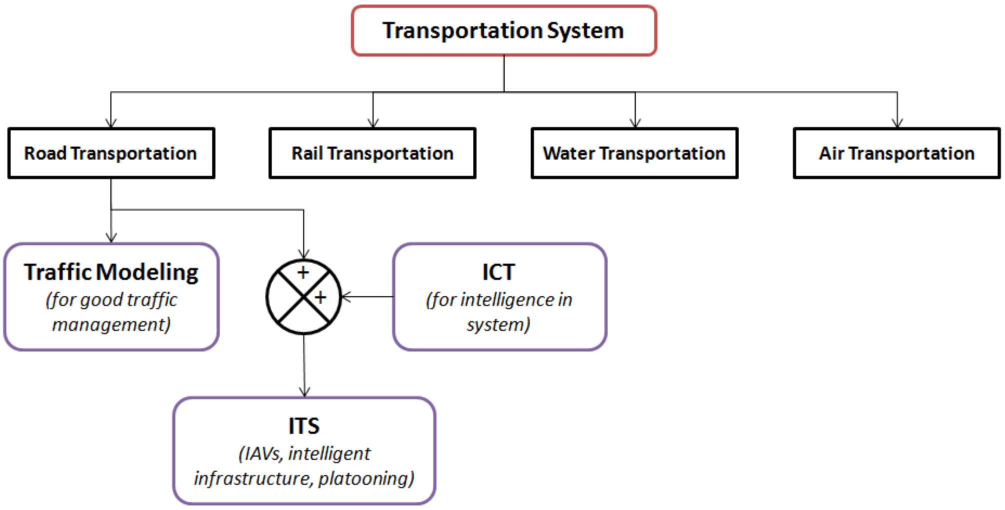


Figure 5.1 – Different transport modes of transportation system.

In literature, several studies have been conducted with the aim of tracking moving objects by means of a UAV [Pall 14], [Nitschke 14]. However, providing navigation information to moving objects by means of a UAV is an original idea proposed in this thesis.

5.1.2 Problematic

One of the main capabilities of IAVs is to roam autonomously with the help of a set of sensors, actuators, processing unit and a Global Positioning System (GPS) receiver responsible for providing navigation information (longitude, latitude and altitude). The type of GPS receiver used depends on the resolution of localization needed. If high precision is required, then an Real Time Kinematics (RTK) GPS receiver should be deployed, based on a differential solution of a base and a hover. An RTK GPS receiver can achieve resolution down to few centimeters, whereas using an ordinary Single Point Positioning (SPP) GPS receiver would give resolution in navigation information up to ten meters.

In SPP GPS, the receiver must be able to see a minimum of 4 satellites to be able to provide a solution. Whereas an RTK initialization demands that at least 5 common satellites must be tracked at base and rover units to produce a differential solution first, then after some time a float RTK solution, then a fixed RTK solution with best resolution [Bancroft 85].

In open geographic areas, the IAV equipped with a GPS receiver, whether RTK or SPP, is expected to see a large number of satellites (greater than 8) on a sunny day and accordingly can provide localization information with minimum perturbations. The problem arises in confined spaces. The IAV will be surrounded by obstacles, such as buildings, trees, or other IAVs which will disrupt the signal received from satellites and will eventually lead to either faulty localization information or even complete loss of signal [Guier 97].

5.1.3 Proposed Solution in the notion of System of Systems

In this chapter, **we intend to provide a solution for localizing IAVs in confined spaces with the help of UAV in the notion of SoS.** A UAV (Figure 5.2) is an air vehicle that can maneuver over the confined space such that no near obstacles are available to disrupt the satellites signals received by an equipped GPS receiver on the UAV itself. The localization information received by the UAV will be processed and then relayed to the IAV where a Kalman Filter is used to perform data fusion with the localization information it receives from its own GPS receiver. To conclude, **UAV will be used as an external GPS sensor with respect to IAV**.

The interaction between the robuTAINer and the UAV can be seen as a cooperative behavior between two technological CSs. Both the UAV and the robuTAINer are operationally and managerially independent. They cooperate to achieve one global mission which is safe maneuver of robuTAINer in a confined space. The robuTAINer alone cannot achieve this mission since its equipped GPS receiver can loose satellites signals at any time due to perturbations from nearby obstacles. Both CSs, the UAV and the robuTAINer are

Figure 5.2 – [Walkera] QR X800 UAV used in our project.

geographically dispersed with no physical interaction but they can exchange information. Consequently, all SoS requirements are successfully met in this system.

Modeling IAV UAV SoS using Unified Methodology

Our objective now is to model the the SoS composed of IAV and UAV in the notion of unified methodology proposed in chapter 4.

1. Model SoS set-based representation using steps defined in section 4.2.1 (Figure 4.11)

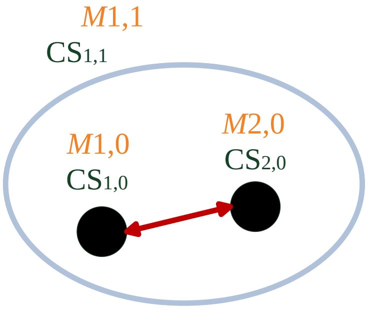


Figure 5.3 – HG set-based representation of SoS Hockey team.

2. Deduce the valued graph corresponding to set-based representation of SoS (Figure 5.4)
3. Generate the dual graph of the resulting valued graph (Figure 5.5)

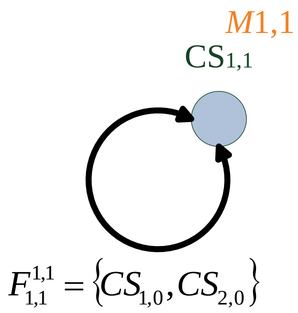


Figure 5.4 – Valued graph of set-based representation.

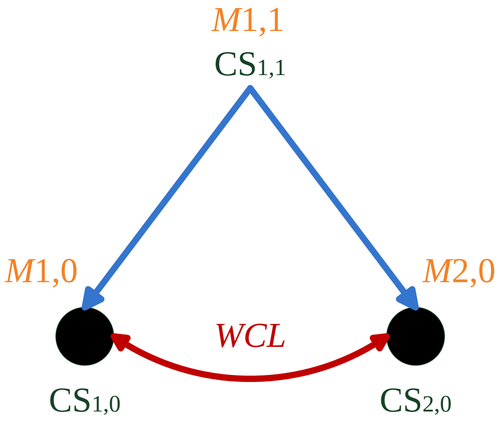


Figure 5.5 – Dual graph of valued graph representation.

4. Use BG to model physical CSs in the resulting dual graph. In our case, to reduce complexity of model, we suppress the behavioral model of UAV and IAV since this is not our main concern here. (Figure 5.6)
5. Use BG to model WCL among physical CSs in the resulting dual graph (Figure 5.7)
6. Use HBG to model management of missions in attributed to CSs (Figure 5.8)

5.2 RobuTAINer Intelligent Autonomous Vehicle

RobuTAINer is an omni-directional container carrier IAV developed by Intelligent Transportation for Dynamic Environment [InTrade] to help small

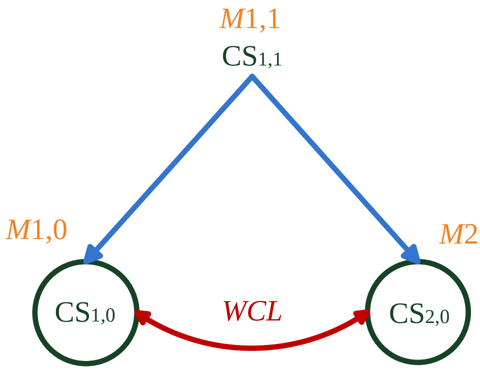


Figure 5.6 – Behavioral model of physical CSs using BG added to dual graph.

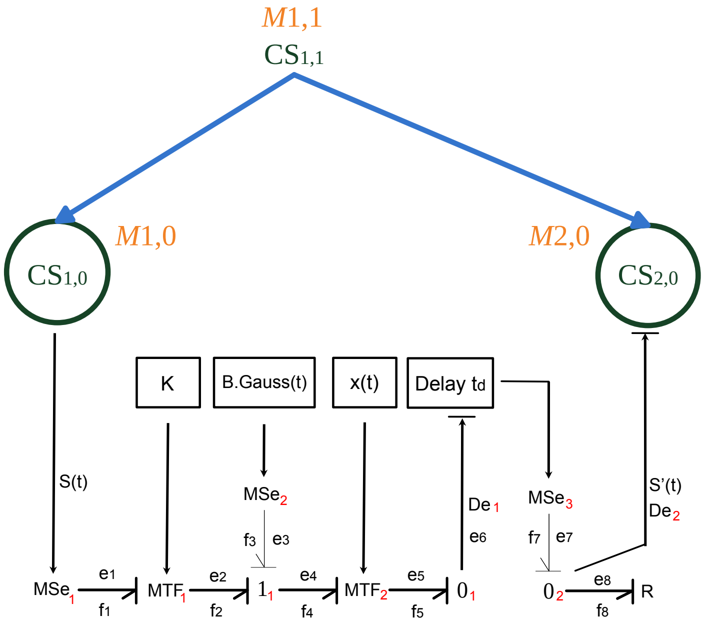


Figure 5.7 – BG model of WCL among physical CSs added to model.

to medium-sized ports to achieve gains through automated freight transport. RobuTAINeR is operationally effective and reliable in real port environments and adaptable to different port areas and configurations. It relieves congestion and provides an environmentally friendly public transport solution. RobuTAINeR IAV operational functionalities were successfully demonstrated to public at the port of Rouen in France and Dublin Ferry port in Ireland (Figure 5.9).

