Research in Engineering Design (2023) 34:497–512
https://doi.org/10.1007/s00163-023-00416-4

ORIGINAL PAPER

Towards an integrated design methodology for mechatronic systems

J. A. Vazquez-Santacruz[1] · R. Portillo-Velez[1] · J. Torres-Figueroa[2] · L. F. Marin-Urias[1] · E. Portilla-Flores[3]

Received: 31 May 2022 / Revised: 21 October 2022 / Accepted: 4 March 2023 / Published online: 21 March 2023

Abstract

A design methodology for mechatronic systems is proposed, relying on an integrated framework for engineering solutions with continuous interaction among different fields of knowledge. It incorporates the full life-cycle of mechatronic design, from the problem statement to the attainment of conditions for physical implementations. MBSE domains are addressed into a three dimensional cube shape model where each face is focused on a local analysis through individual and interacting V-models with their own time lines. The design is developed under a centralized tool framework with dependency conditions allowing traceability capabilities in multiple hierarchy levels of analysis for generation, updating and management of information among conceptual analysis, specifications, logical architecture, tasks, detailed design, and manufacturing conditions for production.

Keywords Design methodology · Interdisciplinarity · MBSE · Mechatronics · V-cube

Abbreviations

CAD	Computer aided design
CAM	Computer aided engineering
CFD	Computational fluid dynamics
DFMA	Design for manufacturing and assembling
DT	Digital twin
EAST-ADL	EAST architecture description language
EOS	Engineering operating system
FEM	Finite element method
FMI	Functional mock-up interface
IDEF	Integration definition
MARTE	Modeling and analysis of real time and embedded systems
MDI	Mechatronic design indicator
MDQ	Mechatronic design quotient
MIV	Mechatronics index vector
MMP	Mechatronic multicriteria profile
MOE	Measure of effectiveness
MOP	Measurement of performance
PDM	Product design management
ROS	Robot operating system
SysML	Systems modeling language
TPM	Technical performance measure
UML	Unified modeling language
URDF	Unified robot description format

R. Portillo-Velez, J. Torres-Figueroa, L. F. Marin-Urias and E. Portilla-Flores have contributed equally to this work.

✉ J. A. Vazquez-Santacruz
alejanvasquez@uv.mx

R. Portillo-Velez
rportillo@uv.mx

J. Torres-Figueroa
jtorresf1900@alumno.ipn.mx

L. F. Marin-Urias
luismarin@uv.mx

E. Portilla-Flores
aportilla@ipn.mx

1 Facultad de Ingeniería Eléctrica y Electrónica, Universidad Veracruzana, Av. Ruiz Cortines 455, Boca del Río 94294, Veracruz, México

2 Centro de Innovación y Desarrollo Tecnológico en Cómputo, IPN, Av. Juan de Dios Bátiz S/N, Gustavo A. Madero 07700, Ciudad de México, México

3 Unidad Profesional Interdisciplinaria de Ingeniería, Campus Tlaxcala, Instituto Politécnico Nacional, Av. Guillermo Valle 11, Tlaxcala 90000, Tlaxcala, México

1 Introduction

Mechatronics is evolving from its first appearance in 1969 and multiple methodologies have been developed for multidisciplinary design (Zheng et al. 2014). To render truly mechatronic systems, several engineering disciplines must concur into merged systems, imposing challenges for efficiency, reliability, robustness, and high functionality of the resulting system as a whole. Regardless recent technologies and new concepts for multidisciplinary systems as human-oriented systems and cyberphysical perspectives (Hehenberg and Bradley 2016), the mechatronic approach has always been a reference for synergistic integration becoming an increasingly complex concept (Bradley et al. 2015; Masior et al. 2020) and the design paradigms have been adjusted to provide better engineering solutions. This multidisciplinary nature demands for an updating repository in order to dispose of all information along the design process to ensure consistency throughout the design; therefore, a model based approach (Borchani et al. 2018) seems to be a natural choice by automatizing the interactions among all engineering disciplines involved and providing elements for complex decision making.

There exist several Model Based Systems Enginnering (MBSE) methodologies for product design and development (Huldt and Stenius 2018; Morkevicius et al. 2017) recognized by organizations like INCOSE and OMG; however, in the last decade, MBSE is also getting adopted for mechatronic systems design as Borchani et al. (2018), Qamar et al. (2010) and Vazquez-Santacruz et al. (2019), where methods are mainly based on the V-model (Gausemeier and Moehringer 2002). For instance, Chami and JM (2015) uses it for conceptual design evaluation; in Zheng et al. (2014) and Graessler and Bruckmann (2018), surveys for mechatronic design methods include V-model and extensions; in Graessler and Hentze (2020), a new V-model is proposed for mechatronics and recent technologies design; even more, a *W*-model in the context of MBSE for mechatronic systems has been proposed in Barbieri et al. (2014), in order to integrate multiple phases of design. Recently, in Stark (2022), a study of main MBSE capabilites is addressed as disciplines including *System Environment Analytics* as a problem delimitation, *System Definition and Derivation* for system specifications and high level behavior, *System Interaction Modeling* for numerical analysis behavior, *Systems Lifeciclye Engineering* for product lifecycle analysis, and *Capability and Maturation Matrix* for verification and validation of the MBSE development capabilities; this allows to construct methodologies for complex systems design.

In complement, some modeling languages like *UML, SysML, MARTE, IDEF, EAST-ADL* or *Modelica* have been adopted to support the product/systems design. Particularly, since *SysML* scopes into technical capabilities over the MBSE domains, it is widely considered to describe mechatronic designs as Barbieri et al. (2014), Qamar et al. (2010) for global mechatronic systems modeling; Mhenni et al. (2014), Qamar et al. (2011) for general architectural design; in cases of study as Sell and Tamre (2005) for an unmanned ground vehicle design; Chami et al. (2012) for wheeled and aerial robots modeled under *SysML* to propose a specific framework for conceptual design evaluation, Rahman and Mizukawa (2013) for a mobile robot analysis with *SysML* and interacting numerical tools; Vazquez-Santacruz et al. (2019), for a biped robot developed by using *SysML* to describe the design process into a concurrent V-model interaction; or Friedenthal et al. (2015), Holt and Perry (2018), where the basics of modeling are developed with interesting cases of study.

Since multiple engineering disciplines are addressed in mechatronic complex systems, consistency should be satisfied to avoid conflicting design decisions into a specific methodology. In Lettner et al. (2015), a robotic manipulator is studied by integrating tools and components with consistency checking; in Chen et al. (2018) a verification method is proposed to evaluate consistency between requirements and design; and in Egyed et al. (2018), tools for detecting and checking consistencies are studied.

An interesting approach to guarantee consistency is the implementation of a real time monitoring by virtual models that accurately mirror physical objects, as digital twins (DT) Wu et al. (2021), that have attracted the interest of researchers due of its benefits (Barricelli et al. 2019). The object under study is equipped with various sensors related to vital areas of functionality to produce data on different aspects of performance, such as power output, temperature, weather conditions, and more. This data is then transmitted to a processing system and applied to the digital copy to run simulations and generate potential improvements, all with the goal of generating valuable insights, which can then be applied back to the original physical object. In this sense, a DT connects and expands the digital engineering models for new data streams of new products, machines, and services. Therefore, it is expected digital twins will permanently change product creation and enable new business and value creation models (Barricelli et al. 2019).

This paper introduces an MBSE methodology that serves as a global guideline for mechatronic design under *SysML* modeling capabilities, allowing to deal with complex systems. Though an interesting approach for dealing with complexity is Vahid and Wang (2019) where *agile* concept is adopted to define a methodology, a particular attention has been paid to both, the MBSE-Vitech methodology (Long and Scott 2011) and MBSE capabilities defined in Stark (2022) from where, in an attempt to generalize the problem-solution cycle, it is proposed a whole analysis methodology. To develop an integral and consistent design, MBSE is considered on the methodology for dealing with the high amount of updating information along the process into an interacting framework defined over a three-dimensional V-model.

