A. GLOSARIO EXTENDIDO DE TÉRMINOS EN SYSML/UML

El siguiente glosario ha sido extraído de [22].

Abstraction [other]: The result of a way of mapping things that emphasizes certain aspects while omitting others.

Abstraction dependency [UML]: The mapping between model elements onto various abstraction levels.

AcceptEventAction [UML]: An elementary ¬ action that receives an ¬ event, or accepts the arrival of a signal.

Action [UML]: An action is an elementary executable step within an activity.

Activity [UML]: An activity describes the coordinated sequencing of elementary actions. The sequencing can be parallel or synchronized, or split and recomposed on the basis of conditions.

Activity diagram [UML]: A diagram that depicts behavior associated with activities using input/output and control flow. It is the visualization of an ¬ activity.

Activity edge [UML]: An activity edge is an abstract class for directed connections between two ¬ activity nodes. We distinguish between ¬ object flow edges and ¬ control flow edges.

Synonym: Edge.

Activity final node [UML]: The activity final node terminates the entire ¬ activity as soon as a single flow arrives at the node.

Activity parameter node [UML]: An activity parameter node is an ¬ object node for inputs and outputs to activities.

Activity partition [UML]: An activity partition is a kind of activity group for identifying activities that have some characteristic in common.

Actor [UML]: An actor is a role that interacts with the system. This role can be played by a user or any other system. An actor is external to the system.

Actuator [SYSMOD]: An actuator is a particular ¬ external system that serves another system to influence its environment.

Aggregation [UML]: Describes a ¬ class as an aggregate and specifies a whole-part relationship between the aggregate (whole) and a component part. In contrast to a ¬ composition, the aggregate is not responsible for its parts.

Allocate activity partition [SysML]: A special ¬ activity partition that allocates each ¬ action within the partition to the structure that is represented by the partition.

Allocation [SysML]: An abstract relationship between elements of different types or on different levels. It allocates a target element to one or more source elements.

Association [UML]: An association is a structural relationship between two ¬ classes.

Association block [SysML]: An association block describes the structural properties of an ¬ association.

Association class [UML]: An association class unifies the properties of an ¬ association with those of a ¬ class.

Asynchronous message [UML]: A ¬ message where its sender does not wait for the receiver to complete processing it, but continues with its flow immediately upon sending it.

Atomic flow port [SysML]: An ¬ object flow port that is not typed by an ¬ object flow specification, but which is a ¬ system component or a ¬ data type.

Attribute [UML]: An attribute defines a structural property of a class. This description consists of visibility, name, type, and a multiplicity.
B

Behavior diagram [UML]: A diagram used in →SysML and →UML to describe dynamic aspects. Block [SysML]: A modular unit that describes the structure of a system or element. Block definition diagram [SysML]: A diagram that shows the definition of →blocks and their relationships (e.g., a →composition).

Boundary system [SYSMOD]: A special →external system that provides an interface to another external system.

C

Call event [UML]: →Event.

CallBehaviorAction [UML]: An elementary action that invokes a behavior element, such as an activity, directly.

CallOperationAction [UML]: An elementary action that invokes an →operation.

Change event [UML]: →Event.

Choice pseudo state [UML]: The decision is a →pseudo state which, when reached, evaluates conditions to select the next →transition.

Class [UML]: A class describes the structure and behavior of →objects, which have identical characteristics and semantic. The structure is described by →attributes, while the behavior is described by →operations.

Class diagram [UML]: A diagram that shows →classes and their relationships. This diagram is available in UML only. SysML uses a different form called →block definition diagram.

Combined fragment [UML]: A combined fragment describes an expression consisting of an →interaction operator and →interaction fragments as operands.

Comment [UML]: A comment is a textual annotation that can be attached to a set of elements. Synonym: Note.

Communication path [UML]: A communication path is an →association between →actor and →use case, or between actor and system. The name is used synonymously for a relationship in the →deployment diagram.

Complexity [other]: Refers to the number and type of relationships between elements in a system.

Composite state [UML]: A →state that has at least one →region.

Composite structure diagram [UML]: A diagram that describes the internal structure of a →class consisting of →roles and →connectors. The diagram is available in UML only. SysML uses a different form called →internal block diagram.