RobuTAINeR is controlled by means of PURE, a low level control soft-

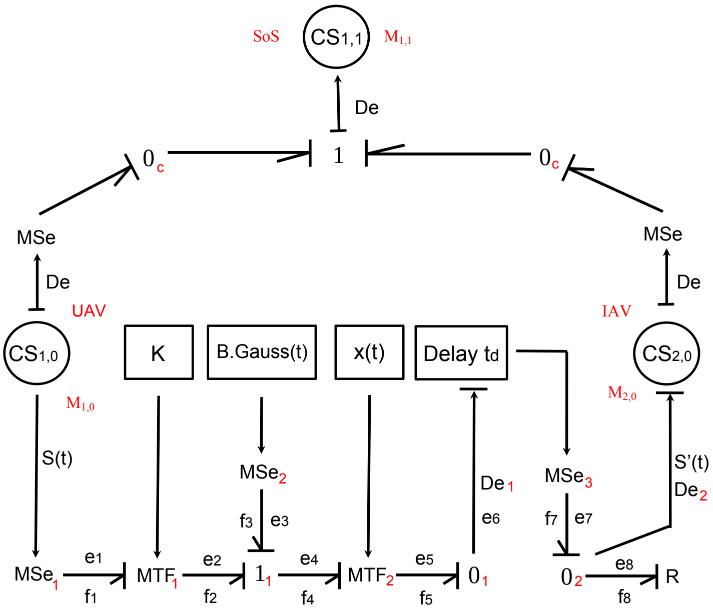


Figure 5.8 – Complete model of Hockey team SoS after adding HBG model of mission organization.

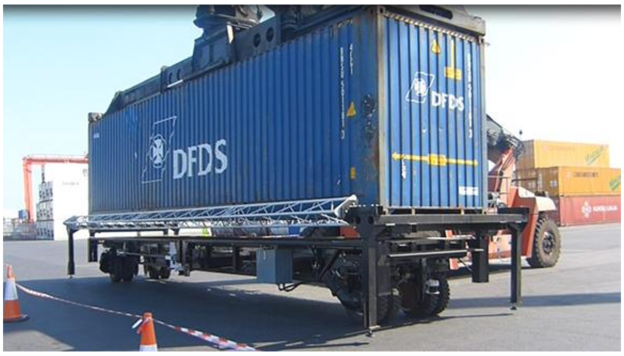


Figure 5.9 – RobuTAINer loaded with a container.

ware, designed by [Robosoft] to perform tasks such as actuator control, sensor data acquisition, feedback control and robot supervision. In addition, it has an RTK GPS receiver to acquire navigation information from satellites.

5.3 Block Diagram for RobuTAINer Navigation using a UAV

In this section we discuss our implemented solution for localizing RobuTAINer IAV using a UAV. Recall that the RobuTAINer itself has an RTK GPS receiver but the main problem is that the signal received from satellites might be disturbed by nearby obstacles, such as buildings, trees or even similar IAVs. Note that RobuTAINer IAV has a set of sensors that when it looses the satellites signals it may continue roaming safely but only for a very short time (around 20 seconds).

We build a localization box composed of the following components:

1. Raspberry Pi: This module is the processing unit of the localization box. It has a 900 MHz quad-core with 1 GB of RAM.
2. Raspberry Pi Camera: This is a very lightweight camera (3 gm) with good resolution of 5 Mega pixels.
3. Piksi GPS Receiver: This is probably the cheapest RTK GPS available in the market. The base and hover modules come with separate Tx and RX units to build the differential solution. Here we mount the hover module and at the roof of a nearby building, away from any obstacles, we place the base module.
4. Xbee TX: Reliable transmitter with a line-of-sight outdoor coverage range up to 1 mile and data rates up to 250 kbps.
5. BMP180: This is an atmospheric pressure sensor that can be used as an altimeter. It is interfaced to raspberry pi over I2C bus.
6. LSM303: This is a 3D accelerometer and 3D magnetometer sensor that can be used to measure roll, pitch and yaw angles. It is interfaced to raspberry pi over I2C bus.

This compact localization box is mounted below the UAV. On RobuTAINer IAV, we place a special drawing of set of concentric circles (more than 3). This drawing will be used to detect the RobuTAINer IAV from the UAV by means of image detection algorithm. The mechanism for detecting this RobuTAINer from UAV will be discussed in the section 5.4.

Now we place a laptop over the RobuTAINer. This laptop will run our .Net program that has two main roles:

1. Interface with RobuTAINer via PURE communication protocol to read navigation information generated by GPS receiver built inside

the RobuTAINer. A log file will be created showing navigation information from RobuTAINer.

2. Interface with UAV via Xbee pro receiver connected to usb port of the laptop. We read navigation information related to RobuTAINer generated by UAV. A log file will be created showing navigation information of RobuTAINer from UAV.

The block diagram of our solution for localizing RobuTAINer IAV by means of UAV is shown in Figure 5.10 (left) and the realistic images of UAV with localization box installed and IAV is shown in Fiure 5.10 (right). Note that the complete details for how the UAV generates the navigation information of robuTAINer will be discussed in section 5.5.

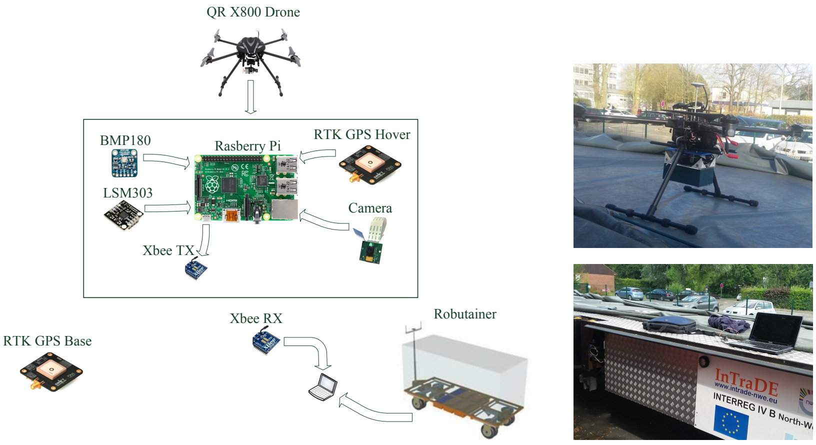


Figure 5.10 – Block diagram for robuTAINer navigation using a UAV.

5.4 RobuTAINer Detection from UAV

Now our job in this section is to detect the RobuTAINer IAV from the UAV using image processing techniques. Recall that in the previous section we said that a special drawing of concentric circles (Figure 5.11) must be placed on top of robuTAINer. Using OpenCV library in Raspberry Pi, we are able to detect circles and locate them in the frame of the camera used for imaging.

Why Choose to Draw Several Concentric Circles Instead of One?

Well, in fact, at the start of our work, we used a drawing of one circle and used ”HoughCircles” method in OpenCV to detect this circle. The

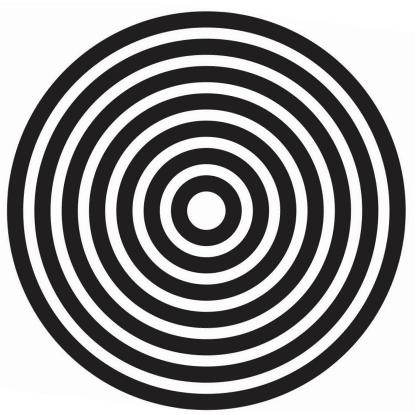


Figure 5.11 – Concentric circles drawing used to detect robuTAINer from UAV.

problem is that this method is not very accurate especially under different light conditions or indoor/outdoor environment. Accordingly, you end up detecting other false circles even if you deploy "Otsu" method for adapting to dynamic light changes.

Next, we thought of detecting special patterns on the robuTAINer but the problem was the relatively large distance between the UAV and the robuTAINer, since the UAV is supposed to maneuver 20 to 40 meters over the robuTAINer to guarantee its safety away from any nearby obstacles.

Then we consider that the optimal solution for our case is to detect a drawing where a set of concentric circles exist. Now if we run the image detection algorithm using "HoughCircles" method, several true and false circles will be detected. However, if we divide the captured image into small zones, then we are able to study the density of detected circles in each zone and accordingly, the set of concentric circles will be successfully detected regardless of other false circles.

We use high resolution images (1800 px, 1200 px). We divide each image into square zones, each with a side of 100 px. Then we detect circles. We mark the center and radius for each detected circle. Then we choose the zone that has maximum centers of circle such that this maximum exceeds 3, which means a minimum detection of three concentric circles since our drawing includes more than three concentric circles. For that zone, we average the coordinates of the detected circles and the radii. then we draw one circle having that average center and average radius. This would be the robuTAINer. The detection algorithm is depicted in Figure 5.12. Demonstration results after running this algorithm in an indoor environment are displayed in Figure 5.13.