The contributions of this paper are summarized as follows:

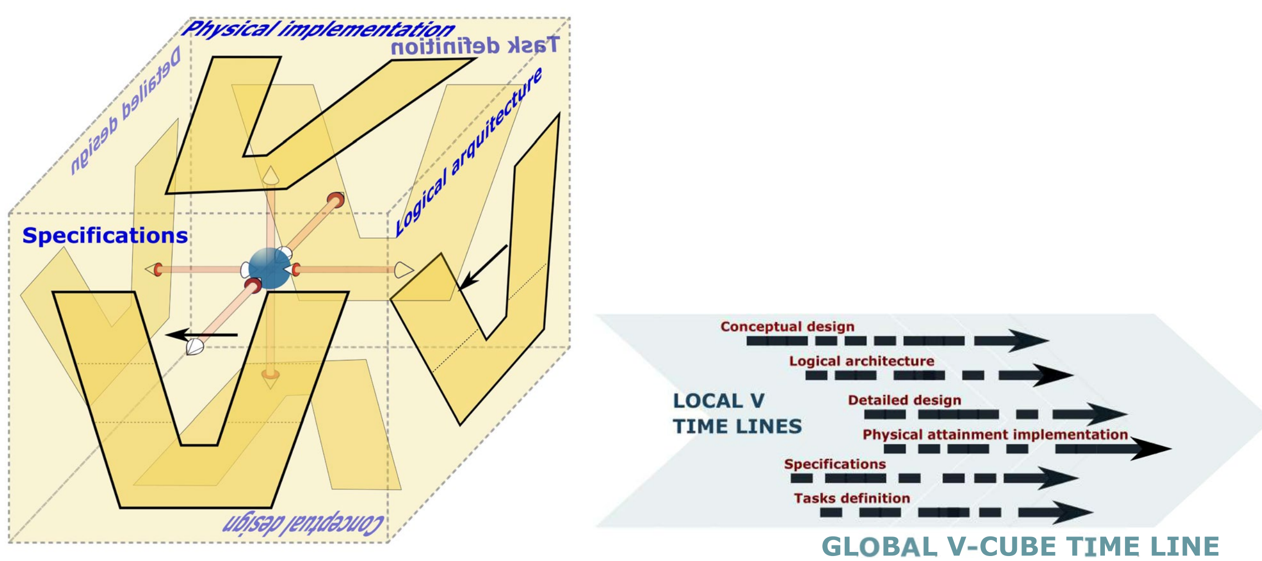
- A methodology for mechatronic systems that covers the full cycle design from the problem identification to the physical implementation of the designed solution over a centralized model.
- A well structured method defined by a cubic distribution based on six V-models focused each on different MBSE domains over the design process.
- A methodology that highlights the hierarchy from high level definitions as a black-box analysis to lower levels for technical implementations as a white-box.
- A full dynamic framework where the interaction among tools with sharing information is more noticeable, even with tools from different engineering disciplines and different hierarchy levels of study. This approach should facilitate the incorporation of the diverse and required engineering tools, including recent technologies, to synergistically converge into an efficient solution for the problem under specific directives.

The paper is structured as follows, the methodology is defined in Sect. 2, where the general three dimensional distribution of the different MBSE domains is described and each face from a cubic distribution is showed by expressing the concurrency along individual processes. In Sect. 3, the interaction among those domains is stated by describing how different tools are incorporated into a general framework into the methodology; in Sect. 4, a case of study illustrates the functionality of the proposed methodology and finally, a discussion and some conclusions are developed in Sect. 6.

2 A V-cube for mechatronic systems design

They are identified in literature four principal domains for MBSE: requirements, behavior, structure and V &V (verification and validation); however, with the aim of defining a methodology to be adapted for any mechatronic system design, the proposed approach includes them from different levels of hierarchy into six particular analysis relying into a cubic distribution where, unlike common methodologies, a full design process also considers the physical conditions for manufacturing and test quality for possible market entry.

Inspired on the traditional V-model (Federal Ministry 1997) and based on the guideline VDI 2206 (Gausemeier and Moehringer 2002), the proposed methodology is based on the MBSE paradigm and a cube shape representation as depicted in Fig. 1a, where each face is focused in a particular analysis by means of a single V-model as follows: the *conceptual design* at the down face is addressed in a high level approach with the system as a black-box (Portillo-Velez et al. 2022) to delimit the problem in order to identify the engineering fields where the solution could be developed; it determines the expertise of the team design for proposing a candidate solution satisfying the stakeholder requirements. The *specifications* face is a behavior and structure high/low level analysis that allows to determine technical system specifications and an operational delimitation for a possible solution. A *tasks definition* face determines the system operations in lower levels of hierarchy and it early nourishes the structural *logical architecture* face, from where system



(a) V-cube distribution method (b) Local timelines and global flow

Fig. 1 Three-dimensional V-method for mechatronic design modules are defined to be analyzed in detail as the design evolves in the *detailed architecture* face where a component level study is developed. The top face, the *physical implementation*, is defined to be analyzed by considering standards for manufacturing and assembling in order to verify the system implementation.

It is important to remark that the faces describe individual processes, however they are not sequential neither independent; moreover, even when a local time line flows as a V-model determines according to its own structure, each face is subject to the information derived from another face, making the design to mature in time as a whole dynamic design process. It must be understood that thought all local V-models concurrently evolve and since they include analysis as either black or white-box, they do not all end or start at the same time; actually since they are dependent each other, the local time lines are discontinuous as the dashed lines attempt to indicate in Fig. 1b. Notice also that the local time lines construct a global one, describing the design evolution in time. This required/provided information among faces produces in consequence a nonlinear flow in time, ending only when all six V-models concur into their final phase.

The six multiple, interacting and concurrent V-models include interaction tools lying in a common framework to update the configuration and parameters of the system under design as the information changes as detailed in Sect. 3. Naturally, depending on the design application, the individual V-models could empathize in a particular engineering discipline and the interaction among faces, given by artifacts and interfaces lying at the center of the cube as a graphical interpretation, gradually nourish the final mechatronic solution. The cubical approach promotes concurrent processes according to the the common framework, an alternative to an Engineering Operating System (EOS), as described in Stark (2022). These studies include some particular sub processes that are set to be adapted according to the nature of the problem and considering interfaces as described in Sect. 2.1 by using interoperating tools in order to automatize the global design process by exchanging data to let each V-model evolute. Naturally, in any situation, the design starts with the conceptual analysis and ends with the physical realization.

The V &V analysis is an important issue to be considered since different testing tools are defined as the design is evolving regarding a particular subject. Each model is proposed to include different tools regarding the engineering analysis and the hierarchy level as described in Sect. 3.

2.1 Individual and concurrent V-models

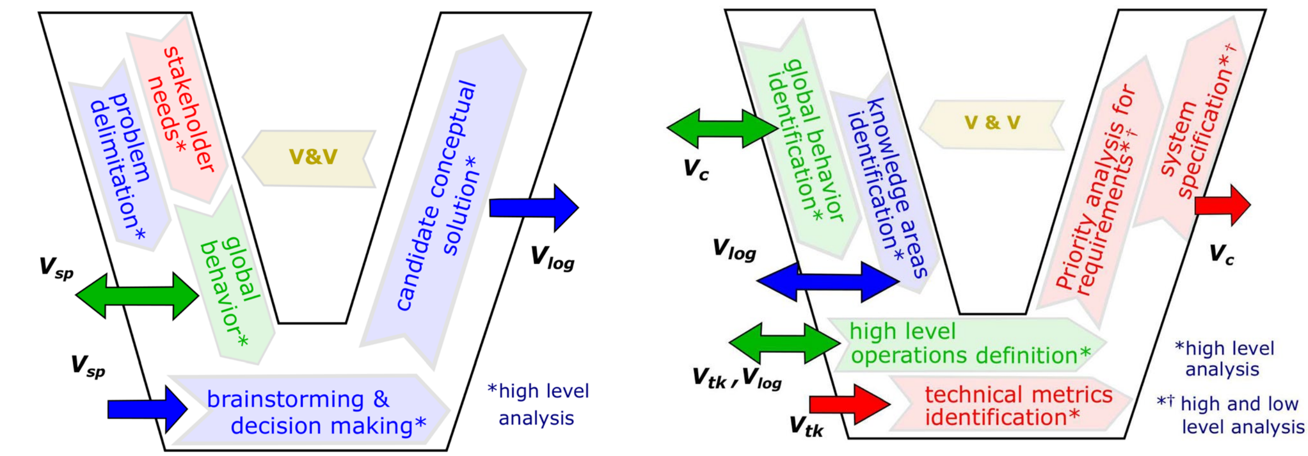
Depending on the MBSE domain to be analyzed, multiple individual and transversal sub processes are well defined into the natural structure of a single V-model with interaction capabilities among others and model dependencies for data management, setting up a complex structure of updating information and enabling traceability on the overall design process. This interdependence strengthens the time lines discontinuities since the data should be generated by the corresponding tool into either co-simulation or model exchange capabilities (Blochwitz et al. 2011) under the addressed problem. This also highlights the concurrency on the mechatronic design process, an important contribution of the proposed methodology, allowing each face to evolve not only for specific design domain, but with a strong dynamic interaction with the others.

A detailed description of each V-model is depicted in Figs. 2, 3 and 4. The interaction among them is described by colored arrows, playing the role of data sharing channels that might be either single or bidirectional by means of both, automated *SysML* diagram models and artifacts such as plug-ins, Functional Mock-up Interfaces (FMI) or files; even more, since there exists a centralized design model where the V-models are structured, the allocation of model elements is quite important to guarantee traceability. In this way, basically, requirements are allocated to specifications, the specifications are allocated to both tasks and operations, while they are allocated to the logical architecture, to be allocated at the same time with elements in the physical and detailed architecture; all of them in a possible bidirectional interconnection.