Composition [UML]: A composition denotes a →class as an aggregate and describes a whole-part hierarchy. The aggregate is existentially responsible for its parts.

Conform [SysML]: A relationship that connects a →model view with a →viewpoint the requirements of which it meets.

Connector [UML]: A connector specifies a relationship between two →roles that allows them to communicate.

Constraint [UML]: A condition that constrains the semantic of model elements, and which must always be met.

Constraint block [SysML]: A block that describes →constraints of system structures and the parameters required for this.

Context object [UML]: A relative term that refers to a behavior and the →object in which that behavior is executed.

Continuous use case [SYSMOD]: A special →use case that starts in a defined system state and continually supplies results. A final result is not required.

Control flow [UML]: An →activity edge that is traversed by →control tokens only.

Control node [UML]: A node in an →activity that controls the flow of →control tokens or →object tokens.

Control operator [SysML]: A control operator specifies a behavior that can enable and disable →actions by use of →control values.

Control pin [UML]: A →pin that accepts →object tokens. It can cause an →action to be executed, but does not forward the →object to the action.
Control token [UML]: Token.
Control value [SysML]: An enumerated value that is used to control a control operator.
Copy [SysML]: A relationship describing that a requirement is a copy of another requirement.
Core requirement [SYSMOD]: A requirement that refers to the entire system; it is independent of the particularities of system variants.

D
DataType [UML]: A type with instances that can be identified by their values only.
Decision node [UML]: The decision is a node in an activity where several optional flows branch. There is exactly one incoming edge and an arbitrary number of outgoing edges, each having a condition.
Dependency [UML]: A relationship between two elements which describes that one element requires another element for its specification or implementation.
Deployment diagram [UML]: A diagram that shows the hardware structure and the deployment of software. This diagram is not part of SysML.
Derive requirement [UML]: A requirement that has been derived from another requirement.
Derived association/attribute [UML]: An association or attribute that can be derived from other model elements. Not to be confused with generalization.
 Destruction event [UML]: An event that specifies the time in an interaction at which the instance belonging to the =e lifeline will be destroyed.
Diagram frame [UML]: A rectangle around a SysML/UML diagram with a diagram heading in the upper left corner which describes the diagram (type, name, and other information).
Dimension [SysML]: A dimension describes the quantity of a unit.
Distribution definition [SysML]: A definition that describes in the form of a defined value range how values are distributed.
Do behavior [UML]: An optional behavior that is executed if the state is active.
Domain block [SYSMOD]: A domain block represents an object, a concept, a location, or a person from the real-world domain. A domain block is directly known to the system.
Domain experts [other]: A domain consists of domain experts who supply domain-specific requirements.
Domain model [SYSMOD]: A domain model describes the domain blocks of a system and their relationships.

F
Edge [UML]: Synonym: Activity edge.
Enhanced Functional Flow Block Diagram (EFFBD) [other]: A diagram based on the Functional Flow Block Diagram (FFBD), which was developed by TRW in the 1950s, representing complex flows in a simple way. An EFFBD adds data flow information.
Entry behavior [UML]: An optional behavior that is executed immediately upon entering a state.
Entry point [UML]: A particular point of entry in a state machine. From the entry point, a transition leads to a state, or to a state in each region in case of orthogonal regions.
Enumeration [UML]: A special data type with a value range consisting of a limited set of defined literals.
Environmental effect [SYSMOD]: A factor in the environment that influences the system without communicating directly with it.
Essential activity [SYSMOD]: An essential activity denotes activities that describe the details of an essential use case step.
Essential requirement [SYSMOD]: A requirement describing the pure domain intention, regardless of the technical implementation (solution).
Event [UML]: An occurrence, the time and location of which can be measured, that can trigger behavior in an object. SysML/UML distinguishes between call, change, signal, and time events.
Execution specification [UML]: Specifies that the object represented by the =lifeline executes behavior at this point.
Exit behavior [UML]: An optional behavior that is executed immediately prior to exiting a state.

Exit point [UML]: An exit point stops a transition in any region of the state machine when a transition in any region of the state machine reaches an exit point, then the state machine is terminated, and the transition outgoing from the exit point is activated.