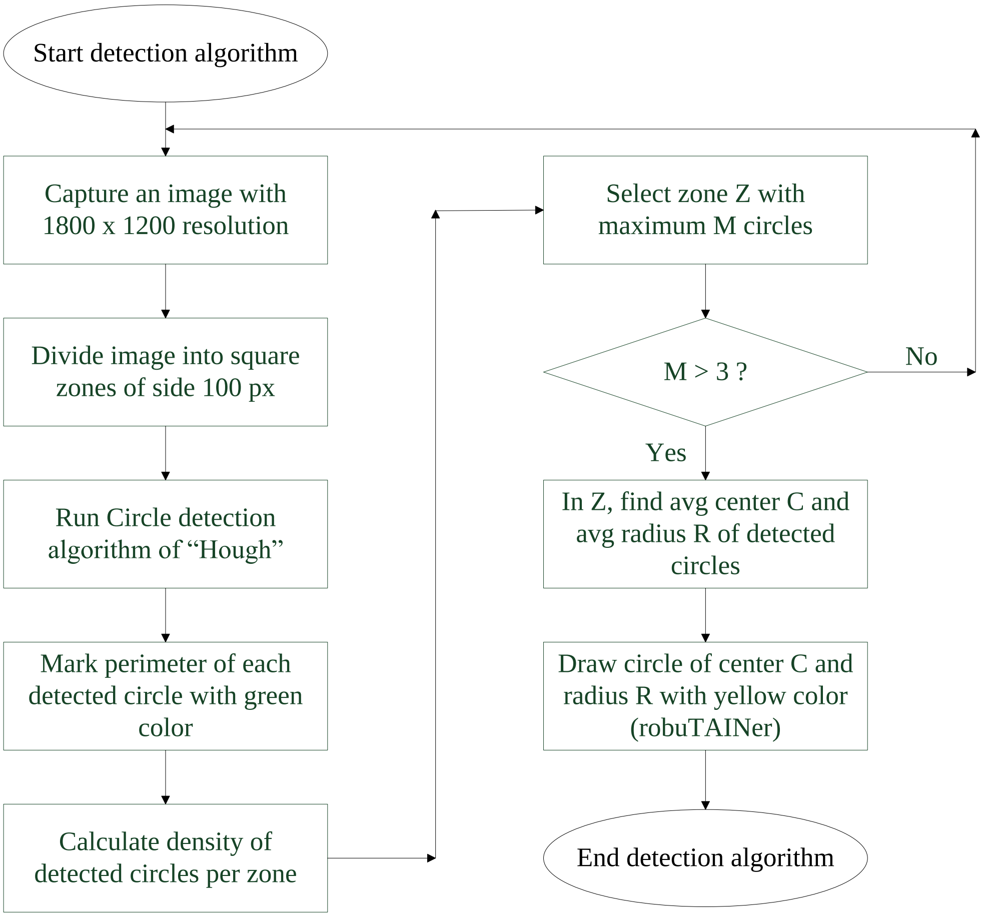

Start detection algorithm
Capture an image with 1800 x 1200 resolution
Divide image into square zones of side 100 px
Run Circle detection algorithm of "Hough"
Mark perimeter of each detected circle with green color
Calculate density of detected circles per zone
Select zone Z with maximum M circles
M > 3 ?
No
Yes
In Z, find avg center C and avg radius R of detected circles
Draw circle of center C and radius R with yellow color (robuTAINer)
End detection algorithm

Figure 5.12 – Algorithm for robuTAINer detection from UAV.

5.5 Algorithm for Robutainer Navigation

Here it is useful to note that the localization information depicted by robuTAINer itself using its own GPS receiver are relative to chosen fixed point $P\{x_0, y_0\}$. Accordingly, navigation information are supplied by PURE as x and y coordinates where unit is in meters. ***We denote by*** $\{x_r, y_r\}$***, the navigation information of robuTAINer generated by robuTAINer itself using its own GPS receiver and according to the chosen fixed point*** $P\{x_0, y_0\}$***.*** The navigation information of robuTAINer IAV supplied by the UAV should be in the same format and according to the same chosen fixed point to be able to compare both results.

The RTK GPS receiver available in localization box mounted on UAV will provide a differential solution in the form of NMEA messages containing

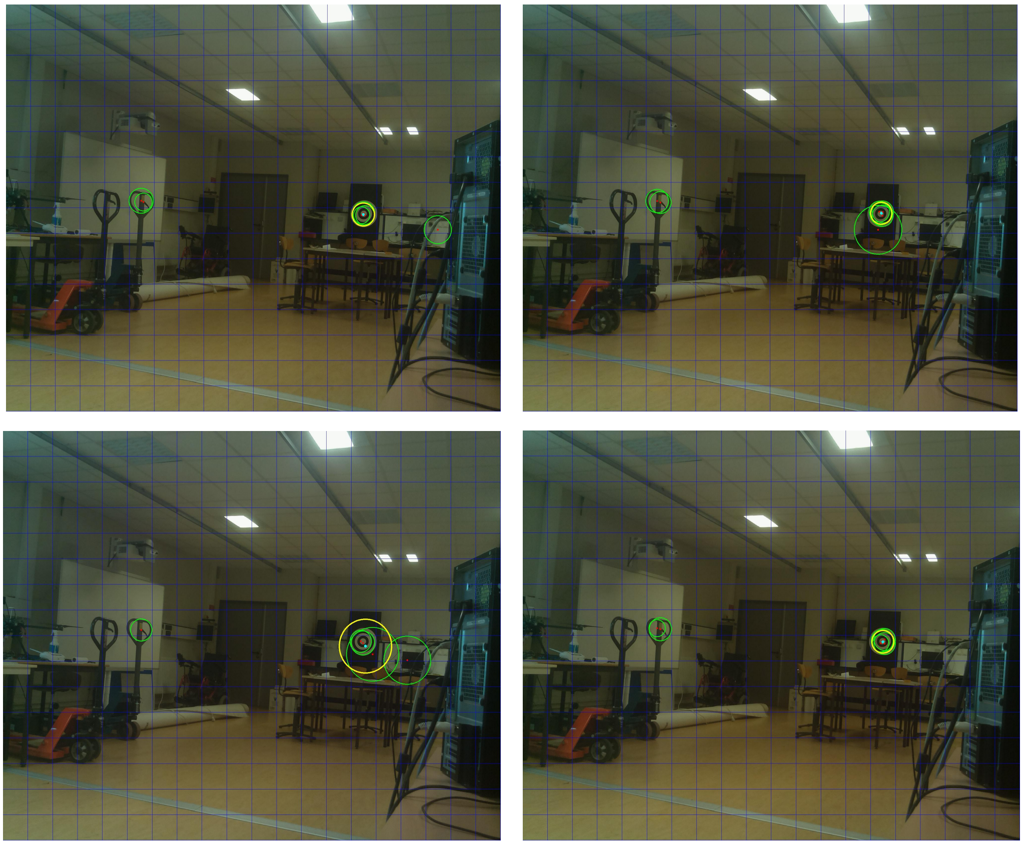


Figure 5.13 – Demonstration results of detection algorithm.

longitude in degrees, latitude in degrees and altitude in meters. We have to transform those localization information into Universal Transverse Mercator (UTM) coordinate system to get x and y coordinates in meters. Also note that we need to do translations to get the coordinates according to the chosen fixed point $P\{x_0, y_0\}$. ***We denote by*** $\{x_d, y_d\}$***, the navigation information of UAV generated by UAV itself and according to the same chosen fixed point*** $P\{x_0, y_0\}$***.***

In the previous section we used a detection algorithm for detecting the robuTAINer IAV in the frame of UAV camera. Now to be able to localize the robuTAINer in real world coordinates, we need to do an image calibration procedure. We calibrate our camera using a known pattern (Figure 5.14) and get the camera matrix and distortion coefficients which will be used next to correlate 2D points in camera framework to real world 3D coordinates. For more information on camera calibration, refer to appendix A. ***We denote by*** $\{\delta_x, \delta_y\}$ ***the differential distance in meters separating the UAV from the robuTAINer along x and y directions respectively.*** Note that the pair $\{\delta_x, \delta_y\}$ is calculated after doing an image calibration and using the yaw angle supplied by LSM303 magnetometer.

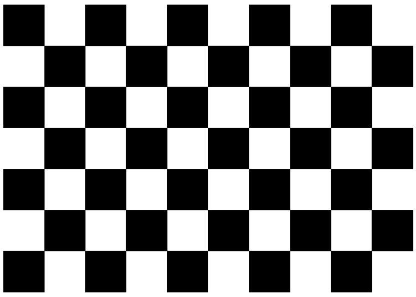


Figure 5.14 – Pattern used to calibrate camera in OpenCV.

Now we denote by $\{x_{dr}, y_{dr}\}$***, the localization information of robuTAINer generated by the UAV and according to the same chosen fixed point*** $P\{x_0, y_0\}$***.*** We conclude:

$$x_{dr} = x_d + \delta_x \tag{5.1}$$

$$y_{dr} = y_d + \delta_y \tag{5.2}$$

The algorithm for robuTAINer navigation using the UAV UAV is shown in Figure 5.15.

5.6 Experimentation and Results

In this section, we now run an experiment to test the navigation algorithm. We drive the robutainer and the UAV both manually such that the UAV is maneuvering above the robuTAINer. We had to move the robuTAINer in a very confined space with a rectangular trajectory due to limitation in available space. Recall that our concern now is not the autonomous feature but to test the navigation of robuTAINer by means of a UAV. We run a .NET application (Figure 5.16) that receives informative data from the UAV including localization information $\{x_{dr}, y_{dr}\}$ and sensors data (altimeter, magnetometer and accelerometer), and object detection information (Zone with most circles, average raduis and center of circle). This application also reads localization information $\{x_r, y_r\}$ from the robuTAINer using PURE. Both localization information are saved into log files and the results are displayed graphically for comprehensive comparison.