The local sub processes in each V-model are also differentiated among them by colors, referring to the pillars of MBSE design as: requirements (red), structure (blue), behavior (green) and V &V (amber); this allows to understand the consistent interaction among models to be defined with specific tools. It is important to highlight that these sub processes can be addressed in different levels of hierarchy according to the global status of the design process and as the V &V indicates; this means that the more times the V-model is developed, the higher the maturity level of the design.

2.1.1 Conceptual design

The proposed methodology always begins with the V_c model (Fig. 2a) for the *conceptual design* face where a context analysis delimits the problem based on the stakeholder needs. The aim is to dispose of a global perspective as an engineering problem to propose a conceptual solution with a convenient technical nature, early enabling the concurrency among both, a *logical architecture* of the system in the V_{log} model and the technical specifications in the V_{sp} model. The V_c model also addresses decision making by using mechatronics tools as MDQ, MMP, MIV, MDI, even recent mechatronic indexes (Moulianitis et al. 2018), according to the required criteria in a conceptual level. Tools as the Choquet Integral for example, allows the



(a) Conceptual design V-model (V_c) (b) Specifications V-model (V_{sp})

Fig. 2 V-cube faces: conceptual design and specifications

decision making for multicriteria analysis by considering configurability, dependability, interaction ability, perception, autonomy, motion ability, among others (Katrantzis et al. 2020), from preliminary system specifications. Being analyzed in a high level approach, this face incorporates the four domains of engineering design, and it is an early solution of the problem as a black-box. An implementation of this approach is developed in Portillo-Velez et al. (2022) for a conceptual design of mechatronic system design.

Every local sub process is proposed to contribute with the whole design as the interaction channels suggest, however they can be developed by different tools. In this paper, *SysML* is proposed to be the language to describe them, except for some cases where any other natural tool is suggested as in Table 1. Notice the presence of multiple

Table 1 Suggested tools for local studies: V_c and V_{sp} models

Sub-processes	Suggested tool/artifact	Sub-processes	Suggested tool/artifact
V_c model		**V_{sp} model**	
Stakeholder needs	*SysML* diagrams (Friedenthal et al. 2015): *req*, *Allocation matrix,* Excel sheets	Global behavior identification	*SysML* diagrams: *stm*, *act*, *sd*
Problem delimitation	*SysML* diagrams: *bdd*, *ibd,* Excel sheets	Knowledge areas identification	*SysML* diagrams: *bdd*, *req*
Global performance	*SysML* diagram: *uc*,*act Allocation matrix*	High level operation settings	*SysML* diagrams: *bdd*, *Allocation matrix*
Brainstorming & Decision making	*SysML* diagram: *bdd,* Multicriteria evaluation: Choquet integral (Katrantzis et al. 2020)	Technical metrics identification	Dynamic system model
Candidate conceptual solution	*SysML* diagram: *bdd*, Sketches, CAD	Priority analysis for requirements	Ponderation matrices, Kano model (Chang-Tzuoh et al. 2015), Choquet integral (Katrantzis et al. 2020)
V &V	*SysML* diagram: *SatisfyReqMatrix,* Animations, Simulations (Hazle and Towers 2020)	System Specifications	*SysML* diagram: *req*, *Allocation matrix*
	Model checking & testing (Hazle and Towers 2020)	V &V	Dependency matrices, High level simulations, MDI (Hammadi et al. 2012), MOEs (Kaslow et al. 2018)

allocations tools, which enables the traceability among requirements and other sub processes along the design.

2.1.2 Specification analysis

The *specifications* face developed by the V_{sp} model in Fig. 2b is quite important since it determines the course of the technical design based on information from the *conceptual design* and the current state of the process. It is presented as a technical extension of the stakeholder requirements by combining behavior settings. The suitable engineering areas are identified to organize a preliminary modular design at the *logical architecture* and the behavior domain is technically addressed from the conceptual definition. Notice that this model includes not only a high level analysis but also refers the *tasks definition*, which is a low level approach of the operational definition.

Since multiple technical requirements could be considered to satisfy the user needs, these are weighting to output the system specifications with metric assignments, which shrink to an increasingly narrow range once the V &V for them is stated and the satisfaction of the expected functionality is evaluated by means of indexes of performance under behavior criteria, as the MDI (Hammadi et al. 2012). In this way, as both the *logical architecture* and *detailed design* maturate it is expected that the final specifications gather information from the current centralized model design and they become more specific. This means that technical specifications are not necessarily fixed and they can be updated along the design as global consistency condition from the V &V. In Table 1 are also listed some suggested tools to address the local sub processes of the V_{sp} model.

2.1.3 Low level operational analysis: tasks definition

Once the high level operational analysis is being developed from V_c and V_{sp} models, there are derived tasks of the system in low level approaches with the same characteristics of the *operation* concept by *SysML*, but defined to be accomplished in a subsystem/component level. For example, a sensor monitoring and a control or command execution are tasks derived from a high level operation of path tracking in a manipulator robot; or a voice signal processing is a task of a voice recognition operation in an intelligent machine. Those tasks are set to be developed according to a desired performance at low levels and, unlike operations, technical detailed conditions as references and constraints might be defined for control goals and interaction among tasks.

The objective of this V_{tk} model is the parameterization and full definition of the behavior. The V_{tk} model is depicted in Fig. 3 where it can be noticed that it is totally constructed under the behavior MBSE domain.

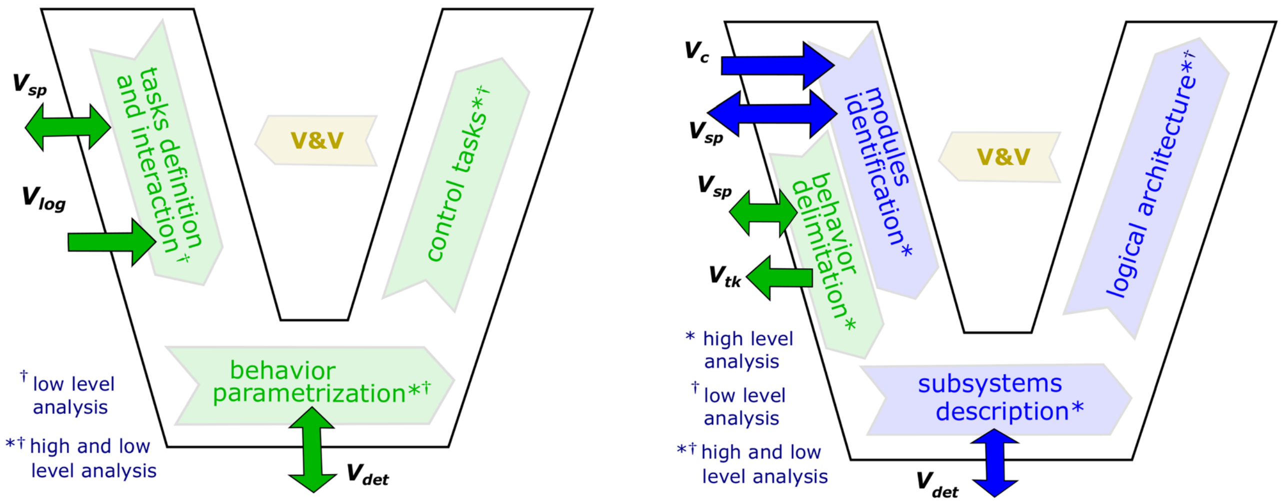
In order to govern the behavior for a given system, control techniques for a particular task, have to be also verified mainly in a low level by tools as suggested in Table 2, that considers numerical simulations linked to *SysML* parametric diagrams to assure a desired performance satisfying the operational requirements and, in consequence, strengthen the *detailed design* face. Notice also that the evolution of this model under the V &V developments, not only affects the tasks definition but also could result in specifications update at the V_{sp} model, exhibiting the codependency among models from different approaches as described in Sect. 3 with some tools for interaction.