Expansion region [UML]: A node in an activity that accepts a set of objects, then processes each of these objects individually, and finally returns the set of processed objects.

Extension [UML]: A relationship that extends an UML model element by additional properties that are defined as stereotype.

External system [SYSMOD]: A system that interacts directly with the system to be modeled. In its role as an interaction partner, an external system is considered merely a black box.

F

Final state [UML]: A state that describes the end of a composite state or a state machine.

Flow allocation [SysML]: Flow allocation connects an information object flow in a structure diagram with a flow edge in an activity diagram.

Flow final node [UML]: A final node that terminates a flow in an activity.

Flow port [SysML]: Describes an interaction point of a system block, including its environment, over which objects can flow into the block or out of it.

Flow specification [SysML]: A special interface that specifies data incoming and outgoing over a flow port.

Fork node [UML]: A node in an activity, which splits a flow into several concurrent flows.

Frequency [SysML]: A rate.

G

Generalization [UML]: A generalization is a taxonomic relationship between a more general class and a more specific class. Synonym: Specialization, Inheritance.

Glossary [SYSMOD]: The glossary explains all domain-specific terms of a project in a style similar to a lexicon.

H

History pseudo state [UML]: A pseudo state that stores the last state configuration of a region in which it resides. We distinguish between deep history (with substates) and shallow history (without substates).

I

Include relationship [UML]: A relationship describing that a use case is included in another use case.

Information flow [UML]: A directed relationship between actors, use cases, classes, ports, roles, interfaces, packages, or objects. It shows that information items are exchanged between these elements.

Information item [UML]: An abstract concept of UML used to model the presence and conveyance of information on a coarse level.

Inheritance [UML]: Generalization.

Initial node [UML]: The starting point for a flow that is started when an activity is invoked.

Initial state [UML]: An initial state is a pseudo state with an outgoing transition that points to the initial state.

Input pin [UML]: An input.

Instance [UML]: Synonym: Item, Instance specification, Object.
Instance specification [UML]: Describes a specific instance that has been created by the building plan of a type description (e.g., a class). Synonym: Item, Instance, Object.

Interaction [UML]: Describes a communication between =lifelines. This communication is based on the exchange of messages in the form of =operation calls or =signals.

Interaction fragment [UML]: Part of an =interaction.

Interaction operator [UML]: The operator of a =combined fragment. SysML/UML defines these operators: alt, opt, break, loop, seq, strict, par, critical, neg, assert, consider, and ignore.

Interaction use [UML]: A reference to an =interaction. The model is designed such that the reference could be substituted by the referenced interaction.

Interface [UML]: An interface specifies structure and behavior. It does not contain any implementation, and no =object can be created by its building plan.

Internal block diagram [SysML]: A special composite structure diagram that describes the structure of a =block.

Interruptible activity region [UML]: A region within an =activity that can be terminated by a =token flow via special interruptible edges.

Intricacy [other]: Refers to the number of different elements in a system.

Item flow [SysML]: A special =information flow, which describes at a =connector in the internal block diagram that specific =objects are being transported.

J

Join node [UML]: A node in an =activity that synchronizes several concurrent flows grouping them into one. There is an arbitrary number of incoming edges and exactly one outgoing =edge.

Join pseudo state [UML]: A =pseudo state that groups =transitions from orthogonal =regions.

Junction pseudo state [UML]: A =pseudo state that connects =transitions and composes them into a path.

L

Lifeline [UML]: A lifeline represents a communication partner in an =interaction. It describes the element’s name, type, and lifecycle.

Link [UML]: An =instance of an =association (i.e., a specific relationship) between two =objects.

M

Measure of Effectiveness (MOE) [SysML]: MOE, also called Effectiveness Measure, is a metric stating a customer’s satisfaction with the technical properties of a system.

Mechanical system (SYSMOD): A special external system that has only mechanical aspects from the own system’s view.

Merge node [UML]: A node in an =activity at which several flows are merged into one flow. There is an arbitrary number of incoming edges and exactly one outgoing =edge.