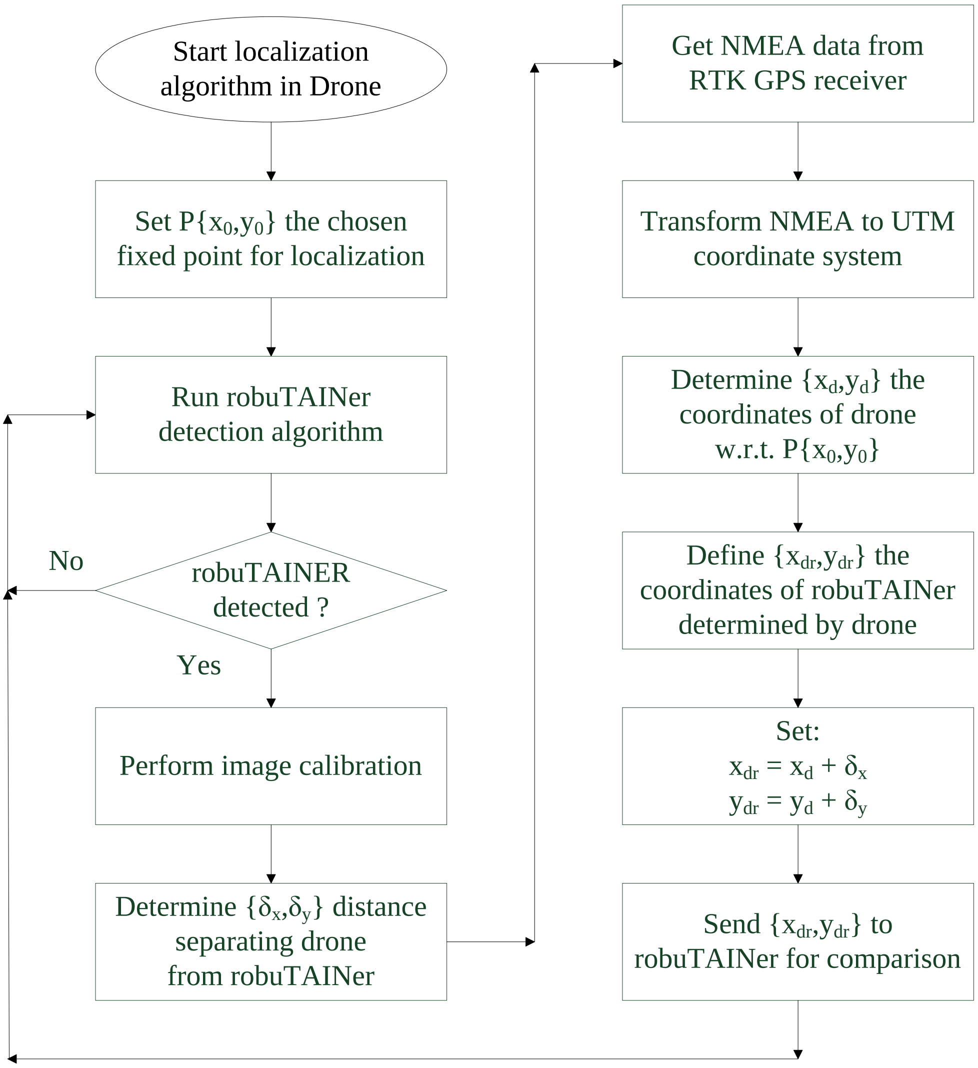
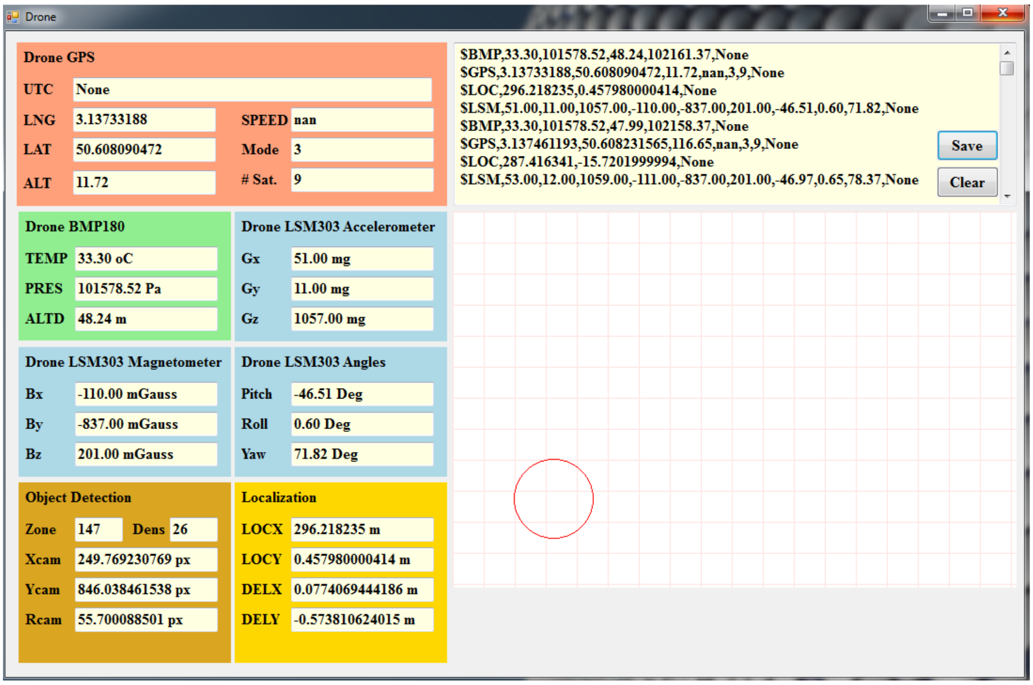

Start localization algorithm in Drone
Set P{x0,y0} the chosen fixed point for localization
Run robuTAINer detection algorithm
robuTAINER detected ?
No
Yes
Perform image calibration
Determine {δx,δy} distance separating drone from robuTAINer
Get NMEA data from RTK GPS receiver
Transform NMEA to UTM coordinate system
Determine {xd,yd} the coordinates of drone w.r.t. P{x0,y0}
Define {xdr,ydr} the coordinates of robuTAINer determined by drone
Set: xdr = xd + δx ydr = yd + δy
Send {xdr,ydr} to robuTAINer for comparison

Figure 5.15 – Algorithm for robuTAINer navigation using UAV.

After we perform the experiment, we obtain satisfactory results for most of the experiment where the RTK GPS receiver in UAV is able to see the minimum required number of satellites. Results are displayed in Figure 5.17. In all our sub-figures, we plot versus time stamps. **A time stamp is a time interval needed by UAV to complete one cycle of its navigation algorithm.**

In sub-Figure 5.17(a), we plot versus time, x_r, the abscissa of localization information of robuTAINer generated by its own GPS receiver, and x_{dr}, the



Drone GPS

UTC None

LNG 3.13733188 SPEED nan

LAT 50.608090472 Mode 3

ALT 11.72 # Sat. 9

Drone BMP180

TEMP 33.30 oC

PRES 101578.52 Pa

ALTD 48.24 m

Drone LSM303 Accelerometer

Gx 51.00 mg

Gy 11.00 mg

Gz 1057.00 mg

Drone LSM303 Magnetometer

Bx -110.00 mGauss

By -837.00 mGauss

Bz 201.00 mGauss

Drone LSM303 Angles

Pitch -46.51 Deg

Roll 0.60 Deg

Yaw 71.82 Deg

Object Detection

Zone 147 Dens 26

Xcam 249.769230769 px

Ycam 846.038461538 px

Rcam 55.700088501 px

Localization

LOCX 296.218235 m

LOCY 0.457980000414 m

DELX 0.0774069444186 m

DELY -0.573810624015 m

```
$BMP,33.30,101578.52,48.24,102161.37,None
$GPS,3.13733188,50.608090472,11.72,nan,3,9,None
$LOC,296.218235,0.457980000414,None
$LSM,51.00,11.00,1057.00,-110.00,-837.00,201.00,-46.51,0.60,71.82,None
$BMP,33.30,101578.52,47.99,102158.37,None
$GPS,3.137461193,50.608231565,116.65,nan,3,9,None
$LOC,287.416341,-15.7201999994,None
$LSM,53.00,12.00,1059.00,-111.00,-837.00,201.00,-46.97,0.65,78.37,None
```

Figure 5.16 – Application for interfacing with UAV and robutainer.

abscissa of localization information of robuTAINer generated by UAV. The maximum error, around 7 meters, being in the time stamp interval {85,95}.

In sub-Figure 5.17(b), we plot versus time, y_r, the ordinate of localization information of robuTAINer generated by its own GPS receiver, and y_{dr}, the ordinate of localization information of robuTAINer generated by UAV. The maximum error, around 12 meters, being in the time stamp interval {20,30}.

In sub-Figure 5.17(c), we plot y_r versus x_r and y_{dr} versus x_{dr} simultaneously. Again, as we can see, UAV localization results are acceptable for when the RTK GPS receiver in UAV is able to see the minimum required number of satellites to generate an RTK solution.

5.7 Conclusion

In this chapter we present a case study on cooperative behavior of two CSs in a SoS concept, with one CS being a UAV and the other CS being an IAV. The main objective of this SoS is to provide the IAV with accurate navigation information for safe roaming in confined space, say seaport terminal for example. The main drive for this cooperative behavior between UAV and IAV is that the IAV, robuTAINer in our case, cannot continue its mission if

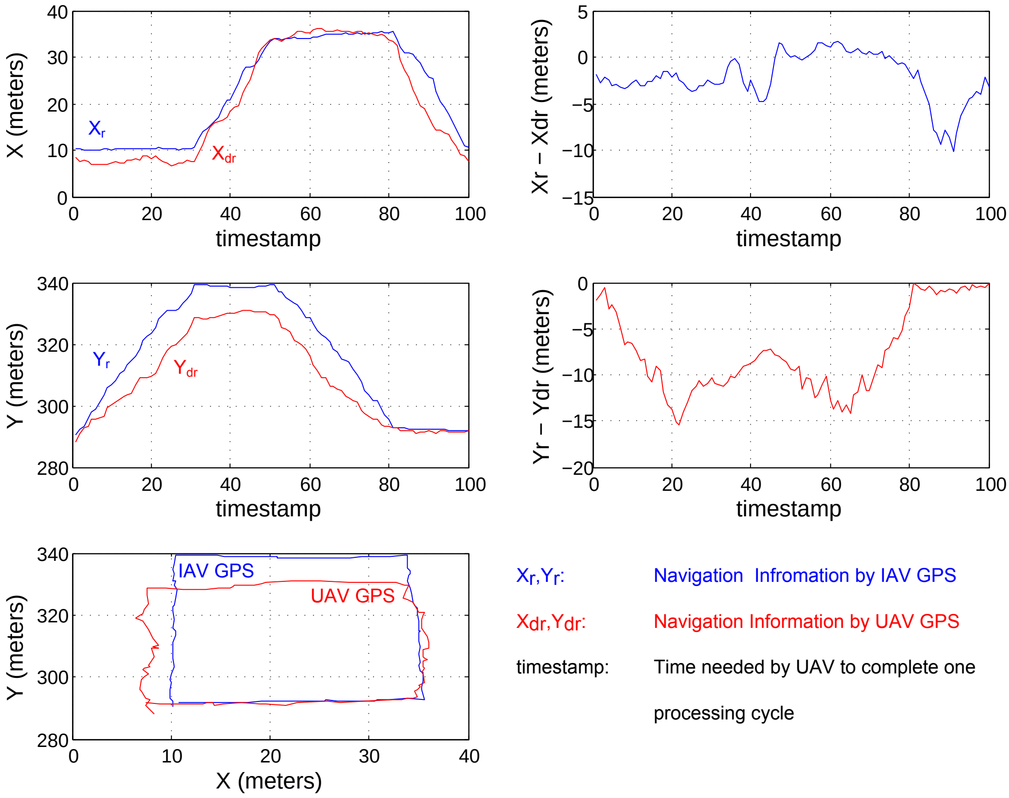


Figure 5.17 – Experiment results.

its equipped GPS receiver loses satellites signals due to perturbations from nearby obstacles.