2.1.4 Logical architecture

The *logical architecture* analysis is also focused in a combined high/low level design of subsystems and it supports

Table 2 Suggested tools for local studies: V_{tk} and V_{log} models

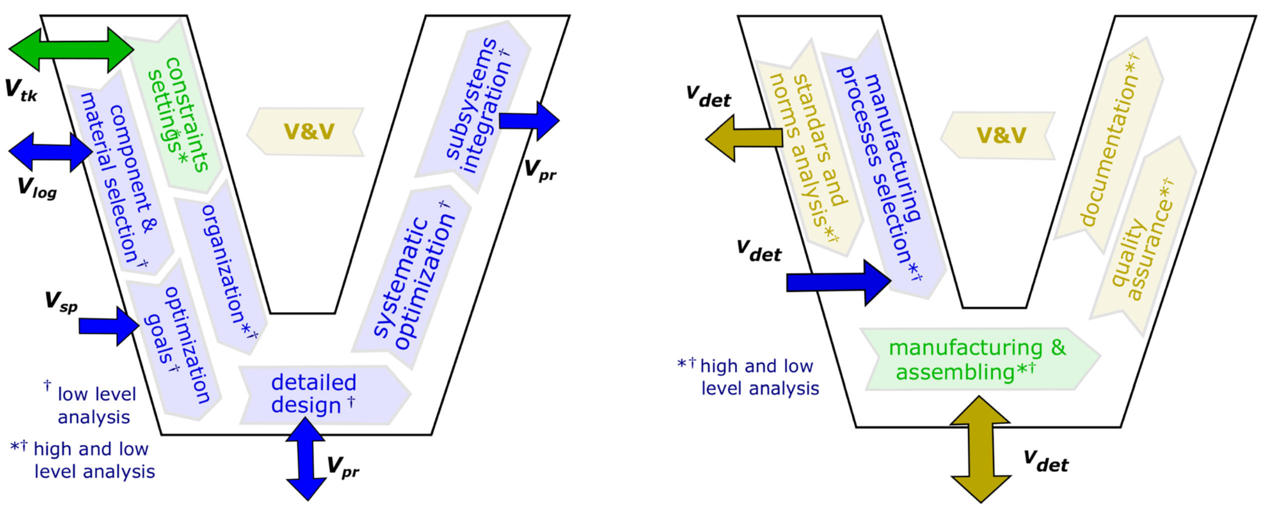
Sub-processes	Suggested tool/artifact	Sub-processes	Suggested tool/artifact
V_{tk} model		**V_{log} model**	
Tasks definition and interaction	*SysML* diagrams (Friedenthal et al. 2015): *bdd*, *sd*, *act*, *allocation matrix*	Modules identification	*SysML* diagrams :*bdd*
Behavior parameterization	*SysML* diagrams: *bdd,* Dynamic models	Behavior delimitation	*SysML* diagrams: *bdd*, *par,* Model checking (Hazle and Towers 2020)
Control tasks	Interfaces (Hause 2018), FMI (Blochwitz et al. 2011), Automation/synchronization, Dynamic/kinematic models	Subsystem description	*SysML* diagrams: *bdd*, *ibd*, *Allocation matrix*
V &V	*SysML* diagrams: *par,* Simulation/animation, MOPs, TPMs, MOEs (Kaslow et al. 2018)	Logical architecture	*SysML* diagrams: *bdd*, *Allocation matrices*
	Model checking & testing (Hazle and Towers 2020)	V &V	Dependency matrices, Model checking (Hazle and Towers 2020)



(a) Task definition V-model (V_{tk})

(b) Logical architecture V-model (V_{log})

Fig. 3 V-cube faces: task definition and logical architecture



(a) Detailed design V-model (V_{det})

(b) Physical implementation V-model (V_{pr})

Fig. 4 V-cube faces: Detailed design and physical implementation

component selection and detailed design. The analysis is developed by the V_{log} model in Fig. 3 which is almost entirely defined under the structure MBSE domain.

An important result from this model is a modular design, which is motivated mainly by the technical characteristics and leads to the subsystem and component definition. Typically, for a mechatronic system, the subsystems are identified as mechanical, electronic, power, control, sensor and actuator systems among others for recent technologies; the goal of this model is precisely the analysis of these subsystems as an extension of the global structure from the V_c model, exhibiting a strong interconnection among technical disciplines in each subsystem by tools as suggested in Table 2.

2.1.5 Detailed design

The *detailed design* is carried out by the V_{det} model depicted in Fig. 4a and it should exhibit a maturity in design since all V models concur in this analysis. The model refers to a refinement in the system, looking for a low level integration of components in consistency with other developments over the process. Though the V_{det} could be predominantly structural, the detailed design also includes behavioral details according to the nature of the problem. There are defined constraints for both behavior and structure, which are suggested to be expressed by *SysML* parametric diagrams as listed in Table 3. These constraints allow the final component selection and detailed subsystems from the logical architecture in V_{log}, including design for manufacturing.

Once a first detailed study is developed, a white-box perspective is clear and a systematic optimization process is suggested to carry out according to a given performance indexed. It is clear that a full optimization over all performance and structural dimensions is a very complex problem, so the required criteria regarding the defined specifications should be identified, leading to a multi-objective problem if required. The detailed achievement is addressed for every single subsystem/module from the corresponding engineering field by considering even the post process for manufacturing, assembly, instrumentation, power and control, as well as the full integration. Naturally, the V_{det} model is developed in a low level approach in order to build up a consistent mechatronic solution, even taking into account considerations for practical implementation as standards and manufacturing methods; however, it is important to highlight that the optimization can be always carried out on any level of hierarchy, this is, at component or subsystem level for different behavior and structural criteria (Sinha et al. 2018).

Since all engineering fields concur on the integration, it implies that much of the design team also interact, and several tools are proposed as suggested in Table 3 as software design tools, optimization techniques, CAD/CAM/CAE tools, and V &V validation strategies as rapid prototyping and performance tests.

2.1.6 Physical implementation

The prototype manufacturing and assembling is developed to verify and validate the design; this phase consists in the physical materialization of the design according to the nature or the system and submitted to quality testing processes to validate both stakeholder requirements and system specifications at the beginning of the design. Recalling that the detailed design considers regulation issues by standards to be complied, the prototyping also evaluates the performance in order to have a certified product with highest usability and, even, market opportunities.

Actually, as it can be depicted in Fig. 4b, the V_{pr} model acts as a V &V process itself since it is focused on the physical realization with functionality assurance. Manufacturing processes (Koc and Ozel 2020) are set according to the concurrent analysis in logical and detailed design, they are selected in both, component and subsystem level in consistency with a suitable assembling from DFMA (Formentini et al. 2022) and other assembling tools. Prototyping is also submitted to quality tests as suggested in Table 3 to finally attain the designed solution in a real environment according to standards from the technical development as ISO, IEEE, IEC, among others (INCOSE 2022).

Clearly, at this phase of the design, the solution is mature enough to develop system testing and it is expected minimum changes towards a final solution. The local time lines at all V-models are then ending as in Fig. 1b, so the global design process is also closing up, leading to commercial stages of the product life cycle.

3 Tools interaction framework

As described, the V-cube demands the integration of multiple tools to analyze specific characteristics of the system with interoperability capabilities. In Fig. 5, it is shown a proposed tool interaction framework where multiple disciplines are involved according to the technical study to be developed on any V-model on the cube, to update the configuration and parameters as the information changes along the process. The full set of tools are grouped regarding their nature in two important classes, *physical environment analysis tools* where the system is verified as a solution of the problem with respective standards and innovative considerations; and the *numerical analysis tools*, where the process is developed under design and control perspectives into a digital centralized model over the behavior, structure and V &V domains. An interesting approach for interoperability conditions can also be considered for an EOS (Stark 2022) where explicitly the human role is centered among artifacts, tools and design hierarchies; however, it is important to clarify that in the framework of Fig. 5 the human activities are implicitly defined in all process design since they are included into the *SysML* centralized model, particularly in the use cases and context diagrams where humans are described by actors.

In general, a mechatronic system is subject to be analyzed by means of numerical tools as simulations for mechanical, electrical, control and connectivity performance supported by digital technologies. Then, in the modeling environment for interoperability conditions where the centralized *SysML* model stands, mathematical and numerical tools are online processing information until they are no longer required and the design has maturated to its last detailed expression.