Message [UML]: A form of communication between two =lifelines. It can be either =synchronous or =asynchronous. It can invoke an =operation, or transport a =signal, or create an =object.

Model [UML]: A model describes a =system for a specific purpose.

Model of Models (MoM) [other]: The =model of a =system of systems. It consists of independent models that, together, describe a system.

Modeling tool [other]: A software application used to create and manage =SysML or =UML models.

Multiplicity [UML]: An interval of positive integers that describes how many objects an =attribute can accept.

N

Namespace [UML]: The namespace contains all elements that can be uniquely identified by their names. Examples for namespaces are =packages and =system blocks.
Namespace containment [UML]: A relationship describing that a ⇒ namespace is included in another ⇒ namespace.

Navigation [UML]: A property of associations specifying that objects at one ⇒ association end can access objects at the other end.

Note [UML]: Synonym: ⇒ Comment.

Null token [UML]: A special ⇒ object token that contains a value of null.

O

Object [UML]: Synonym: Instance, ⇒ Object.

Object [UML]: Synonym: Item, Instance, ⇒ Instance specification.

Object Constraint Language [UML]: A text-driven formal language used to formulate ⇒ constraints in SysML/UML models. The language supports, among other things, navigation in object models, Boolean Algebra, and set operations.

Object diagram [UML]: A diagram that shows ⇒ objects and their relationships. It is a diagram of UML and does not exist in SysML.

Object flow [UML]: An ⇒ activity edge that can be traversed by object tokens only.

Object identity [UML]: A property of the ⇒ objects of a ⇒ class that distinguishes them uniquely from other objects, regardless of the ⇒ attribute values.

Object node [UML]: An object node is an abstract ⇒ activity node that is part of defining object flow in an activity.

Object token [UML]: ⇒ Token.

OpaqueAction [UML]: An elementary executable ⇒ action; its implementation is formulated in an arbitrary language (e.g., in a programming language).

Operation [UML]: An operation defines a behavior property of a ⇒ class. The description consists of visibility, name, parameters, and return type.

Optional parameter [SysML]: Describes ⇒ parameters that do not have to have values for the pertaining behavior to be executed.

Output pin [UML]: ⇒ Pin.

P

Package [UML]: A package groups model elements and forms a namespace.

Package diagram [UML]: A diagram that shows how ⇒ packages relate, and how model elements are distributed across packages.

Parameter [UML]: A parameter describes values that are forwarded to, or returned from, a behavior element (e.g., an ⇒ operation).

Parameter set [UML]: A parameter set is a complete set of input or output parameters of a behavior, which is selected regardless of other parameter sets of that behavior.

Parametric diagram [SysML]: A diagram that shows a network of ⇒ constraints for the purpose of modeling performance and reliability models.

Part decomposition [UML]: Describes the internal ⇒ interactions of a ⇒ lifeline.

Participant property [SysML]: A participant property describes the end of an ⇒ association in the internal structure of an ⇒ association block.

Partition [UML]: ⇒ Activity partition.

Pin [UML]: A pin is a link between the parameters of an ⇒ action and the object flow. We distinguish between ⇒ input pin and ⇒ output pin.

Port [UML]: A port describes an interaction point that is used by a ⇒ class (UML) or system block (SysML) of the environment provides or requests services over ⇒ interfaces.

Postcondition [UML]: A Boolean expression that is true once a behavior has executed.

Precondition [UML]: A Boolean expression that has to be true before a given behavior can be executed.

Primary use case [SYSMOD]: A ⇒ use case that describes a central service of the system.

Primitive type [UML]: A type describing a ⇒ data type that has no structures.

Probability [SysML]: Describes the outgoing edges of a ⇒ decision or an ⇒ object node the probability that this ⇒ edge will be used by a ⇒ token.
Problem [SysML]: A problem documents an (potential) error or weakness in the model or in the modeled system.

Profile [UML]: A profile is a set of stereotypes.

Profile application [UML]: A profile application assigns a profile to a package, allowing the use of the stereotypes contained in the profile at the model elements in the package.

Property [UML]: A property describes a part of the structure of a structural element (e.g., a class).

Property string [UML]: A string that, in a diagram, shows a certain property of the pertaining model element (e.g., (readonly)).