Accordingly, we introduce another CS, namely the UAV, equipped with a localization box composed of a Raspberry Pi module, camera, altimeter, magnetometer, accelerometer, RTK GPS receiver, and a radio frequency transmitter. Using this localization box, the UAV is capable to send navigation information to the UAV such that the robuTAINer can continue its mission even if its equipped GPS receiver loses satellites signals. The GPS on the UAV has a very limited possibility of losing the satellites signals since the UAV is supposed to fly high away from ground obstacles.

We test our navigation algorithm by manually driving the robuTAINer and the UAV. we install a laptop on the robuTAINer and communicate with both CSs at the same time. We record the localization information of the robuTAINer generated by the robuTAINer itself and that transmitted by the UAV.

Experiment results have been satisfactory when the GPS equipped on the UAV was able to provide an RTK solution. The major problem is that the time cycle, or stamp, needed by the localization box to complete one cycle of its operation requires around 6 seconds, mainly due to object detection method for detecting the robuTAINer. The fact that we are using high resolution images to be able to detect circles from a considerable range around 10 to 20 meters causes the "HoughCircles" method to operate slowly on Raspberry Pi. One future solution for this problem would be using compact computer system (evaluated embedded systems), other than Raspberry Pi, with much higher processing speed, or planning for more effective object detection techniques. It is also useful to note that the mentioned results were acquired with only one experimentation session. With more time allocated for testing, the obtained results can be optimized further. However, we have demonstrated that the principle of reproducing a redundancy in GPS information through a cooperation in SoS works appropriately.

Chapter 6

Conclusion and Perspective

A SoS consists of CSs that are spread across several hierarchical levels. In each level, CSs admit various structures [Jamshidi 09b]. A graphical model is essential to represent CSs at each level and for each structure [Luzeaux 08]. Since CSs are dispersed geographically, then such a model must expose the operational independence among CSs and their interaction. A graphical approach was proposed in [Khalil 12] using HG model of SoS. HG generalizes the concept of the graph by introducing the notion of hyperedge which is not subject to a limit on the number of connected nodes, in our case CSs. HG model of SoS allows defining superposition strategies with possibility of detecting local and global faults within the modeled system. However, accidental failure of elements within a CS, say actuator for example, at microscopic level cannot be modeled using HG alone, which means that we are unable to relate the overall operation of a SoS when such failure occurs.

In engineering, another graphical modeling tool is used to model the interaction of several energy domains in power systems called BG. We can model the bidirectional exchange of energy in a physical dynamic CS using BG and deduce the structural and behavioral equations, which means we are able to model faults within a CS using BG. Accordingly, our main contribution in this thesis is to develop a multilevel graphical model for SoS based on integrating BG model at microscopic level with HG model at macroscopic level. We start by presenting a BG model for a WCL between two CSs in SoS based on the fact that CSs are dispersed geographically with no energy exchange but only information exchange.

In [Khalil 12], WCL between two CSs is not well defined. In [Kumar 14], WCL is modeled using BG by means of detectors of effort or flow. In [Soyez 15], the communication link between CSs is modeled mathematically by means of a function that specifies the minimum number of messages transmitted per second. Such CS is called "*quayTransmitter*". Consequently, Channel effects are never considered in all previous SoS models in [Khalil 12], [Kumar 14] and [Soyez 15]. In this thesis work, we consider linear channel effects such as attenuation and nonlinear channel effects including distortion, time delays and data loss. The main benefit of this model is that we can determine the waveform of the received signal at the information sink. We define five parameters in our model: PTX, NL, RSS, DPL and RTT. We test our model by choosing numerical values for the parameters in two different case. Then we perform an experiment on cooperative behavior of humanoids in a SoS concept and demonstrate how we can measure the values of those parameters experimentally.

Next we use BG model of WCL to assess quantitatively, the FTL of WCL in SoS. We define the term fault tolerance in the notion of SoS on communication basis. We define two square matrices DL and R dedicated for direct links and redundancy of each link respectively. DL_{ij} entries are either set or cleared depending whether the corresponding WCL from $CS_{i,0}$ to $CS_{i,j}$ is valid or not. A valid link is one where parameter values RSS, DPL and RTT are within acceptable range. R_{ij} entries denote the reliability of WCL from $CS_{i,0}$ to $CS_{i,j}$. A unity value indicates full reliability. We developed a simulator that demonstrates how we calculate R_{ij} entries in a SoS of n CSs.

After that, we present our main contribution in this thesis which is a methodology for coupling HG and BG in SoS modeling. Recall that the main advantage for this coupled model is that we are able to model simultaneously the organization and behavior of CSs in SoS. this methodology is organized by 7 different steps as follows:

1. Model SoS set-based representation using HG.
2. Generate the valued graph corresponding to set-based representation of SoS.
3. Generate the dual graph of the resulting valued graph.
4. Use BG to model physical CSs in the resulting dual graph.
5. Use BG to model WCL among physical CSs in the resulting dual graph.
6. Use HBG to model the management of missions attributed to CSs.
7. Check the fundamental organizational properties of the SoS.

Then we consider an example on SoS Hockey team of six omnidirectional robotic systems called Robotino. We model the behavior of CSs, WCL among CSs and the organization of missions using controlled junction concept in HBG.

Finally, we study the cooperative behavior of a UAV and an IAV called robuTAINer in the context of SoS. We discuss the main problematic in maneuvering IAV in a confined space where nearby obstacles can destroy the satellites signals received by GPS receiver mounted on IAV to supply navigation information. We develop a localization box mounted on the UAV to localize the IAV using an RTK GPS receiver on UAV and other sensors such as accelerometer, magnetometer, a camera and radio frequency transmitter. We develop an application that reads navigation information from the IAV and the UAV simultaneously. We run an experiment by manually driving the UAV and robuTAINer such that the UAV is above the robuTAINer. The role of the UAV is to detect the IAV and then localize it. The experiment produced satisfactory results when the GPS receiver mounted on UAV was able to provide a fixed RTK solution in case the hover and base modules can see a minimum of six satellites in common.

At the end, we believe that the research and contributions conducted in this thesis are essential for unified multilevel modeling of a SoSE. However, some points can be considered for future work:

— From the modeling perspective, we need to study the feasibility of developing a unified multi level model of SoS using alternative graphical tool to BG including both aspects: behavioral and organizational. This is due to the fact that using BG to model information exchange among CSs and their organization is not effective when it comes to fault detection and isolation since there is no flow of power, unlike at the physical level. Recall that all bonds in WCL model and organization of CSs are effort activated (flow = 0) with no flow of power.

  On the other hand, when modeling WCL among CSs, we can consider two additional nonlinear channel parameters to be able to depict better the received waveform: delay spread and doppler spread. Recall that in wireless communications, the receiver will receive multiple versions of the same transmitted signal due to reflections, diffractions, and scattering of transmitted signal along the communication path. The delay spread is the average time elapsed between first version and last version received of the same signal. Delay spread decides the minimum symbol rate of transmitted signal. Doppler spread is the range of frequency shifts due to doppler effect resulting from movement of receiver or movement of nearby objects. Doppler spread causes random frequency modulations in the received waveform.

— From the experimental point of view, we need to conduct an experiment with several robotized CSs in the notion of SoS and try to simulate all channel effects, linear and non linear, and exploit the usage of redundant links in case any of the WCLs fail and at the same time simulate a failure in one of the actuators of a CS. Although such experiment needs a lot of engineering and logistics, it would be great to verify, by experimentation, the whole benefits of multilevel graphical modeling of SoS. In addition, we need to optimize the object detection algorithm used in UAV and use a controller much faster that Raspberry Pi to minimize the time cycle for generating localization information of the robuTAINer.

Appendix

Appendix A

Camera Calibration using OpenCV

Camera calibration is a necessary step in 3D computer vision in order to extract metric information from 2D images. We compute intrinsic and extrinsic camera parameters using correspondences between a set of point features in the world (X, Y, Z) and their projections in an image (u, v) (Figure A.1).

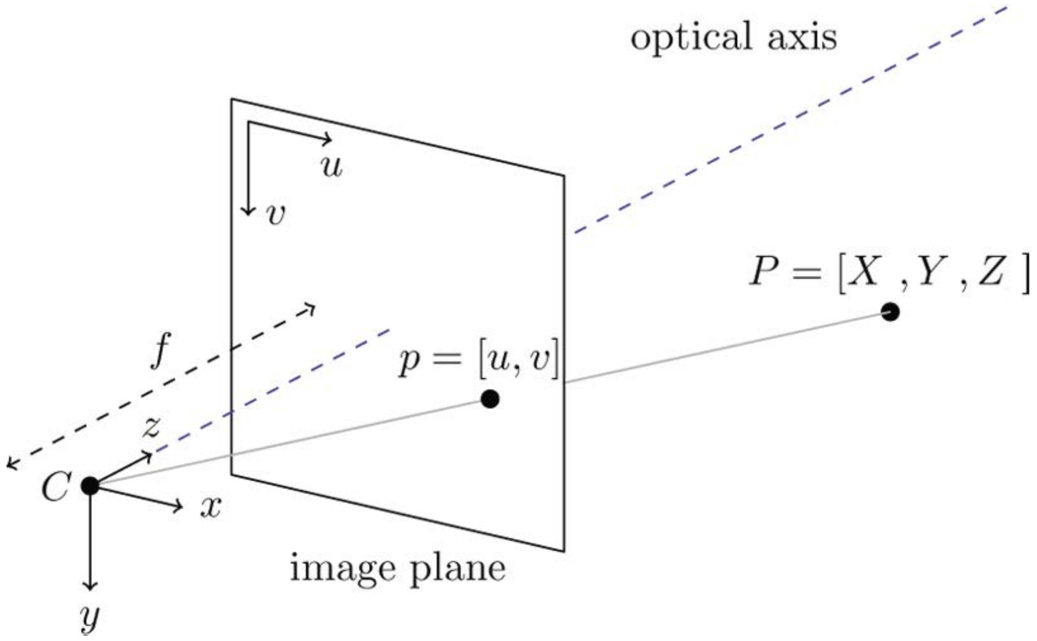


Figure A.1 – Projection of 3D point on image plane.