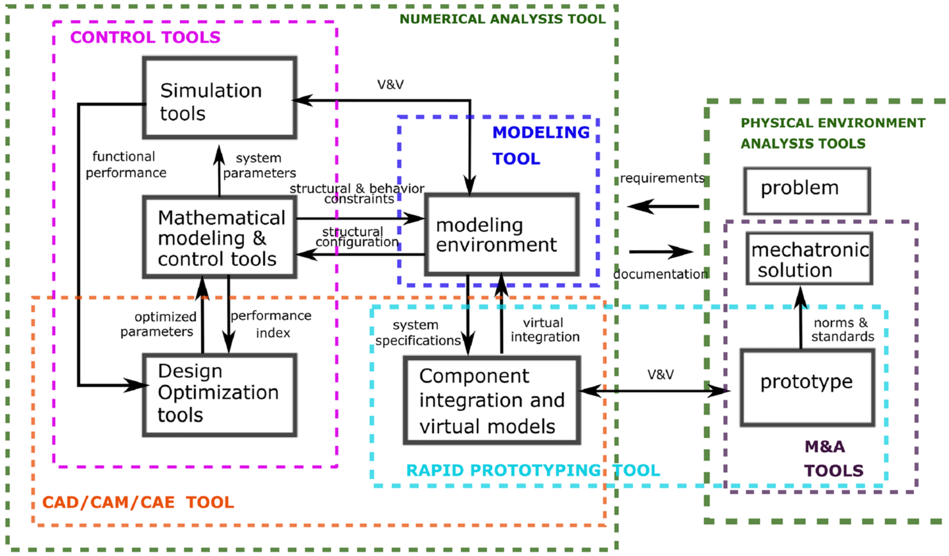


Fig. 5 Global tools interaction into the design process

Table 3 Suggested tools for local studies: V_{det} and V_{pr} models

Sub-processes	Suggested tool/artifact	Sub-processes	Suggested tool/artifact
V_{det} **model**		V_{pr} **model**	
Constraints settings	*SysML* diagrams: *bdd*, *par*, *allocation matrices*	Standards & norms analysis	IEEE, ISO, IEC (INCOSE 2022)
Component selection	*SysML* diagrams: *bdd*, *allocation matrices,* Catalogs	Manufacturing processes selection	*SysML* diagrams: *allocation matrix,* Manuf. processes (Koc and Ozel 2020)
Organization	*SysML* diagrams: *bdd*, *ibd*	Manufacturing & assembling	Machinery, *SysML* diagrams: *allocation matrix*
Detailed Design	Mechanics & Electronics, Control & Communication, DFMA (Formentini et al. 2022)/CAD (Brahmi et al. 2022), FEM, CFD (Smith et al. 2018)	Quality assurance	Quality control testing (Hazle and Towers 2020)
Component optimization	Topology optimization (Vazquez-Santacruz et al. 2019), Heuristic algorithms (Portillo-Velez et al. 2022)	Documentation	Report generation
Subsystem optimization	Multicriteria mechatronic indexes: MDI, MDQ, MIV, MMP (Moulianitis et al. 2018; Katrantzis et al. 2020)	V &V	Testing (Hazle and Towers 2020)
Subsystem integration	*SysML* diagrams: *bdd*, *allocation matrices,* Interfaces & artifacts		
V &V	*SysML* diagrams: *par,* Rapid prototyping, Simulation & animation, Consistency checking (Chen et al. 2018; Egyed et al. 2018)		

In Table 4, some common tools are depicted regarding main fields of knowledge into a mechatronic system: tools for optimization and control are common choices for mathematical modeling analysis of dynamical systems; programming tools for control algorithms are also important complements from the information technology; electrical analysis could be developed for power and electronic instrumentation simulations, including technologies for IoT applications; even artificial intelligence is numerically implemented to produce a hybrid between control theory and heuristic algorithms for decision making; spatial configuration for physical dynamics including thermal/flow conditions is able to be studied into the framework with CAD/CAM/CAE tools, generating increasingly solid

Table 4 Common tools for mechatronic systems analysis

Mechanical	Electrical	Control	Requirements
Autodesk Fusion 360	PSIM	MATLAB/Simulink	Inflectra: spiratest-spirateam
CATIA	Proteus	SciLab - Xcos	Jama Software
SolidWorks	Autodesk Eagle	Labview	Orcanos
Autodesk Inventor	Ansys	GNU Octave	IBM ERM DOORS Next
Pro-E/CREO	NI Multisim	OpenModelica	Accompa
Siemens NX	LTSpice	Python	Visure Requirements
Autocad Mechanical	Modelsim	ROS	Caliber
Ansys	Altium	Gazebo	ReqSuite
Solid Edge	DesignSpark		Pearls
COMSOL Multiphysics	Cadstar		Perforce Helix RM

conditions for prototyping with an adequate performance until the final solutions.

Notice in Fig. 5 that the modeling environment plays the central role where each sub processes from the V-models converges by storing data from tools that nourish the mechatronic solution; all decisions and modifications fall into the centralized model which dictates the evolution of the system from the contribution of all engineering fields of the design team until the final physical implementation.

Though it is proposed a general interaction framework, it is aimed to be automatized, especially if different subsystems studies are built by different tools. A formal system integration can be developed according to standard interoperability tools; however, interfaces as the FMI (Blochwitz et al. 2011), a semantic mediation container SMC (Shani et al. 2016), or a semantic web technology SWT (Bone et al. 2018) for recent technologies, could serve as base for the implementation. Data exchange and visualization, by using CAD, CAM, CAE, or PDM tools can also be considered by using JL standard according to Beckers et al. (2016) for data management; for CAE applications, the MpCCI standard interface supported by *SysML* and Modelica for multidiscplinary modeling in mechatronic systems is considered in Lefevre et al. (2017). An interesting interoperability approach is also presented in Hammadi et al. (2016) for integrate a complex *SysML* model and a 3D modeling environment to reconfigure designs. In Stark (2022), a study of different data and model standards for interoperablity into a V-model is addressed for different phases of design and based on White paper (2019).

In the context of MBSE, a DT renders the challenge of a holistic model integrating several aspects from different fields of engineering to provide a feasible architecture for interoperability (Göllner et al. 2022). Moreover, a DT might play a fundamental role for *SysML* system models integrating a physical counterpart, bidirectional orientation and real-time data capabilities. A DT provides a linking support of physical objects with virtual environments; it is however clear that recent studies points out that, despite its complexity, a DT is becoming a feasible alternative to consider within *SysML* models (Wilking et al. 2022).

In the proposed methodology, *SysML* allows the model construction of the design process by using commercial tools as either *Cameo Sytems Modeler* (CMS) or *Magic Draw* from *No Magic*; *Enterprise Architec* from *Sparkx*; *Rational Rhapsody* from *IBM*; *Visual Paradigm*; as well as open source tools as *Modelio*, *Papyrus*, *Capella*, etc.; however, though recent efforts have been made (Khandoker et al. 2022), it is clear that there is no rule to select the best tool for any application. Since technology and software tools are in permanent evolution, the best tool is always subject to fit with those complementary tools to guarantee an even better performance in the design process; however, interoperability is accomplished by a convenient tools selection from different types of analysis to satisfy functionality and usability over the application in order to construct a detailed model from different technical domain perspectives (Rashid et al. 2015).

4 Case of study: A 12-DOF biped robot

The design of a biped robot is described with the aim of validate the proposed MBSE approach. It can be verified that the V-cube is not rigid, which means that the nature of the problem adopts the V-models to perform local sub processes and the required tools for interoperability, according to the needs. Figure 6 depicts an overview of the framework over the guideline where it can be observed the global interaction of tools and data sharing among the V-models. This framework is made up of five main modules: a *SysML* model repository based on CMS in yellow, a CAD/CAM model by Autodesk Inventor in orange, a Matlab environment for dynamics and optimization in green, a ROS ecosystem for control and validation tests in blue, and the module where the Manufacturing and Final Assembly (M &A) of the robot, in purple, is carried out.