Pseudo state [UML]: A control element that influences the flow of a state machine. It is not a real state, so that the pseudo state does not represent any value combination.

R
Rate [SysML]: The rate describes the frequency in which elements traverse an activity edge, or in which they flow to or from a parameter. Synonym: Frequency.

Rationale [SysML]: A rationale documents the principles or reasons for a modeling decision.

Realization [UML]: A relationship that connects an implementation with a specification. The implementation is responsible for realizing that specification.

Refine [UML]: A relationship describing that a model element describes the properties of a requirement in more detail.

Region [UML]: A region is an orthogonal area in a state or state machine.

Representation [UML]: A relationship that describes the model element that is represented by a piece of information.

Requirement [SysML]: A requirement describes properties or behavior of a system that always have to be met.

Requirement diagram [SysML]: A diagram that shows requirements and their relationships.

Risk management [SYSMOD]: Risk management denotes the planned handling of risks. Potential risks are identified and evaluated, and counteractions are formulated for prevention and limitation of damages.

Role [UML]: A role describes a structure in the context of a class.

S
Satisfy [SysML]: A relationship describing that a design element meets a requirement.

Scenario (other): A specific sequence for example, a possible variant of a use case.

Secondary use case [SYSMOD]: A secondary use case is an incomplete use case fragment. It lacks domain action, result, and actor.

SendSignalAction [UML]: An elementary action that sends a signal.

Sensor [SYSMOD]: A sensor is a special external system that accepts information from the environment and forwards it to the system.

Sequence diagram [UML]: A diagram that shows an interaction, focusing on the temporal sequence of messages.

SI – International System of Units (other): The International System of Units (French: Le Système international d'unités) is the widest used system for physical units. It defines the meter, kilogram, second, ampere, Kelvin, mole, and candela basic units.

Sign [UML]: A signal describes the structure of a communication object.

Signal event [UML]: Event.

Specialization [UML]: Generalization.

Stakeholder [SYSMOD]: A stakeholder is an individual or organization that has a direct interest in the system and that may have requirements.

Stakeholder [SYSMOD]: A stakeholder is a person or institution that has an interest in the system and may make requirements.

Standard port [SysML]: Synonym: Port.

State [UML]: A state represents a set of value combinations for a given element. A state has a name and may have an internal behavior that is executed based on defined events.
State invariant [UML]: A constraint that refers to a lifetime, and which must be met at system runtime.

State machine [UML]: A state machine describes the states and transitions of a structure.

State machine diagram [UML]: A diagram that depicts a state machine.

Stereotype [UML]: A stereotype expands an existing model element by additional properties and semantics. The newly defined model element can include a new notation, in addition to the name. Stereotypes are grouped in profiles.

Streaming [UML]: A property describing that activities or actions can accept or supply new values during active operation.

Structural allocation [SysML]: A structural allocation is used to separate logical from physical structures by producing a relationship between the two levels.

Structure diagram [UML]: A generic term for all static diagrams in SysML and UML.

Subsystem [SysML]: A subsystem describes a closed unit within a larger system.

Synchronous message [UML]: The sender of the message waits until the receiver has processed the message.

SysML [SysML]: The Systems Modeling Language (SysML) is a graphical language for modeling systems in the systems engineering discipline.

SysML tool [other]: A modeling tool.

System [SysML]: A system is a collection of system blocks that pursue a common goal, which cannot be achieved by the individual elements. A block can be software, hardware, a person, or any other unit.

System actor [SysML]: A role.

System context diagram [SysML]: A diagram that shows the system as a black box, including its environment, and information that can be exchanged with the environment.

System context element [SysML]: A virtual wrapper that comprises the entire system and its actors.

System of Systems (SoS) [other]: A system composed of blocks that can, in turn, be independent systems.

System port [SysML]: A port at a system.

System process [SysML]: The process that describes a flow beyond the use cases. It consists of a set of use cases that have a domain-specific sequence.

Systems engineering [other]: Systems engineering is a discipline that concentrates on the definition and documentation of system requirements in the early development stage, the elaboration of a system design, and the verification of the system as to compliance with the requirements, taking the entire problem—operation, time, test, creation, cost, and planning, training, and support, and disposal—into account.