The world coordinates (X, Y, Z) and their projections in an image (u, v) are related by the following equation:

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \overbrace{\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}}^{A} \overbrace{\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix}}^{[R|T]} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Where A is the intrinsic camera matrix, $[R|T]$ is the joint rotational-translational matrix of extrinsic parameters, (c_x,c_y) is a principal point at image center, f_x and f_y are focal lengths expressed in pixels. The matrix of intrinsic parameters A does not depend on the scene viewed. So, once estimated, it can be re-used as long as the focal length is fixed. [R—T] is used to describe the camera motion around a static scene, or rigid motion of an object in front of a still camera.

Real lenses usually have some distortion, mostly radial distortion and slight tangential distortion. We denote by k_1, k_2, k_3, k_4, k_5, and k_6 the radial distortion coefficients and by p_1 and p_2 the tangential distortion coefficients. Distortion coefficients are considered as intrinsic parameters and do not depend on scene viewed or camera resolution, whereas f_x, f_y, c_x, and c_y need to be re-scaled if camera resolution is changed.

A calibration pattern is an image with known patterns such as chess board. In [OpenCV], to find the camera intrinsic and extrinsic parameters from several views of a calibration pattern, you may use a built in function "*calibrateCamera*". This function takes, as arguments, a set of corresponding world and image points, their count, image size. It outputs the camera distortion coefficients, matrix of intrinsic parameters A and matrix of extrinsic parameters $[R|T]$ as translational vectors and rotational vectors.

Appendix B

Bond Graph Basics

BG is a graphical modeling tool for exchange of power between elements in multi-physical domains. *Henry Painter* from MIT primarily contributes to existence of BG back in 1959. In BG, power is denoted by two variables effort e and flow f. As shown in Figure B.1(a), the power exchanged between two elements A and B is indicated by a bond and is the product of the two variables e and f. The direction of the positive power is indicated by the half-arrow on the bond. The power Pu corresponds to the product between power variables effort and flow:

$$P_u(t) = e(t).f(t) \tag{B.1}$$

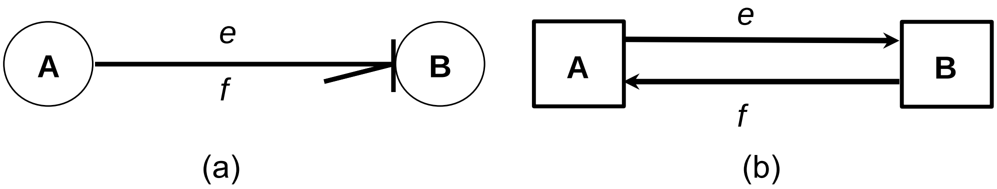


Figure B.1 – BG representation and causality.

The notion of causality in BG provides a robust tool for identifying system equations, discussion of system behavior, controllability, observability, and fault diagnosis ([Loureiro 12], [Samantaray 08]). The direction of the bond indicates the way in which power flows. Causality in a BG represents the way in which the unknown variable is calculated and is denoted by a perpendicular stroke to one end of the bond. Effort flows from the side without the stroke to that with stroke. In block diagram given by Figure B.1(b), the direction of action is indicated by an arrow on each connection as illustrated. Figure B.2 depicts the information given by BG representation.

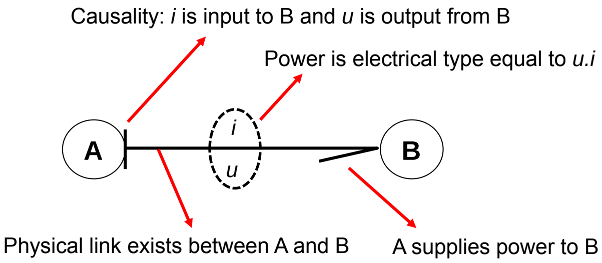

Causality: *i* is input to B and *u* is output from B
Power is electrical type equal to *u.i*
A
i
u
B
Physical link exists between A and B
A supplies power to B

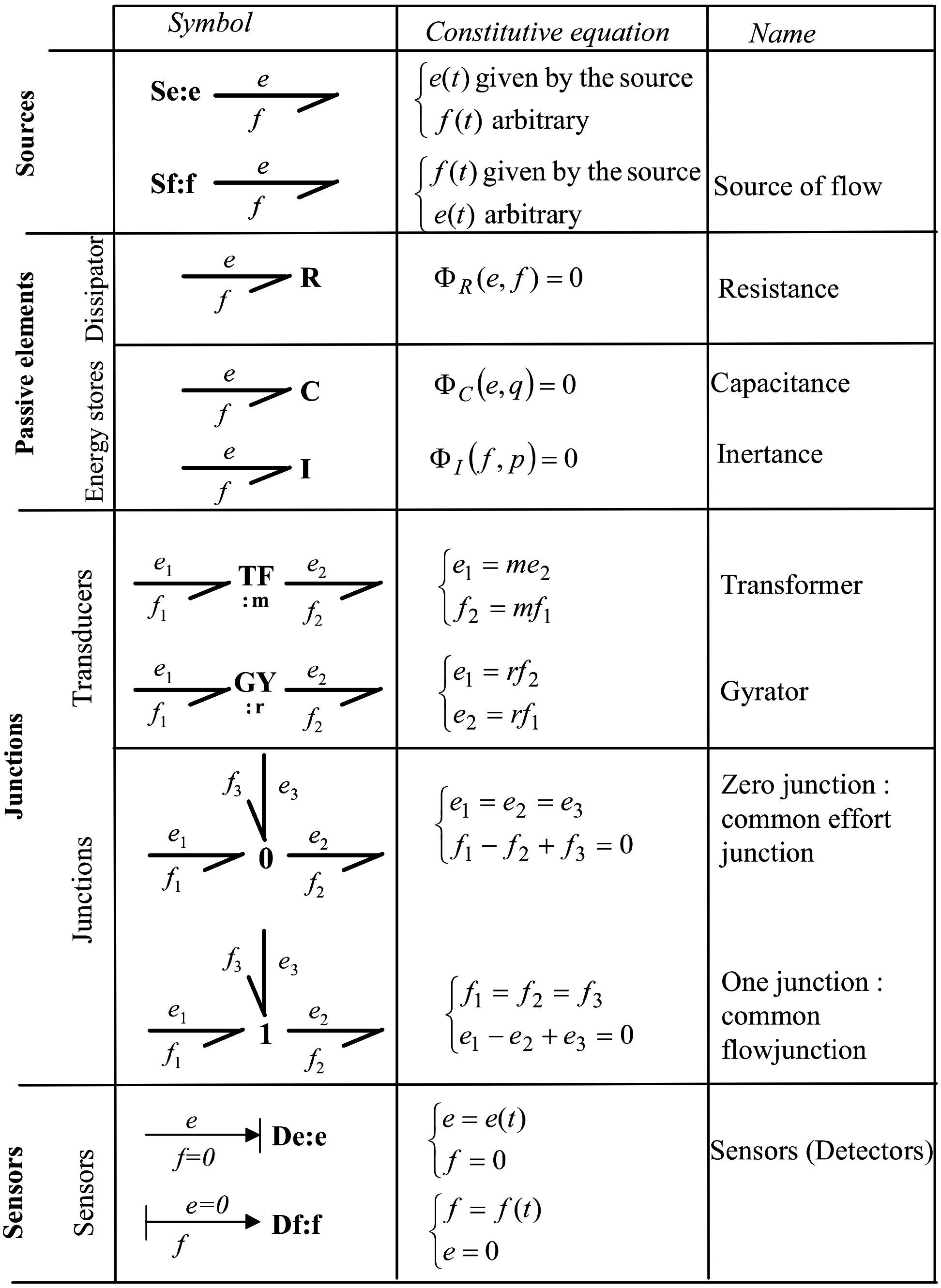
Figure B.2 – Information given by BG representation.

Two types of causality may be defined: the integral causality used for dynamic simulation and the derivative causality used for diagnosis. In integral causality, energy variables are expressed as the integral of the input (effort or flow). Whereas in derivative causality, energy variables are expressed as the derivative of the input. Table 1 gives power variables, effort and flow, for some of physical domains.

Table 1
Power variables in a true bond graph

Domain	Effort $e(t)$	Flow $f(t)$
Electric	Voltage u (V)	Current i (A)
Mechanics rotation	Torque Γ (N m)	Angular velocity ω (rad/s)
Mechanics translation	Force F (N)	Velocity v (m/s)
Hydraulics	Pressure (Pa)	Volume flow rate $\dot{V}$ (m^3/s)
Thermodynamics	Temperature (K)	Entropy flow (J/K s)
Chemistry		
Transformation phenomenon	Chemical potential μ (J/mol)	Molar flow rate $\dot{n}$ (mol/s)
Kinetic phenomena	Chemical affinity A	Speed of reaction $\dot{\xi}$

In BG, we define nine elements used to model any energetic process: two active elements (called sources) (Se and Sf), three generalized passive elements (I, C,and R) two junctions (0 and 1) and two transducers (TF and GY). When the exchanged power is negligible (either e or f is zero), it is represented by an information bond. Figure B.3 shows a summary of BG elements and their corresponding definitions.