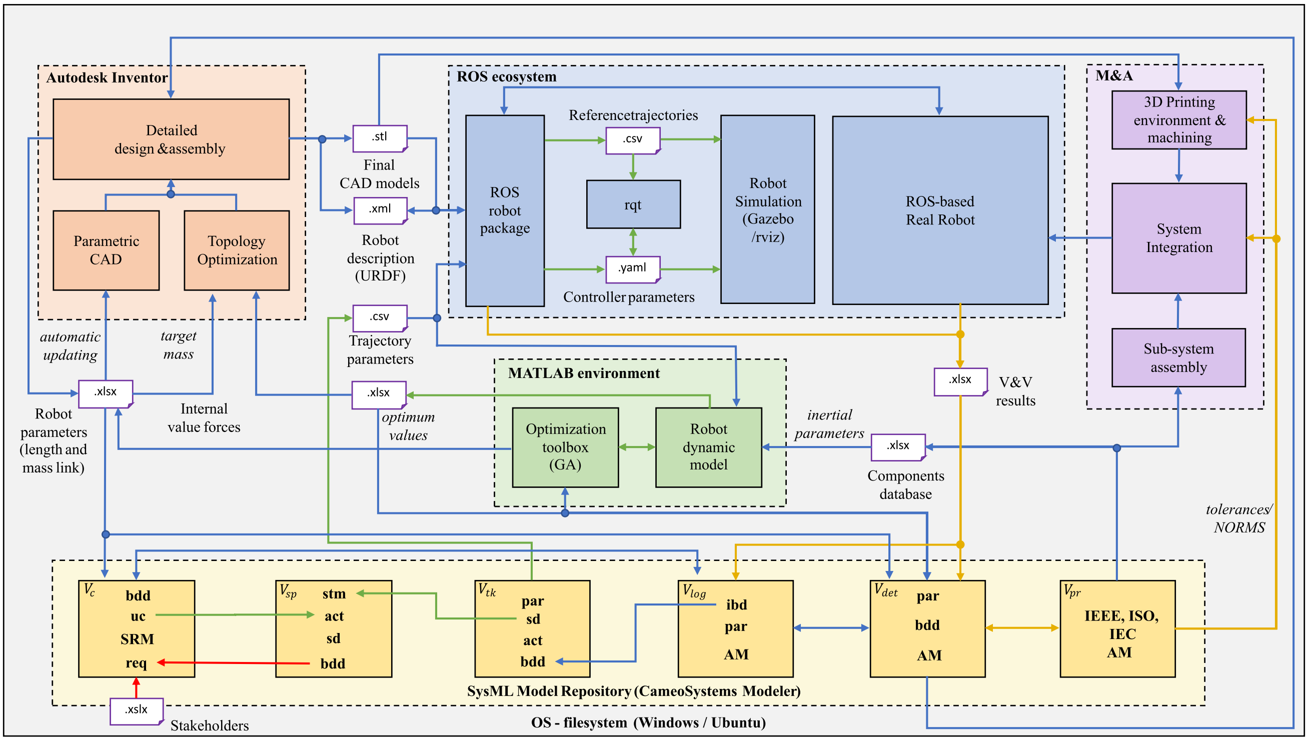


Fig. 6 V-cube methodology on a biped robot design

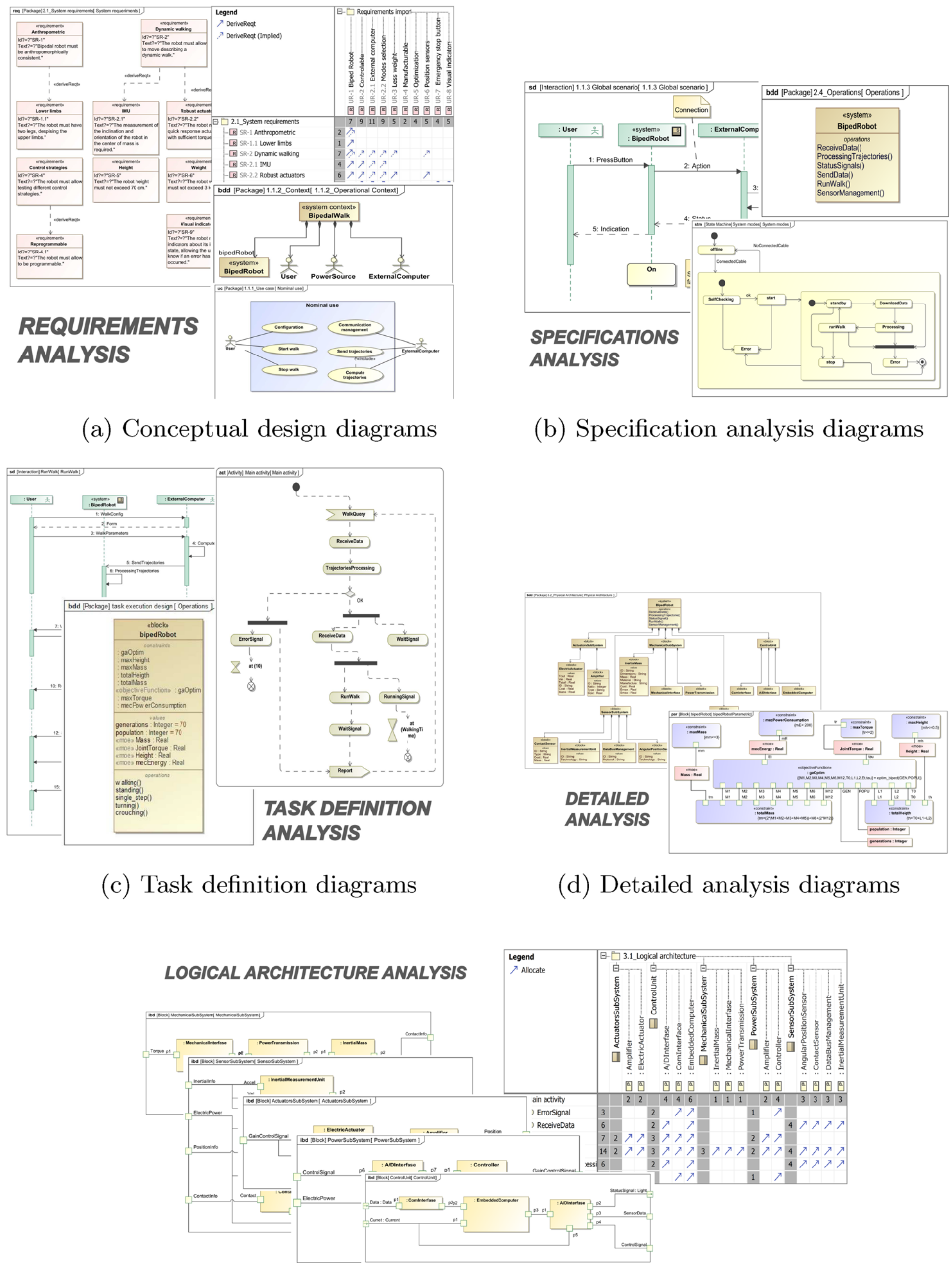
The centralized model is structured based on *SysML* and it is depicted at the bottom of Fig. 7 where some relevant diagrams are depicted according to the repository from Fig. 6 for each V-model and following suggested artifacts as in Tables 1, 2 and 3. These diagrams are associated among them by means of parameter and values which are getting updated from parametric diagrams as a response of a automated interoperation among CMS and Matlab for structural optimization based on behavior dynamics; as well as a manual parameters updating with parametric CAD on Inventor and the behavior analysis extension into a ROS environment with bidirectional flow of information with parametric diagrams. This interaction is possible by using different interfaces among tools as the files and plug-ins (stl, xlsx,cvs, yaml, etc.) specified for each interconnection and depicted in Fig. 6.

The V-cube methodology is then applied to a biped robot design with anthropomorphic characteristics and punctual requirements listed on a Excel sheet and translated into a *SysML* environment. Thereby the centralized model is connected via plug-in with Matlab environment, through a *par* diagram from the V_{det} allowing co-simulation to obtain, by using a heuristic numerical optimization strategy based on genetic algorithms (GA), the optimal parameters of mass and length of the robot links along a walking task. When the optimization process is finished, the optimal parameters are returned to the *par* diagram in CSM for constraint verification on the V_{det} model, and they are stored in the *Robotparameters.xlsx* file, which in turn has a direct link to the CAD robot model designed in Autodesk Inventor, allowing a continuous updating design in real time.

A second optimization stage is carried out to define the final shape of the robot links by topological optimization in Autodesk Inventor. The target mass for each link is the mass obtained by the optimization from the Matlab GA tool, while the internal forces to maximize stiffness are obtained automatically from the *InternalValuesForces.xlsx* file, which imports the results from the dynamic model simulation in Matlab. These results are also automatically updated in the centralized model in CMS, to perform the verification and validation in real time at the V_c and V_{det} models by allocation and verification matrices where warnings appear in consistency is violated. Once the kinematic and inertial parameters of the robot have been defined, a detailed design and assembling process is carried out in Autodesk Inventor, taking into account the design constraints and standards defined in the V_{pr} model for integration. The validated links are finally exported as *finalCADmodel.stl* files to develop the manufacturing process, defined in this case with 3D printing and machining only when the V &V is developed.

For the robot dynamic simulation into the V_{tk} and the V_{det} models, a ROS ecosystem is constructed where kinematic and dynamic description of the system is generated through the *RobotDescription.xml* file, an URDF that allows to

Fig. 7 Relevant *SysML* diagrams from global design model



(a) Conceptual design diagrams

(b) Specification analysis diagrams

(c) Task definition diagrams

(d) Detailed analysis diagrams

(e) Logical arquitecture diagrams

perform model exchange between Autodesk Inventor and Gazebo simulator. The file, *RobotDescription.urdf*, is automatically exported and it uses the latest versions of the CAD models in *.stl* format, which becomes at a time, the optimal one.

From the V_{tk} model, a file called *TrajectoryParameters.csv* is exported with the encoded information of foot placements for walking according to the width and length of the dynamic step established under the constraints from the *SysML par* diagram. By considering this information, different ROS-based nodes calculate the desired trajectories for the walking task in the robot, that are automatically stored in the file *Referencetrajectories.csv* which, together with the values established for the control parameters in the *ControllerParameters.yaml* file, are used in the dynamic reconfiguration tool available in the ROS ecosystem for tuning the control gains in real time.

As a V &V test for behavior at the V_{tk} model, in Fig. 8 it is shown a successful performance with control designed throughout a stable dynamic walking in Gazebo.