Technical requirement [SysML]: A requirement that describes a requirement based on a solution approach.

Terminiate [UML]: A pseudo state describing that the pertaining state machine or the pertaining context object has terminated.

Test case [SysML]: A test case is a sequence that verifies whether or not the system meets a given requirement.

Time event [UML]: An event.

Token [UML]: A virtual element that describes the position of a flow in an activity. Control tokens mark only the flow, while object tokens additionally show that there is a defined object at that position.

Token flow [UML]: A flow that uses control tokens and object tokens to describe the flow of an activity.

Tool [other]: A software application designed to support the development process (e.g., a modeling tool).

Trace [UML]: A relationship between two model elements, describing a general context.
Transition [UML]: A transition specifies the passing from one state into another. It is a directed relationship between two states. It defines a trigger and a condition that both cause the state transition and behavior that is executed during that transition.

Trigger [UML]: A trigger connects exactly one event with a behavior.

Type [UML]: A type defines a value range (e.g., a primitive data type or a block).

U

UML [UML]: The Unified Modeling Language (UML) is a graphical modeling language used to describe software and other systems.

UML tool [other]: A modeling tool.

Unit [SysML]: A unit describes the structure of a physical unit (e.g., kilogram, meter).

Use case [UML]: A use case describes a temporally related and targeted interaction between an actor and a system. Its beginning is a domain trigger and the outcome is a defined result of domain value.

Use case diagram [UML]: A diagram that shows actors and use cases and their relationships.

User [SysMOD]: A user is a human actor.

User system [SysMOD]: A user system is a special external system that serves the user as a medium to interact with the system.

V

Value type [SysML]: A type that defines values, which have no identity, and are not referenced by a block, and which can have a unit or dimension.

Variant requirement [SysMOD]: A requirement that refers exclusively to a variant, and which is valid only for the system design of that variant.

Verify [SysML]: A relationship that connects a test case with the requirement that is tested by that test case.

View [SysML]: A representation of an entire system seen from a defined viewpoint.

Viewpoint [SysML]: A viewpoint specifies the structure of a model view based on the targets defined by a number of stakeholders.

W

Weighted satisfy [SysMOD]: A relationship that adds coverage information to a satisfy relationship.

Weighted verify [SysMOD]: A relationship that adds coverage information to a verify relationship.

X

XMI [UML]: The XML Metadata Interchange (XMI) is a data exchange format for models formulated in XML.
ANEXOS

B. PROGRAMA EN MATLAB

El programa en Matlab lee las entradas correspondientes a los valores mencionados en la Tabla 4-1 que refieren al diagrama paramétrico de la Figura 4.23 y que están presentes en la Instancia del diagrama paramétrico (ver Figura B.1). ParaMagic genera automáticamente un documento de texto llamado input.txt al momento de ejecutar la solución (ver Figura B.2 y Figura B.3). El script en Matlab se muestra en la Figura B.4.

Figura B.1 Instancia con valores en SysML;
Fuente propia
Figura B.2 Ventana de *ParaMagic* para resolver la instancia (contexto en Figura 4.25); Fuente propia