		Symbol	*Constitutive equation*	*Name*
Sources		**Se:e** $\overset{e}{\underset{f}{\rightharpoonup}}$	$\begin{cases} e(t) \text{ given by the source} \\ f(t) \text{ arbitrary} \end{cases}$	
		Sf:f $\overset{e}{\underset{f}{\rightharpoonup}}$	$\begin{cases} f(t) \text{ given by the source} \\ e(t) \text{ arbitrary} \end{cases}$	Source of flow
Passive elements	Dissipator	$\overset{e}{\underset{f}{\rightharpoonup}}$ **R**	$\Phi_R(e,f)=0$	Resistance
	Energy stores	$\overset{e}{\underset{f}{\rightharpoonup}}$ **C**	$\Phi_C(e,q)=0$	Capacitance
		$\overset{e}{\underset{f}{\rightharpoonup}}$ **I**	$\Phi_I(f,p)=0$	Inertance
Junctions	Transducers	$\overset{e_1}{\underset{f_1}{\rightharpoonup}}$ **TF :m** $\overset{e_2}{\underset{f_2}{\rightharpoonup}}$	$\begin{cases} e_1 = me_2 \\ f_2 = mf_1 \end{cases}$	Transformer
		$\overset{e_1}{\underset{f_1}{\rightharpoonup}}$ **GY :r** $\overset{e_2}{\underset{f_2}{\rightharpoonup}}$	$\begin{cases} e_1 = rf_2 \\ e_2 = rf_1 \end{cases}$	Gyrator
	Junctions	f_3 $\downarrow$ e_3 $\overset{e_1}{\underset{f_1}{\rightharpoonup}}$ **0** $\overset{e_2}{\underset{f_2}{\rightharpoonup}}$	$\begin{cases} e_1 = e_2 = e_3 \\ f_1 - f_2 + f_3 = 0 \end{cases}$	Zero junction : common effort junction
		f_3 $\downarrow$ e_3 $\overset{e_1}{\underset{f_1}{\rightharpoonup}}$ **1** $\overset{e_2}{\underset{f_2}{\rightharpoonup}}$	$\begin{cases} f_1 = f_2 = f_3 \\ e_1 - e_2 + e_3 = 0 \end{cases}$	One junction : common flowjunction
Sensors	Sensors	$\overset{e}{\underset{f=0}{\rightarrow}}\vert$ **De:e**	$\begin{cases} e = e(t) \\ f = 0 \end{cases}$	Sensors (Detectors)
		$\vert\overset{e=0}{\underset{f}{\rightarrow}}$ **Df:f**	$\begin{cases} f = f(t) \\ e = 0 \end{cases}$	

Figure B.3 – Basic BG elements with their definitions.

Appendix C

Notions on Hyperset Theory

Definition C.1: A hyperset *HE* is a set containing both: simple elements and hyper elements (set of elements) [Chemero 08].

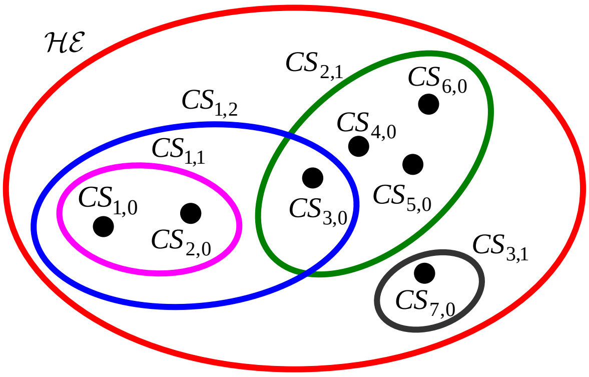


Figure C.1 – A hyperset HE.

For the hyperset *HE* presented in Figure C.1, we note:

$$HE = \{\{\{CS_{1,0}, CS_{2,0}\}, CS_{3,0}\}, \{CS_{3,0}, CS_{4,0}, CS_{5,0}, CS_{6,0}\}, \{CS_{7,0}\}\}$$

Where $CS_{1,0}$, $CS_{2,0}$, $CS_{3,0}$, $CS_{4,0}$, $CS_{5,0}$, $CS_{6,0}$, $CS_{7,0}$ are (vertices) in *HE*. We realize that the hyper set is composed of four elements, stated as follows:

Three basic elements:

$$CS_{1,1} = \{CS_{1,0}, CS_{2,0}\}$$

$$CS_{2,1} = \{CS_{3,0}, CS_{4,0}, CS_{5,0}, CS_{6,0}\}$$

$$CS_{3,1} = \{CS_{7,0}\}$$

One hyper element:

$$CS_{1,2} = \{\{CS_{1,0}, CS_{2,0}\}, CS_{3,0}\}$$

Definition C.2: We say that $CS_{i,j}$ is an element in HE for j $\neq 0$, noted as $CS_{i,j} \Subset HE$, if and only if:

$$CS_{i,j} \in HE \, or$$

$$CS_{i,j} \Subset CS_{i',j'} / CS_{i',j'} \in HE$$

Definition C.3: Two elements $CS_{i,j}$, $CS_{i',j'} \Subset HE$ are said to be independent if:

$$CS_{i,j} \bigcap CS_{i',j'} = \emptyset$$

Definition C.4: Two elements $CS_{i,j}$, $CS_{i',j'} \Subset HE$ are said to be inclusive if:

$$CS_{i,j} \subseteq CS_{i',j'} or$$

$$CS_{i',j'} \subseteq CS_{i,j}$$

Definition C.5: A hyperset HE is said to be nested if there is no independence or inclusion in any of its elements (Figure C.2).

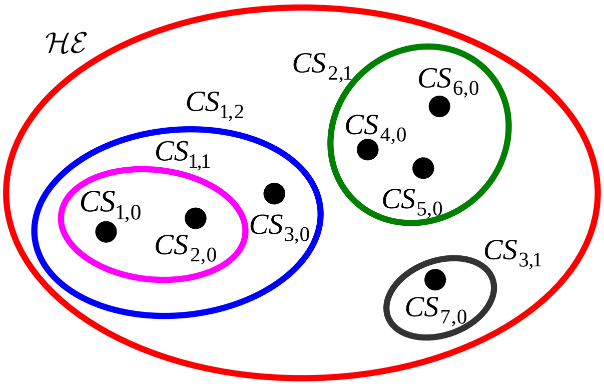


Figure C.2 – A nested hyperset HE.

Bibliography

[20Sim] *www.20sim.com*

[Agusdinata 09] Agusdinata, D. B., Dittmar, L., and DeLaurentis, D. (2009). *Systems of systems engineering: principles and applications.* CRC press, Taylor and Francis Group, editor: M. Jamshidi. Chapter 8.

[Aldebaran] *www.aldebaran.com*

[Bancroft 85] Bancroft, S. (January 1985). *An Algebraic Solution of the GPS Equations.* IEEE Transactions on Aerospace and Electronic Systems. AES-21: 5659. Bibcode:1985ITAES..21...56B. doi:10.1109/TAES.1985.310538. Archived from the original on 20 May 2013.

[Berge 87] C. Berge, *Hypergraphes : Combinatoires des ensembles finis*, Gauthier-Villars, Paris, 1987.

[Caffall 09] Caffall, D. S. and Michael, J. B. (2009). *Systems of systems engineering: principles and applications.* CRC press, Taylor and Francis Group, editor: M. Jamshidi. Chapter 15.

[Campbell 05] James E. Campbell, Dennis E. Longsine, Donald Shirah, and Dennis J. Anderson, *System of Systems Modeling and Analysis*, SAND REPORT , SAND2005-0020, January 2005

[Carlock 01] Carlock, P. G. and Fenton, R. E. (2001). *System of systems (sos) enterprise systems engineering for information-intensive organizations.* Systems Engineering, 4(4):242261.

[Chemero 08] A. Chemero and M. T. Turvey, *Autonomy and hypersets*, BioSystems, vol. 91, no. 2, pp. 320-330, Feb. 2008.

[Colgren 09] Colgren, R. (2009). *Systems of systems engineering: principles and applications.* CRC press, Taylor and Francis Group, editor: M. Jamshidi. Chapter 13.

[Dahmann 09] Dahmann, J. S. (2009). *System of Systems Engineering Innovations for the 21st Century.* John Wiley and Sons, editor: M. Jamshidi. Chapter 9.

[DAG 10] DAG (2010). Defense Acquisition Guidebook (DAG), August 2010. Available online: *http://at.dod.mil/docs/DefenseAcquisitionGuidebook.pdf.*

[DeLaurentis 09] DeLaurentis, D. A. (2009). *System of Systems Engineering Innovations for the 21st Century.* John Wiley and Sons, editor: M. Jamshidi. Chapter 20.

[Dickerson 09] Dickerson, C. E. (2009). *Systems of systems engineering: principles and applications.* CRC press, Taylor and Francis Group, editor: M. Jamshidi. Chapter 12.

[Dubrova 13] E. Dubrova, *Fault-Tolerant Design*, Springer, 2013, ISBN 978-1-4614-2112-2, 2013

[Duchon 12] F. Duchon, P. Hubinsky, J. Hanzela, A. Babineca, M. Tolgyessy, *Intelligent vehicles as the robotic applications*, Science Direct, Procedia Engineering 48, pages 105 to 114, 2012

[Gelareh 13] S. Gelareh, R. Merzouki, K. McGinley, R. Murray, *Scheduling of Intelligent and Autonomous Vehicles Under Pairing/Unpairing Collaboration Strategy in Container Terminals*, Transportation Research Part C 33 (2013) 1-21.

[Guier 97] Guier, William H.; Weiffenbach, George C. *Genesis of Satellite Navigation* (PDF). Johns Hopkins APL Technical Digest 19 (1): 178181. (1997)

[Guo 11] H. Guo, *A simple algorithm for fitting a Gaussian function*, IEEE Sign. Proc. Mag. 28(9): 134-137 (2011).

[Gough 13] L. Gough, G. Thomas, B. Li, *Differences in RSSI readings made by different Wi-Fi chipsets: A limitation of WLAN localization*, Localization and GNSS (ICL-GNSS), 2011 International Conference on, Retrieved 24 June 2013.

[Hata 09] Hata, Y., Kobashi, S., and Nakajima, H. (2009). *Systems of systems engineering: principles and applications.* CRC press, Taylor and Francis Group, editor: M. Jamshidi. Chapter 9.

[Hatch 06] M.J. Hatch, *Organization Theory*, Oxford University Press, Oxford, 2006.

[Haykin 01] S. Haykin, *Communication Systems*, John Wiley and Sons, USA, 2001.

[Hipel 09] Hipel, K. W., Obeidi, A., Fang, L., and Kilgour, D. M. (2009). *System of Systems Engineering Innovations for the 21st Century.* John Wiley and Sons, editor: M. Jamshidi. Chapter 18.

[InTrade] *www.intrade-nwe.eu*

[Jacobson 88] V. Jacobson, M. J. Karels, *Introduction to Congestion Avoidance and Control*, November, 1988

[James 13] Ladyman, James; Lambert, James; Wiesner, Karoline (2013). *What is a Complex System?*. European Journal for Philosophy of Science 3: 3367.