When the control parameters tuning is done, the results obtained for the trajectory tracking error are stored in the file

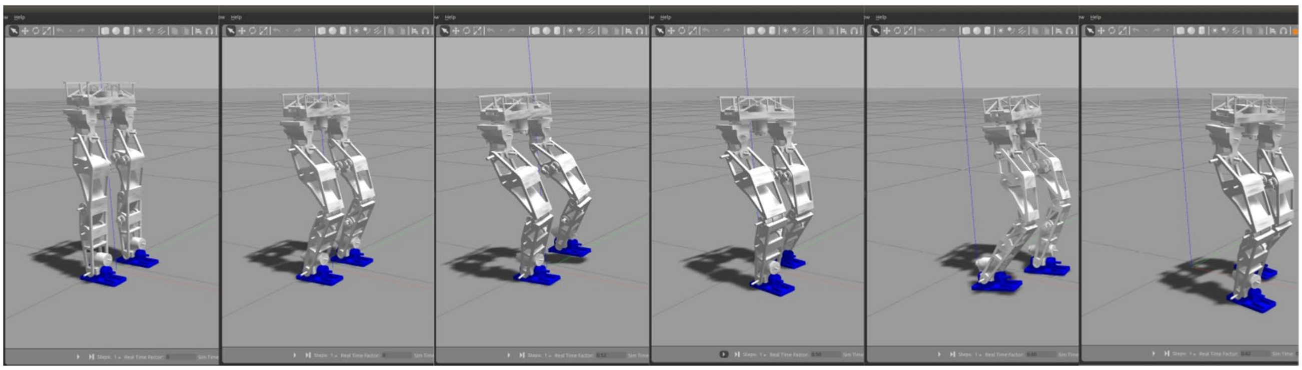
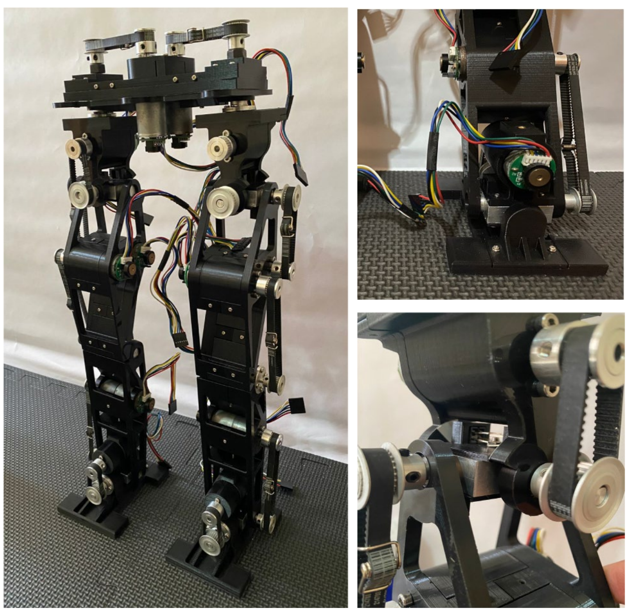


Fig. 8 Verification test for walking performance



(a) Biped robot prototype (b) Prototype detail

Fig. 9 View of the final robot prototype

V &Vresults.xlsx, which is linked to *par* diagrams at both, V_{tk} and V_{log} models, with the aim of performing a verification according to mechatronic performance criteria by using MDI, based on the system specification and the stakeholders needs at V_c and V_{sp} models.

If the results obtained in simulation are satisfactory, in the Manufacturing and Assembly (M &A) stage, the previous *finalCADmodel.stl* file with the final parameters are manufacture, while the various electronic, mechanical and control components are assembled, carrying out both manufacturing and subsystem assembly, to finally integrate the whole system.

The final prototype that is provided by the V_{ps} model is depicted at the Fig. 9, where components and modules are already integrated as a validated system to perform walking in a real environment. In Fig. 9a the global prototype is depicted while in Fig. 9b, some detailed assembling for sensors, transmission and electronic instrumentation is shown.

5 Discussion of the case of study under the methodology

A synergistic methodology for the design of the biped robot is applied as an integrated framework under the MBSE paradigm. Even though the V-cube method dictates specific analysis by means of V-models and sub processes, most of the artifacts and tools might be selected according to the nature of the problem, for example, multiphysics tools as Gazebo, CAD/CAM as Inventor, control design as Matlab, etc. This might be an important labor since it selection should be based on several criteria such as technical domain, high performance capabilities and compatibility in order to dispose of an automated framework. The methodology, however, dictates a wider cycle of the design, compared with literature, since it includes the manufacturing and assembling with data sharing to other stages of the process; even more, it includes several sub processes including structural and behavioral optimization in different hierarchy levels to deal with complex systems, for example structural and topology optimization. Additionally, this proposal provides flexibility to be adapted from solutions ranging from the more basic conceptual design process up to product prototyping in mechatronic engineering, not necessarily including the whole V-model faces this is, it could be used only for conceptual or behavior design considering available required data. Though this implementation is developed with a semi-automated interaction, a huge challenge is the fully automated framework since interfaces for data exchange in general are defined among particular analysis as CAD-requirements, CAD-Data management, physics-CAD, etc., but a global interconnection depends on manual interfaces, even on commercial compatibility among tools. The methodology serves however as a clear guideline where principal pillars of MBSE are addressed and recommended to be modeled as a part of the global centralized model that evolves with all studies in a concurrent way.

6 Conclusions

One of the main contributions of this work consists in the definition of a global framework where all tools interact according to a three-dimensional perspective where the MBSE domains of analysis for a mechatronic system are distributed in a cubical shape and serves as a guideline for design. This is a particular aim in MBSE methodologies which is achieved by enhancing the application of common engineering tools for mechatronic design that demands an automation for manage high amount of information.

The modeling of the system is showed by means of the most representative diagrams from *SysML*, however the software tools allow to define a more complex framework where high amount of information is evolving with the aim of designing a final prototype without inconsistencies among different levels of integration.

It is important to remark that the proposed V-cube describes a generalized design methodology that allows individual V-models to evolve according to its own needs, producing different progress rates at each face but interacting all of them to define a global flow of the process design. However, since the V &V eventually enables cyclic analysis into a particular V-model, the final process of design will end only when all of them concur at the top of the right side. This is a remarkable difference compared to other approaches where a sequential process is evaluated and a partial concurrency is only included for functional and structural domains. In addition, in this paper it should be understood that operational analysis is defined on multiple levels of design referring as tasks to lower levels of study describing a specific action that a system could develop, whereas high level operations are the intrinsic behavior to enable a system to perform those tasks. This high and low level facility of analysis into the global design, represents another contribution of the methodology, since multiple developments are concurrently defined into the global framework with the corresponding sharing channels.

Though in recent years, multiple concepts for complex systems have been emerging, the V-cube approach introduced in this paper, could be easily adapted since the domains of analysis are in general the same, however, in order to be adapted, the local sub processes of the V-models might be specified for particular analysis regarding the nature of the system.

References

Barbieri G, Fantuzzi C, Borsari R (2014) A model-based design methodology for the development of mechatronic systems. Mechatronics 24(7):833–843

Barricelli BR, Casiraghi E, Fogli D (2019) A survey on digital twin: Definitions, characteristics, applications, and design implications. IEEE Access 7:167653–167671. https://doi.org/10.1109/ACCESS.2019.2953499

Beckers R, Giese S, Pfouga A, Stjepandic J (2016) Interoperability and visualization of complex products based on jt standard. Curitiba. https://doi.org/10.3233/978-1-61499-703-0-828

Blochwitz T, Otter M, Arnold M, Bausch C, Clau C, Elmqvist H, Junghanns A, Mauss J, Monteiro M, Neidhold T, Neumerkel D, Olsson H, Peetz J, Wolf S (2011) The functional mockup interface for tool independent exchange of simulation models. In: Proceedings of the 8th International Modelica Conference, Dresden, pp. 105–114

Bone M, Blackburn MR, Kruse B, Dzielski J, Hagedorn T, Grosse I (2018) Toward an interoperability and integration framework to enable digital thread. Systems 6(4):46. https://doi.org/10.3390/systems6040046

Borchani M, Ammar R, Hammadi M, Choley J, Yahia N, Barkallah M, Louati J (2018) Mechatronic system design using model-based systems engineering and set-based concurrent engineering principles. In: 12th France-Japan and 10th Europe-Asia Congress on Mechatronics, pp. 32–38. https://doi.org/10.1109/MECATRONICS.2018.8495824

Bradley D, Russell D, Ferguson I, Isaacs J, MacLeod A, White R (2015) The internet of things: The future or the end of mechatronics. Mechatronics 27:57–74

Brahmi R, Belhadj I, Hammadi M, Aifaoui N, Choley J (2022) CAD-MBSE. Appl Sci 12:566. https://doi.org/10.3390/app12020566

Chami M, JM B (2015) Towards an Integrated Conceptual Design Evaluation of Mechatronic Systems: The SysDICE Approach. In: Int. Conf. on Computational Science (ICCS 2015), pp. 650–659

Chami M, Ammar HB, Voos H, Tuyls K, Weiss G (2012) A Nonparametric Evaluation of SysML-based Mechatronic Conceptual Design. In: 12th France-Japan and 10th Europe-Asia Congress on Mechatronics, pp. 51–58

Chang-Tzuoh W, Ming-Tang W, Nien-Te L, Tien-Szu P (2015) Developing a kano-based evaluation model for innovation design. Math Probl Eng. https://doi.org/10.1155/2015/153694