Figura B.3 Documento de texto generado por *ParaMagic*; Fuente propia

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>clear all</td>
</tr>
<tr>
<td>2</td>
<td>close all</td>
</tr>
<tr>
<td>3</td>
<td>clc</td>
</tr>
<tr>
<td>4</td>
<td>%Leer las entradas</td>
</tr>
<tr>
<td>5</td>
<td>inputs=load('input.txt');</td>
</tr>
<tr>
<td>6</td>
<td>%Cargar valores de las entradas</td>
</tr>
<tr>
<td>7</td>
<td>xd=inputs(1);</td>
</tr>
<tr>
<td>8</td>
<td>alfa=inputs(2);</td>
</tr>
<tr>
<td>9</td>
<td>m=inputs(3);</td>
</tr>
<tr>
<td>10</td>
<td>cb=inputs(5);</td>
</tr>
<tr>
<td>11</td>
<td>xlim=inputs(6);</td>
</tr>
<tr>
<td>12</td>
<td>N=inputs(7);</td>
</tr>
<tr>
<td>13</td>
<td>wmax=inputs(8);</td>
</tr>
<tr>
<td>14</td>
<td>Gc=inputs(9);</td>
</tr>
<tr>
<td>15</td>
<td>G2=inputs(10);</td>
</tr>
<tr>
<td>16</td>
<td>Gb1=inputs(11);</td>
</tr>
<tr>
<td>17</td>
<td>Gbcomp=inputs(12);</td>
</tr>
<tr>
<td>18</td>
<td>Kinner=inputs(13);</td>
</tr>
<tr>
<td>19</td>
<td>Kv=inputs(14);</td>
</tr>
<tr>
<td>20</td>
<td>Vsat=inputs(15);</td>
</tr>
<tr>
<td>21</td>
<td>M=inputs(16);</td>
</tr>
<tr>
<td>22</td>
<td>Bl=inputs(17);</td>
</tr>
<tr>
<td>23</td>
<td>Kl=inputs(18);</td>
</tr>
<tr>
<td>26</td>
<td>%Calcular otros valores</td>
</tr>
<tr>
<td>27</td>
<td>a partir de las entradas</td>
</tr>
<tr>
<td>28</td>
<td>wmaxrad=wmax<em>2</em>pi/60;</td>
</tr>
<tr>
<td>29</td>
<td>Kconv1=wmaxrad/xlim;</td>
</tr>
<tr>
<td>30</td>
<td>M=M/32.2;</td>
</tr>
<tr>
<td>31</td>
<td>Pinner=N;</td>
</tr>
<tr>
<td>32</td>
<td>kf=xd/120;</td>
</tr>
<tr>
<td>33</td>
<td>%Cargar modelo Simulink</td>
</tr>
<tr>
<td>34</td>
<td>mdl='simulinkEMA_v2';</td>
</tr>
<tr>
<td>35</td>
<td>open_system(mdl);</td>
</tr>
<tr>
<td>36</td>
<td>%Correr la simulación y visualizar los resultados</td>
</tr>
<tr>
<td>37</td>
<td>evalc('sim(mdl)');</td>
</tr>
<tr>
<td>38</td>
<td>open_system([mdl,'/Resultados']);</td>
</tr>
<tr>
<td>39</td>
<td>%Guardar la salida</td>
</tr>
<tr>
<td>40</td>
<td>ys=yout(95,1);</td>
</tr>
<tr>
<td>41</td>
<td>%Calcular el error en estado estacionario</td>
</tr>
<tr>
<td>42</td>
<td>e=(xd-ys)/xd*100;</td>
</tr>
<tr>
<td>43</td>
<td>e=e*1000;</td>
</tr>
<tr>
<td>44</td>
<td>e=floor(e)/1000;</td>
</tr>
<tr>
<td>45</td>
<td>%Guardar el error y cerrar</td>
</tr>
<tr>
<td>46</td>
<td>save('output.txt', 'e','-ASCII');</td>
</tr>
<tr>
<td>49</td>
<td>exit</td>
</tr>
</tbody>
</table>

Figura B.4 Programa en *Matlab* EMAscriptSimulink.m; Fuente Propia
Durante la ejecución del *script* este llamará a un programa en *Simulink* para ejecutar el modelo del actuador electro-mecánico. Los valores después de la ejecución teniendo como posición deseada 0.5 pulgadas se muestran en la Figura B.5. El modelo del EMA en *Simulink* se muestra en la Figura B.6. Los resultados de la posición final se muestran en la Figura B.7 en dónde el actuador llega a una posición de 0.48 pulgadas aproximadamente y el error en estado estacionario es de 4.88%. Se muestra el documento ‘*output.txt*’ generado por *Matlab* en la Figura B.8.

![Workspace después de la ejecución del Script en Matlab](image)

Figura B.5 *Workspace* después de la ejecución del *Script* en *Matlab*; Fuente Propia
Figura B.6 Modelo del EMA en Simulink llamado simulinkEMA_v2.mdl;
Fuente Propia
Figura B.7 Scope 'Resultados'; Fuente propia

Figura B.8 Documento de texto 'output.txt’ generado por Matlab; Fuente propia