[Jamshidi 09a] M. Jamshidi, *System of Systems Engineering Innovations for the 21st Century*, John Wiley and Sons, editor: M. Jamshidi, 2009

[Jamshidi 09b] M. Jamshidi, *System of Systems Engineering: Principles and Applications*, CRC press. Taylor and Francis Group, editor: M. Jamshidi, 2009

[Jamshidi 09c] M Jamshidi, *System-of-Systems Engineering - A Definition*, IEEE SMC, 2005, Big Island, Hawaii.

[Johnson 84] B. W. Johnson, *Fault-Tolerant Microprocessor-Based Systems*, IEEE Micro, vol. 4, no. 6, pp. 621, 1984

[Jolly 09] Jolly, S. D. and Muirhead, B. K. (2009). *System of Systems Engineering Innova-tions for the 21st Century.* John Wiley and Sons, editor: M. Jamshidi. Chapter 14.

[Keating 03] Keating, C., Rogers, R., Unal, R., Dryer, D., Sousa-Poza, A., Sakord, R., Peterson, W., and Rabadi, G. (2003). *System of systems engineering.* Engineering Management Journal, 15(3)

[Khalil 11] W. Khalil, R. Merzouki, B. Ould-Bouamama, H. Haffaf, *System of Systems Architectural Modeling*, PAPYRUS Workshop on Fault Diagnosis and Fault Tolerant Control in large scale processing industries, October 6 - 7, 2011, Corsica, France.

[Khalil 12] W. Khalil, R. Merzouki, B. Ould-Bouamama, H. Haffaf, *Hypergraph Models for System of Systems Supervision Design*, IEEE Transactions on Systems, Man, and Cybernetics, Part A, Volume 42, Issue (4), pages 1005-1012, 2012.

[Koubeissi 15] A. Koubeissi, M. Ayache, R. Merzouki, B. Conrard, *Bond Graph Model-Based for Fault Tolerance Level Assessment of a Wireless Communication Link in a System of Systems Concept*, 10^{th} IEEE Systems of Systems Engineering Conference (SoSE), Page(s): 358 - 363, San Antonio, Texas, USA, May 2015.

[Kumar 14] P. Kumar, R. Merzouki, B. Conrard, V. Coelen, B. Ould Bouamama, *Multilevel Modeling of the Traffic Dynamic*, IEEE Transactions on Intelligent Transportation Systems, vol. no.99, pp.1,17, 2014.

[Laprie 85] J. C. Laprie, *Dependable Computing and Fault Tolerance: Concepts and Terminology*, Proceedings of 15th International Symposium on Fault-Tolerant Computing (FTSC-15), pp. 211, 1985

[Loureiro 12] R. Loureiro, R. Merzouki, B. Ould-Boumama, *Bond Graph Model Based on Structural Diagnosability and Recoverability Analysis : Application to Intelligent Autonomous Vehicles*, IEEE Transactions on Vehicular Technology, Vol.61, Issue 3, pp. 986-997, March 2012.

[Lukasik 98] Lukasik, S. J. (1998). *Systems, systems of systems, and the education of engineers.* AI EDAM, 12(01):55-60.

[Luzeaux 08] D. Luzeaux et J.R. Ruault, *Ingnierie des systmes de systmes-mthodes et outils*, IC2 informatique et systmes, Hermes Science Publications, Paris, 2008.

[Maier 96] M. Maier, *Architecting Principles for Systems-of-Systems*, in proceeding of the Sixth Annual International Symposium INCOSE, INCOSE, Boston, 1996, pp. 567-574.

[Maier 98] Maier, M. W. (1998). *Architecting principles for systems-of-systems.* Systems Engineering, 1(4):267-284.

[Mansouri 09] Mansouri, M., Gorod, A., Wakeman, T. H., and Sauser, B. (2009). *Maritime transportation system of systems management framework: a system of systems engineering approach.* International Journal of Ocean Systems Management, 1(2):200226.

[Menychtas 12] A. Menychtas, K. Konstanteli, *Fault Detection and Recovery Mechanisms and Techniques for Service Oriented Infrastructures*, Achieving Real-Time in Distributed Computing: From Grids to Clouds, IGI Global, pp. 259274, doi:10.4018/978-1-60960-827-9.ch014, 2012

[Merzouki 10] R. Merzouki, K. Fawaz, B. Ould-Bouamama, *Hybrid fault diagnosis for telerobotics system*, Mechatronics, Volume 20, Issue 7, Pages 729-738, October 2010.

[Merzouki 14] Industrial Patent, R. Merzouki, V. Coelen, B. Conrard, M. Pollart, *AUTONOMOUSLY ASSISTED AND GUIDED VEHICLE*, 4B:B7:95:7F:E8:20:1F:C8:6E:11:92:28:56:47:11:3B:FB:00:6E:B8

[Ming 10] Y. Ming, *Fault Diagnosis and Prognosis of Hybrid Systems using Bond Graph Models and Computational Intelligence*, Thesis, Nanyang Technological University, 2010

[Molisch 11] A. Molisch, *Structure of a Wireless Communication Link*, Wireless Communications, Pages: 181 - 186, DOI: 10.1002/9781119992806.ch10, Wiley-IEEE Press eBook Chapters, 2011

[Nahavandi 09] Nahavandi, S., Creighton, D., Johnstone, M., and Le, V. T. (2009). *Systems of systems engineering: principles and applications.* CRC press, Taylor and Francis Group, editor: M. Jamshidi. Chapter 16.

[Nitschke 14] C. Nitschke, *Marker-based tracking with unmanned aerial vehicles*, IEEE International Conferenceon on Robotics and Biomimetics (ROBIO), pages 1331 - 1338, 2014

[OpenCV] www.opencv.org

[Ould-Boumama 03] B. Ould-Bouamama, A. Samantaray, M. Staroswiecki, G. Dauphin-Tanguy, *Derivation of constraint relations from bond graph models for fault detection and isolation.* SIMULATION SERIES, 35(2):104109. 2, 63, 107, 132, 2003

[Pall 14] E. Pall, K. Mathe, L. Tamas, L. Busoniu, *Railway track following with the AR.Drone using vanishing point detection*, IEEE International Conference on Automation, Quality and Testing, Robotics, pages 1 - 6, Cluj-Napoca, 2014.

[Raad 14] A. Raad, M. Ayache, A. Abboud, A. Permezel, R. Merzouki, E. Lartigau, *Deformable image tracking of the parotid gland for adaptive radiotherapy application*, 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Pages: 3446 - 3451, 2014

[Robosoft] *www.robosoft.com*

[Sage 01] A. P. Sage and C. D. Cuppan, On the systems engineering and management of systems of systems and federations of systems, Inf.-Knowl.-Syst. Manage., vol. 2, no. 4, pp. 325345, Dec. 2001.

[Sage 07] A. P. Sage and S. M. Biemer, *Processes for system family architecting, design, and integration*, IEEE Syst. J., vol. 1, no. 1, pp. 516, Sep. 2007.

[Sahin 09a] Sahin, F. (2009). *System of Systems Engineering Innovations for the 21st Century.* John Wiley and Sons, editor: M. Jamshidi. Chapter 19.

[Sahin 09b] Sahin, F., Horan, B., Nahavandi, S., Raghavan, V., and Jamshidi, M. (2009). *Systems of systems engineering: principles and applications.* CRC press, Taylor and Francis Group, editor: M. Jamshidi. Chapter 14.

[Samantaray 08] A. K. Samantaray and B. Ould-Bouamama, *Modelbased Process Supervision: A Bond Graph Approach*, New York: Springer-Verlag, 2008.

[Sauser 06] J. Boardman and B. Sauser, *System of systemsThe meaning of of*, inProc. IEEE/SMC, Los Angeles, CA, Apr. 2006.

[Sauser 10] B. Sauser, J. Boardman, and D. Verma, Systomics: *Toward a biology of system of systems*, IEEE Trans. Syst., Man, Cybern. A, Syst., Humans, vol. 40, no. 4, pp. 803814, Jul. 2010.

[Shibasaki 09] Shibasaki, R. and Pearlman, J. S. (2009). *System of Systems Engineering Innova-tions for the 21st Century.* John Wiley and Sons, editor: M. Jamshidi. Chapter 22

[Siciliano 08] B. Siciliano, O. Khatib, *Handbook of Robotics*, Springer-Verlag Berlin Heidelberg, ISBN 978-3-540-23957-4, 2008

[Soyez 15] J-B. Soyez, G. Morvan, R. Merzouki, D. Dupont, *Multilevel Agent-Based Modeling of System of Systems*, IEEE Systems Journal, Volume: PP, Issue: 99, pp. 1 - 12, DOI: 10.1109/JSYST.2015.2429679, 2015.

[Thissen 09] Thissen, W. A. and Herder, P. M. (2009). *System of Systems Engineering Innova-tions for the 21st Century.* John Wiley and Sons, editor: M. Jamshidi. Chapter 11.

[Venkata 12] T. Venkata, S.N. Akshay Uttama Nambi, R. Venkatesha Prasad, I. Niemegeers, *Energy-Harvesting Wireless Sensor Networks*, IEEE Computer Society, pages 31-38, September 2012.

[Wachholder 15] D. Wachholder, C. Stary, *Enabling Emergent Behavior in Systems-of-ystems Through Bigraph-based Modeling*, 10th System of Systems Engineering Conference (SoSE), pages 334 - 339, San Antonio, TX, USA, 2015

[Walkera] *www.walkera.com*

[Wickramasinghe 09] Wickramasinghe, N., Chalasani, S., Boppana, R. V., and Madni, A. M. (2009). *System of Systems Engineering Innovations for the 21st Century.* John Wiley and Sons, editor: M. Jamshidi. Chapter 21.

[Wilber 09] Wilber, G. F. (2009). *System of Systems Engineering Innovations for the 21st Century.* John Wiley and Sons, editor: M. Jamshidi. Chapter 10.

[Witrant 05] E. Witrant, C. Canudas de Wit, D. Georges, *Stabilisation des systemes commandes par reseaux*, Thesis, University of Grenoble, September 2005.