Chen R, Liu Y, Zhao J, Ye X (2018) Model verification for system design of complex mechatronic products. Syst Eng. https://doi.org/10.1002/sys.21470

Egyed A, Zeman K, Hehenberger P, Demuth A (2018) Maintaining consistency across engineering artifacts. Computer 51(2):28–35. https://doi.org/10.1109/MC.2018.1451666

Federal Ministry of Defense, Federal Office of the Bundeswehr for Infor- mation, Management and Information Technology and Federal Ministry of the Interior: The V-model. In: Development Standard for IT-Systems of the Federal Republic of Germany. KBSt, vol. 27 (1997)

Formentini G, Rodriguez B, N, Favi C, (2022) Design for manufacturing and assembly methods in the product development process of mechanical products: a systematic literature review. Adv Manuf Technol, Int J. https://doi.org/10.1007/s00170-022-08837-6

Friedenthal S, Moore A, Steiner R (2015) A Practical Guide to SysML. Third edn. Elsevier, Amsterdam, The Systems Modeling Language

Gausemeier J, Moehringer S (2002) VDI 2206- A new guideline for the design of mechatronic systems. IFAC Proced Vol 35(2):785–790

Graessler I, Bruckmann T (2018) V-Models for Interdisciplinary Systems Engineering. In: 15th Int. Design Conference, pp. 747–756. https://doi.org/10.21278/idc.2018.0333
Graessler I, Hentze J (2020) The new V-model of VDI 2206 and its validation. at - Automatisierungstechnik 68, 312–324. https://doi.org/10.1515/auto-2020-0015
Göllner D, Rasor R, Anacker H, Dumitrescu R (2022) Collaborative modeling of interoperable digital twins in a sos context. Procedia CIRP 107:1089–1094. https://doi.org/10.1016/j.procir.2022.05.113
Hammadi M. Choley J, Penas O, Riviere A, Louati J, Haddar M (2012) A new multi-criteria indicator for mechatronic system performance evaluation in preliminary design level. In: 9th France-Japan and 7th Europe-Asia Congress on Mechatronics (MECATRONICS) / 13th Int. Workshop on Research and Education in Mechatronics (REM), pp. 409–416. IEEE, Paris. https://doi.org/10.1109/MECATRONICS.2012.6451041
Hammadi M, Choley J, Said M, Kellner A, Hehenberger P, (2016) Systems engineering analysis approach based on interoperability for reconfigurable manufacturing systems. In, (2016) IEEE International Symposium on Systems Engineering (ISSE). IEEE, Edinburgh. https://doi.org/10.1109/SysEng.2016.7753179
Hause M (2018) Systems interface management with mbse: from theory to modeling to reality. INCOSE Inter Symp 28:1012–1026. https://doi.org/10.1002/j.2334-5837.2018.00530.x
Hazle A, Towers J (2020) Good practice in MBSE model verification and validation. In: INCOSE UK Annual Systems Engineering Conference (ASEC). INCOSE, Virtual (2020)
Hehenberg P, Bradley D (2016) Mechatronic Futures. Challenges and Solutions for Mechatronic Systems and Their Designers. Springer, Switzerland
Holt J, Perry S (2018) SysML Syst Eng: Model-based Approach, 3rd edn. The Institution of Engineering and Technology, IET Digital Library
Huldt T, Stenius I (2018) State of practice survey of model-based systems engineering. Syst Eng. https://doi.org/10.1002/sys.21466
INCOSE (2022) Systems Engineering Standards. https://www.incose.org/about-systems-engineering/se-standards
Kaslow D, Ayres B, Cahill P, Hart L (2018) A model-based systems engineering approach for technical measurement with application to a cubesat. In: 2018 IEEE Aerospace Conference, pp. 1–10. IEEE, Big Sky (2018). https://doi.org/10.1109/AERO.2018.8396443
Katrantzis E, Moulianitis V, Miatliuk K (2020) Conceptual design evaluation of mechatronic systems. In: Azizi, A. (ed.) Emerging Trends in Mechatronics. IntechOpen, Rijeka. Chap. 2. https://doi.org/10.5772/intechopen.88643
Khandoker A, Sint S, Gessl G, Zeman K, Jungreitmayr F, Wahl H, Wenigwieser A, Kretschmer R (2022) Towards a logical framework for ideal MBSE tool selection based on discipline specific requirements. J Sys Software 189:111306
Koc M, Ozel T (2020) Modern Manuf Proces. John Wiley and Sons, Onlinelibrary
Lefevre J, Charles S, Bosch-Mauchand M, B, E, (2014) Multidisciplinary modelling and simulation for mechatronic design. J Design Res 12(2):127–144
Lettner D, Hehenberger P, Nöhrer A, Anzengruber K, Grünbacher P, Mayrhofer M, Egyed A (2015) Variability and consistency in mechatronic design. Concurr Eng 23(3):213–225
Long D, Scott Z (2011) Primer Model-based Syst Eng, 2nd edn. Vitech, Blacksburg
Masior J, Schneider B, Kürümlüoglu M, Riedel O (2020) Beyond model-based systems engineering towards managing complexity. Procedia CIRP 91:325–329. https://doi.org/10.1016/j.procir.2020.02.183
Mhenni F, Choley J, Penas O, Plateaux R, Hammadi M (2014) A SysML-based methodology for mechatronic systems architectural design. Mechatronics 28:218–231
Morkevicius A, Aleksandraviciene A, Mazeika D, Bisikirskiene L, Strolia Z (2017) MBSE grid: A simplified sysml based approach for modeling complex systems. INCOSE Int. Symposium 27(1):136–150
Moulianitis V, Zachiotis GD, Aspragathos N (2018) A new index based on mechatronics abilities for the conceptual design evaluation. Mechatronics 49:67–76. https://doi.org/10.1016/j.mechatronics.2017.11.011
Portillo-Velez RJ, Burgos-Castro IA, Vazquez-Santacruz JA, Marin-Urias LF (2022) Integrated conceptual mechatronic design of a delta robot. Machines 10(3):45
Qamar A, Wikander J, During C (2011) Designing Mechatronic Systems: A Model Integration Approach. In: International Conference on Engineering Design, ICED11
Qamar A, Törngren M, Wikander J, During C (2010) Integrating multi-domain models for the design and development of mechatronic systems. In: 7th European Systems Engineering Conference. INCOSE, Stockholm
Rahman MAA, Mizukawa M (2013) Modeling and Design of Mechatronics System with SysML, Simscape and Simulink. In: IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics, pp. 1767–1773
Rashid M, Anwar M, Khan A (2015) Towards the tools selection in model based system engineering for embedded systems A systematic literature review. J Syst Software. https://doi.org/10.1016/j.jss.2015.04.089
Sell R, Tamre M (2005) Integration of V-model and SysML for Advanced Mechatronics System design. In: Int. Workshop on Research & Education in Mechatronics
Shani U, Jacobs S, Wengrowicz N, Dori D (2016) Engaging ontologies to break MBSE tools boundaries through semantic mediation. In: Conference on Systems Engineering Research
Sinha A, Malo P, Deb K (2018) A review on bilevel optimization: From classical to evolutionary approaches and applications. IEEE Trans Evolut Comput 22(2):276–295. https://doi.org/10.1109/TEVC.2017.2712906
Smith MJ, Jacobson KE, Afman JP (2018) Towards certification of computational fluid dynamics as numerical experiments for rotorcraft applications. Aeronaut J 122(1247):104–130. https://doi.org/10.1017/aer.2017.118
Stark R (2022) Virtual Product Creation in Industry. The Difficult Transformation from IT Enabler Technology to Core Engineering Competence, Springer, Berlin
Vahid S, Wang S (2019) Munich Agile MBSE concept (MAGIC). In: Wartzack, S., Schleich, B. (eds.) Proceedings of the Design Society: International Conference on Engineering Design, pp. 3701–3710. Cambridge University Press, Munich. https://doi.org/10.1017/dsi.2019.377
Vazquez-Santacruz JA, Torres-Figueroa J, Portillo-Velez R (2019) Design of a human-like biped locomotion system based on a novel mechatronic methodology. Concurr Eng 27(3):249–267
White paper prostep ivip association (2019) collaborative systems engineering on the basis of engineering it standards., 1–12
Wilking F, Sauer C, Schleich B, Wartzack S (2022) Integrating machine learning in digital twins by utilizing sysml system models. In: 2022 17th Annual System of Systems Engineering Conference (SOSE), pp. 297–302. IEEE, Rochester (2022)
Wu Y, Zhang K, Zhang Y (2021) Digital twin networks: A survey. IEEE Internet Things J 8(18):13789–13804. https://doi.org/10.1109/JIOT.2021.3079510

Zheng C, Bricogne M, Duigou J, Eynard B (2014) Survey on mechatronic engineering: A focus on design methods and product models. Adv Eng Inform 28(2014):257–341